Dissimilarity representations in pattern recognition.

Concepts, theory and applications.

Dissimilarity representations in pattern recognition.

Concepts, theory and applications.

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,
op gezag van de Rector Magnificus prof. dr. ir. J.T. Fokkema,
voorzitter van het College voor Promoties,
in het openbaar te verdedigen op 17 januari 2005 om 13.00 uur

door

Elżbieta Małgorzata PĘKALSKA

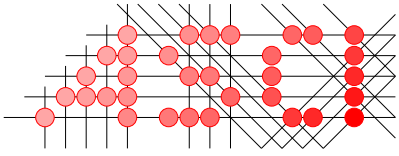
magister infomatyki Uniwersytetu Wrocławskiego,
geboren te Wrocław, Polen.

Dit proefschrift is goedgekeurd door de promotor:
Prof. dr. I.T. Young

Toegevoegd promotor: Dr. ir. R.P.W. Duin

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Prof. dr. I.T. Young,	Technische Universiteit Delft, promotor
Dr. ir. R.P.W. Duin,	Technische Universiteit Delft, toegevoegd promotor
Prof. dr. A.L.N. Fred,	Technische Universiteit van Lisbon, Portugal
Prof. dr. H. Bunke,	Universiteit van Bern, Zwitserland
Prof. dr. E.O. Postma,	Universiteit Maastricht
Prof. dr. ir. A.W.M. Smeulders,	Universiteit van Amsterdam
Prof. dr. J.M. Aarts,	Technische Universiteit Delft
Prof. dr. ir. E. Backer,	Technische Universiteit Delft, reservelid



This work was carried out in the ASCI graduate school. ASCI dissertation series number 109.

This work was partly supported by the Dutch Organisation for Scientific Research (NWO).

ISBN 90-9019021-X

Dissimilarity representations in pattern recognition.
Concepts, theory and applications.

Thesis

presented for the degree of doctor
at Delft University of Technology
under the authority of the Vice-Chancellor, prof. dr. ir. J.T. Fokkema,
to be defended in public in the presence of a committee
appointed by the Board for Doctorates
on 17 Januari 2005 at 13.00

by

Elżbieta Małgorzata PĘKALSKA (GÓRNIEWICZ)

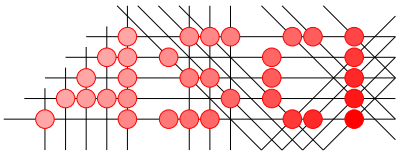
MSc in computer science from Wrocław University,
born in Wrocław, Poland.

This thesis is approved by the supervisor:
Prof. dr. I.T. Young

Adjunct supervisor: Dr. ir. R.P.W. Duin

Composition of the Doctoral Examination Commitee:

Vice-Chancellor	chairman
Prof. dr. I.T. Young,	Delft University of Technology, supervisor
Dr. ir. R.P.W. Duin,	Delft University of Technology, adjunct supervisor
Prof. dr. A.L.N. Fred,	Technical University of Lisbon, Portugal
Prof. dr. H. Bunke,	University of Bern, Switzerland
Prof. dr. E.O. Postma,	University of Maastricht
Prof. dr. ir. A.W.M. Smeulders,	University of Amsterdam
Prof. dr. J.M. Aarts,	Delft University of Technology
Prof. dr. ir. E. Backer,	Delft University of Technology, reserve member



This work was carried out in the ASCI graduate school. ASCI dissertation series number 109.
This work was partly supported by the Dutch Organisation for Scientific Research (NWO).

ISBN 90-9019021-X

Moim rodzicom
Andrzejowi

To my parents
to Andrzej
and ...

to the ones who
ask questions
and look for answers

Contents

Notation and basic terminology

Latin symbols

$A, B, \ldots, Z$	matrices, linear (vector) spaces or sets
$a, b, \ldots, z$	scalars, vectors or object identifiers
$\mathbf{a}, \mathbf{b}, \ldots, \mathbf{z}$	vectors in an m-dimensional vector space $\mathbb{R}^m$
$\overline{\mathbf{x}}, \overline{\mathbf{y}}$	estimated mean vectors in $\mathbb{R}^m$
G	Gram matrix
C	estimated covariance matrix
d	dissimilarity function
D	dissimilarity matrix
f, g, h	functions
I	identity operator or identity matrix
k	number of clusters
k, m, n	space dimensions
K	kernel
n, N	number of objects or vectors, usually in learning
p_i	i-th object in the representation set R
P	projection operator or matrix
Q	orthogonal matrix
R	representation set $R := \{p_1, p_2, \ldots, p_n\}$
s	similarity function
S	similarity matrix or stress function
t_i	i-th object in the training set T
T	training set $T := \{t_1, t_2, \ldots, t_N\}$
$\mathbf{v}, \mathbf{w}$	weight vectors or vectors of discriminant coefficients
X^*	algebraic dual space of a vector space X

Greek symbols

$\alpha, \beta, \ldots, \omega$	scalars or parameters
$\boldsymbol{\alpha}, \boldsymbol{\beta}, \ldots, \boldsymbol{\omega}$	vectors of parameters
δ	Kronecker delta function
Δ	dissimilarity matrix used in multidimensional scaling
γ	trade-off parameter in mathematical programming formulations
Γ	field, usually $\mathbb{R}$ or $\mathbb{C}$, or a gamma function
λ	regularization parameter
λ_i	i-th eigenvalue
Λ	diagonal matrix of eigenvalues
μ	mean, a probability measure or a membership function for fuzzy variables
$\boldsymbol{\mu}$	mean vector
ϕ, ψ, Φ, Ψ	mappings
Σ	covariance matrix
ρ	dissimilarity function
Ω	set, bounded interval or a closed and bounded subset of $\mathbb{R}^m$

Other symbols

$\mathbb{C}$	set of complex numbers
$\mathbb{C}^m$	m-dimensional complex space
$\mathcal{D}$	domain of a mapping or a function
$\mathcal{F}$	set of features
$\mathcal{H}$	Hilbert space
$\mathcal{I}(a)$	identificator (characteristic) function; it takes 1 if the condition a is true and 0 otherwise
$\mathcal{J}$	fundamental symmetry operator in Kreĭn spaces
$\mathcal{G}, \mathcal{K}$	Kreĭn spaces
$\mathcal{N}(\boldsymbol{\mu}, \Sigma)$	normal distribution with the mean $\boldsymbol{\mu}$ and the covariance matrix Σ
$\mathbb{R}$	set of real numbers
$\mathbb{R}_+$	set of real positive numbers
$\mathbb{R}_+^0$	$\mathbb{R}_+ \cup \{0\}$
$\mathbb{R}^m$	m-dimensional real vector space
$\mathcal{S}_r^m$	m-dimensional spherical space, i.e. $\mathcal{S}_r^m = \{\mathbf{x} \in \mathbb{R}^{m+1} : \sum_{i=1}^{m+1} x_i^2 = r^2\}$
$\mathcal{U}, \mathcal{V}, \mathcal{X}$	subsets or subspaces
$\mathbb{Z}$	set of integers

Sets and pretopology

$\lvert A \rvert$	cardinality of the set A
A°	generalized interior of A
A^-	generalized closure of A
$A \cup B$	union of A and B
$A \cap B$	intersection of A and B
$A \backslash B$	set difference of A and B
$A \triangle B$	set symmetric difference, $A \triangle B = (A \backslash B) \cup (B \backslash A)$
$A \times B$	Cartesian product, i.e. $A \times B = \{(a, b) : a \in A \ \wedge \ b \in B\}$
$\mathcal{P}(X)$	power set, i.e. a collection of all subsets of X
$\mathcal{N}(x)$	neighborhood system
$\mathcal{N}_B(x)$	neighborhood basis
$(X, \mathcal{N})$	neighborhood (pretopological) space
$(X, \mathcal{N}_B)$	pretopological space defined by the neighborhood basis
$(X, {}^-)$	neighborhood (pretopological) space defined by the generalized closure
(X, ρ)	generalized metric space with a dissimilarity ρ
$B_\varepsilon(x)$	ε- ball in a generalized metric space (X, ρ), $B_\varepsilon(x) = \{y \in X : \rho(y, x) < \varepsilon\}$
σ-algebra	collection of subsets $\mathcal{A}$ of the set Ω satisfying: (1) $\Omega \in \mathcal{A}$, (2) $A \in \mathcal{A} \Rightarrow (\Omega \backslash A) \in \mathcal{A}$ and (3) $(\forall_k \ A_k \in \mathcal{A} \ \wedge \ A = \bigcup_{k=1}^{\infty} A_k) \Rightarrow A \in \mathcal{A}$
μ	$\mu : \mathcal{A} \to \mathbb{R}_+^0$ is a measure on a σ-algebra $\mathcal{A}$ if $\mu(\emptyset) = 0$ and μ is additive, i.e. $\mu\left(\bigcup_k A_k\right) = \sum_k \mu(A_k)$ for pairwise disjoint sets A_k
$(\Omega, \mathcal{A}, \mu)$	a measure space; Ω is a set, $\mathcal{A}$ is a σ-algebra on Ω and μ is a measure

Vectors, matrices and operators

$\mathbf{0}$	column vector of m zeros in $\mathbb{R}^m$
$\mathbf{1}$	column vector of m ones in $\mathbb{R}^m$
$\mathbf{e}_i$	standard basis vector in $\mathbb{R}^m$
$\mathbf{x}^T \mathbf{y}$	scalar inner product of vectors in $\mathbb{R}^m$
$A = (a_{ij})$	a matrix or an operator A with the elements a_{ij}
$a_{i\cdot}, A_{i\cdot}$	i-th row of a matrix A

$a_{\cdot j}, A_{\cdot j}$	j-th column of a matrix A
A^T	transpose of a real matrix A
$A^\dagger$	conjugate transpose of a complex matrix A
$A^\times$	adjoint of an operator A in a Hilbert space
A^*	adjoint of an operator A in a Kreĭn space
$\|A\|$	determinant of a matrix A
A^{*p}	Hadamard power, $A^{*p} = (a_{ij}^p)$
$A * B$	Hadamard product, $A * B = (a_{ij} b_{ij})$
a^{*B}	Hadamard power, $a^{*B} = (a^{b_{ij}})$, where $a \in \mathbb{R}$
A hermitian	$A = A^\dagger$
A symmetric	$A = A^T$
A orthogonal	$A\,A^T = I$ and $A^T A = I$
A unitary	$A\,A^\dagger = I$ and $A^\dagger A = I$
A cnd	$A = A^\dagger$ is *c*onditionally *n*egative *d*efinite if $\mathbf{x}^\dagger A\,\mathbf{x} \le 0$ and $\mathbf{x}^\dagger\,\mathbf{1} = 0$ for $\mathbf{x} \ne \mathbf{0}$
A cpd	$A = A^\dagger$ is *c*onditionally *p*ositive *d*efinite if $\mathbf{x}^\dagger A\,\mathbf{x} \ge 0$ and $\mathbf{x}^\dagger\,\mathbf{1} = 0$ for $\mathbf{x} \ne \mathbf{0}$
A nd	$A = A^\dagger$ is *n*egative *d*efinite if $\mathbf{x}^\dagger A\,\mathbf{x} < 0$ for $\mathbf{x} \ne \mathbf{0}$
A nsd	$A = A^\dagger$ is *n*egative *s*emi*d*efinite if $\mathbf{x}^\dagger A\,\mathbf{x} \le 0$ for $\mathbf{x} \ne \mathbf{0}$
A pd	$A = A^\dagger$ is *p*ositive *d*efinite if $\mathbf{x}^\dagger A\,\mathbf{x} > 0$ for $\mathbf{x} \ne \mathbf{0}$
A psd	$A = A^\dagger$ is *p*ositive *s*emi*d*efinite if $\mathbf{x}^\dagger A\,\mathbf{x} \ge 0$ for $\mathbf{x} \ne \mathbf{0}$

Sets, inner product and normed spaces

$\mathcal{F}(\Omega)$	set of all functions on a compact set $\Omega \subset \mathbb{R}^m$
$\mathcal{C}(\Omega)$	set of all continuous functions on a compact set $\Omega \subset \mathbb{R}^m$
$\mathcal{M}(\Omega)$	set of classes of functions, Lebesgue measurable on a compact set $\Omega \subset \mathbb{R}^m$
$L_p^{\mathcal{C}}$	$L_p^{\mathcal{C}} = \{f \in \mathcal{C}(\Omega)\colon\ (\int_\Omega \|f(x)\|^p\,dx)^{1/p} < \infty\}$, where $p \ge 1$
$L_p^{\mathcal{M}}$	$L_p^{\mathcal{M}} = \{f \in \mathcal{M}(\Omega)\colon\ (\int_\Omega \|f(x)\|^p\,\mu(dx))^{1/p} < \infty\}$, where $p \ge 1$
$\mathcal{L}_c(X, \Gamma)$	space of continuous linear functionals from X onto Γ
$\mathcal{L}_c(X, Y)$	space of continuous linear operators from X onto Y
$X \oplus Y$	direct sum of subspaces; if $Z = X \oplus Y$, then every $z \in Z$ can be uniquely decomposed into $x \in X$ and $Y \in Y$ such that $z = x + y$ and $X \cap Y = \{0\}$
$X^\perp$	orthogonal complement to X; if $X \subseteq Z$, then $X^\perp = \{z \in Z\colon \forall_{x \in X}\ \langle z, x\rangle = 0\}$ and $Z = X \oplus X^\perp$
$\{x_i\}_{i=1}^n$	$\{x_1, x_2, \ldots, x_n\}$
$\langle\cdot,\cdot\rangle$	inner product
$\|\|\cdot\|\|$	norm
$\|\|\mathbf{x}\|\|_p$	l_p-norm of $\mathbf{x} \in \mathbb{R}^m$, $\|\|\mathbf{x}\|\|_p = (\sum_{i=1}^m \|x_i\|^p)^{1/p}$, $p > 0$
$\|\|f\|\|_p$	l_p-norm of $f \in L_p^{\mathcal{C}}$; $\|\|f\|\|_p = (\int_\Omega \|f(x)\|^p\,dx)^{1/p}$, $p \ge 1$
$(X, \langle\cdot,\cdot\rangle)$	space X with the inner product $\langle\cdot,\cdot\rangle$
$(X, \|\|\cdot\|\|)$	space X with the norm $\|\|\cdot\|\|$
(X, ρ)	space X with the dissimilarity ρ
$\mathcal{H}$	Hilbert space
$\mathcal{H}_K$	reproducing kernel Hilbert space equipped with the kernel K
l_p^m	Banach space $(\mathbb{R}^m, \|\|\cdot\|\|_p),\ p \ge 1$
l_p^∞	Banach space $(\mathbb{R}^\infty, \|\|\cdot\|\|_p),\ p \ge 1$

Indefinite inner product spaces

$\mathcal{E} := \mathbb{R}^{(p,q)}$	pseudo-Euclidean space with the signature (p, q)
$\mathcal{J}_{pq}$	fundamental symmetry in a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$
$\langle\cdot,\cdot\rangle_{\mathcal{E}}$	inner product in a pseudo-Euclidean space $\mathcal{E}$

$\mathcal{K}_+, \mathcal{K}_-$	Hilbert spaces $(\mathcal{K}_+, \langle\cdot,\cdot\rangle)$ and $(\mathcal{K}_-, -\langle\cdot,\cdot\rangle)$
$\mathcal{K}$	Kreĭn space, $\mathcal{K}=\mathcal{K}_+ \oplus \mathcal{K}_-$ and $\mathcal{K}_- = \mathcal{K}_+^\perp$
$\lvert\mathcal{K}\rvert$	Hilbert space associated with a Kreĭn space $\mathcal{K}$
	$\lvert\mathcal{K}\rvert = \mathcal{K}_+ \oplus \lvert\mathcal{K}_-\rvert$, where $\mathcal{K}_- = \mathcal{K}_+^\perp$ and $\lvert\mathcal{K}_-\rvert := (\mathcal{K}_-, \langle\cdot,\cdot\rangle)$
$\mathcal{K}_K$	reproducing kernel Kreĭn space equipped with the kernel K
P_+, P_-	fundamental projections
I	identity operator in a Kreĭn space; $I = P_+ + P_-$
$\mathcal{J}$	fundamental symmetry in a Kreĭn space; $\mathcal{J} = P_+ - P_-$
$\langle\cdot,\cdot\rangle_{\mathcal{K}}$	inner product in a Kreĭn space $\mathcal{K}$
$[x, y]$	H-scalar product, $[x, y] = \langle \mathcal{J}x, y\rangle_{\mathcal{K}}$
$\lvert\lvert x\rvert\rvert_h$	H-norm, $\lvert\lvert x\rvert\rvert_h = [x, x]^{\frac{1}{2}}$
$\mathcal{L}_c(\mathcal{K}, \Gamma)$	space of continuous linear functionals from a Kreĭn space $\mathcal{K}$ into Γ
$\mathcal{L}_c(\mathcal{K}, \mathcal{G})$	space of continuous linear operators from a Kreĭn space $\mathcal{K}$ into a Kreĭn $\mathcal{G}$
A^*	adjoint of $A \in \mathcal{L}(\mathcal{K}, \mathcal{G})$ is defined such that $\langle Af, g\rangle_{\mathcal{G}} = \langle f, A^*g\rangle_{\mathcal{K}}$ holds for all $f \in \mathcal{K}$ and $g \in \mathcal{G}$
A self-adjoint	$A = A^*$
A isometric	$A \in \mathcal{L}(\mathcal{K}, \mathcal{G})$ is isometric if $A^*A = I_{\mathcal{G}}$
A coisometric	$A \in \mathcal{L}(\mathcal{K}, \mathcal{G})$ is coisometric if $AA^* = I_{\mathcal{K}}$
A symmetric	$\langle Af, g\rangle_{\mathcal{K}} = \langle f, Ag\rangle_{\mathcal{K}}$ for all $f, g \in \mathcal{K}$
A unitary	$\langle Af, Ag\rangle_{\mathcal{K}} = \langle f, g\rangle_{\mathcal{K}}$ for all $f, g \in \mathcal{K}$

Mappings, functions and kernels

$\phi\colon X \to Y$	mapping from X to Y; X is the domain of ϕ and Y is the codomain of ϕ
$\phi \circ \gamma$	composition of mappings; if $\phi\colon X \to Y$ and $\gamma\colon Y \to Z$, then $\phi \circ \gamma\colon X \to Z$ is a mapping such that $x \to \gamma(\phi(x))$
null (ϕ), ker (ϕ)	null $(\phi) = \{x \in V : \phi(x) = 0\}$ for a homomorphism $\phi\colon V \to Z$
linear mapping	$\phi\colon X \to Y$ is such that X and Y are vector spaces and for all $x, y \in X$ and $\alpha \in \Gamma$ (1) $\phi(x + y) = \phi x + \phi y$ and (2) $\phi(\alpha x) = \alpha\phi(x)$ hold
bilinear mapping	$\phi\colon X \times Y \to \Gamma$ is such that X and Y are vector spaces and for all $x, x_1, x_2 \in X$, $y, y_1, y_2 \in Y$ and $\alpha, \beta \in \Gamma$ (1) $\phi(\alpha x_1 + \beta x_2, y) = \alpha\phi(x_1, y) + \beta\phi(x_2, y)$ and (2) $\phi(x, \alpha y + \beta z) = \alpha\phi(x, y_1) + \beta\phi(x, y_2)$ hold
injection	$\phi\colon X \to Y$ such that $\forall_{x,y\in X}\ x \neq y; \Rightarrow \phi(x) \neq \phi(y)$, called also a one-to-one mapping; $\mathbb{R}_\phi$ does not need to be equal to Y
bijection	injection which is also a surjection
surjection	mapping $\phi\colon X \to Y$, X *onto* Y, whose range coincides with the codomain
homomorphism	linear mapping from one vector space to another
endomorphism	linear mapping from a vector space to itself
isomorphism	homomorphism which is a bijection
monomorphism	homomorphism which is an injection
concave function	f is concave iff $f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha)f(y)$ holds for all $x, y \in \mathcal{D}_f$ and all $\alpha \in [0, 1]$
convex function	f is convex iff $-f$ is concave
logistic function	$h_{\log}(x) = 1/(1 + \exp(-\sigma x))$
sigmoid function	$f_{\text{sigm}}(x) = 2/(1 + \exp(-x/\sigma)) - 1$
gamma function	$\Gamma(t) = \int_0^\infty x^{t-1}e^{-x}dx,\ t > 0$

Dissimilarities

d	(generalized) dissimilarity measure
D	dissimilarity matrix

$D(T, R)$	dissimilarity representation; a dissimilarity matrix between the training objects from T and the representation objects from R
S	similarity matrix
d_2,D_E, D_2	Euclidean distance (matrix)
d_p, D_p	l_p-distance (matrix); $d_p(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^{m} \lvert x_i - y_i \rvert^p)^{\frac{1}{p}},\ p > 0$
$d_{\max}, D_{\max}$	l_∞-distance (matrix); $d_{\max}(\mathbf{x}, \mathbf{y}) = \max_i \lvert x_i - y_i \rvert$
D_G	Gower dissimilarity
d_{Ham}, D_{Ham}	Hamming distance (matrix)
d_H, D_H	Hausdorff distance (matrix)
d_{MH}, D_{MH}	modified Hausdorff distance (matrix)
d_M^2	square Mahalanobis distance
d_L, d_{nL}	Levenhstein distance, normalized Levenhstein distance
d_{KL}	Kullback-Leibner divergence
d_J	J-coefficient
d_{IR}	information radius divergence
d_{BH}	Bhattacharayya distance
d_{Ch}	Chernoff distance
$s_H^{(t)}$	Hellinger coefficient
d_T, s_T	Tversky dissimilarity and Tversky similarity
δ_V	cut semimetric based on the set V

Graphs and geometry

cut on X	partition of a set X into V and $X \backslash V$
$G = (V, E)$	graph with a set of nodes V and a set of edges $E := \{(u, v) : u, v \in V\}$
adjacent nodes	two nodes in a graph joined by an edge
$\text{hull}_\Gamma(X)$	linear hull; $\text{hull}_\Gamma(X) := \{\sum_{i=1}^{n} \beta_i x_i :\ x_i \in V\ \wedge\ \beta_i \in \Gamma \subseteq \mathbb{R}\}$
cone	$\{X : \text{hull}_{\mathbb{R}_+}(X) = X\}$
convex hull	$\text{convex\_hull}(X) := \{\sum_{i=1}^{n} \beta_i x_i :\ \ x_i \in V,\ \beta_i \geq 0\ \wedge\ \sum_{i=1}^{n} \beta_i = 1\}$
convex set	X is convex if for all $x, y \in X$ and $\beta \in [0, 1]$, $\beta x + (1 - \beta) y \in X$
hypercylinder	hyperprism for which the flat region in $\mathbb{R}^{m-1}$ is a hypersphere
hypersphere	$\{\mathbf{x} \in \mathbb{R}^m : \lvert\lvert\mathbf{x}\rvert\rvert_2^2 = R^2\}$ with the volume $V = \frac{2R^m \pi^{m/2}}{m\Gamma(m/2)}$ and the hyperarea $A = \frac{2R^{m-1}\pi^{m/2}}{\Gamma(m/2)}$
hyperplane	m-dimensional hyperplane $:= \{\mathbf{x} \in \mathbb{R}^{m+1} : \sum_{i=1}^{m+1} w_i x_i = w_0\}$
hyperprism	figure generated by a flat region in $\mathbb{R}^m$, moving parallel to itself along a straight line
parallelotope	collection of points in $\mathbb{R}^m$ bounded by m pairs of $(m-1)$-dimensional hyperplanes (a generalization of a parallelogram)
polyhedron	$\{\mathbf{x} \in \mathbb{R}^m : A\mathbf{x} \leq \mathbf{b},\ A \in \mathbb{R}^{n \times m}\ \wedge\ \mathbf{b} \in \mathbb{R}^n\}$
polyhedral cone	$\{\mathbf{x} \in \mathbb{R}^m : A\mathbf{x} \leq \mathbf{0},\ A \in \mathbb{R}^{n \times m}\}$
polytope	collection of points bounded by m-dimensional hyperplanes (a generalization of a triangle in 2D)
simplex	form of a polytope, a collection of points in $\mathbb{R}^m$ enclosed by $(m+1)$ $(m-1)$-dimensional hyperplanes

Abbreviations

iff	if and only if
df	degrees of freedom
cnd	conditionally negative definite
cpd	conditionally positive definite
nd	negative definite
nsd	negative semidefinite
pd	positive definite
pdf	probability density function
psd	positive semidefinite
wrt	with respect to
k-CDD	k-Centres Data Description
k-NN	k-Nearest Neighbor rule
NN	Nearest Neighbors
k-NNDD	k-Nearest Neighbor Data Description
AL	Average Linkage
CCA	Curvilinear Component Analysis
CH	Compactness Hypothesis
CL	Complete Linkage
CNN	Condensed Nearest Neighbor
CS	Classical Scaling
DR	Dissimilarity representation
GNMC	Generalized Nearest Mean Classifier
GMDD	Generalized Mean Data Description
ID	Intrinsic dimensionality
LC	Linear Classifier
LLE	Locally Linear Embedding
LP	Linear Programming
LPDD	Linear Programming Dissimilarity data Description
MDS	Multidimensional Scaling
MST	Minimum Spanning Tree
NLC	Normal density based Linear Classifier
NMC	Nearest Mean Classifier
NQC	Normal density based Quadratic Classifier
NN	Nearest Neighbor rule
PCA	Principal Component Analysis
PR	Pattern Recognition
RKHS	Reproducing Kernel Hilbert Space
RKKS	Reproducing Kernel Kreĭn Space
RNLC	Requlatized Normal density based Linear Classifier
RNQC	Requlatized Normal density based Quadratic Classifier
QC	Quadratic Classifier
QP	Quadratic Programming
SL	Single Linkage
SRQC	Strongly Requlatized Quadratic Classifier
SV	Support Vector
SVM	Support Vector Machine
SVDD	Support Vector Data Description
SO	Support Object

Two roads diverged in a yellow wood,
And sorry I could not travel both
And be one traveler, long I stood
And looked down one as far as I could
To where it bent in the undergrowth;

Then took the other, as just as fair,
And having perhaps the better claim,
Because it was grassy and wanted wear;
Though as for that the passing there
Had worn them really about the same,

And both that morning equally lay
In leaves no step had trodden black.
Oh, I kept the first for another day!
Yet knowing how way leads on to way,
I doubted if I should ever come back.

I shall be telling this with a sigh
Somewhere ages and ages hence:
Two roads diverged in a wood, and I –
I took the one less traveled by
And that has made all the difference.

”THE ROAD LESS TRAVELED”, ROBERT FROST

1. Introduction

Every beginning
is only a sequel, after all,
and the book of events
is always open halfway through.

"LOVE AT FIRST SIGHT", WISŁAWA SZYMBORSKA

Human perception and inference skills allow us to recognize what the common characteristics of a collection of objects are. It might be, however, difficult to formalize such observations. Imagine, for instance, a set of fish shape contours [127], as presented in Fig. 1.1. Is it possible to define a simple rule that divides them into two or three groups? If we look at the contours, we will find out that some of them are rather long and without characteristic fins (shape C, H and I), whereas others have distinctive tails as well as fins, say a group of fin-type fish. Judging shapes F and K in the context of all fish shapes presented here, they could be found similar to other fin-type fish: A, B, D, E, G and J. yet by visual inspection, they do not really appear alike, as they seem to be thinner and somewhat larger. If the examples of C, H and I had been absent, the differences between F and K and other fin-type fish would have been more pronounced. Furthermore, shape A could be considered similar to F and K, but also different due to the position and shape of its tail and fins.

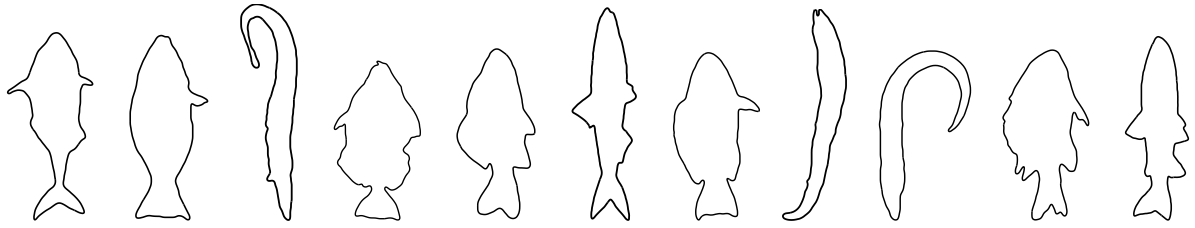


Fig. 1.1: Fish contours.

This simple example shows that without any extra knowledge or a clear context, one cannot claim that the identification of two groups is better than the identification of three groups. This decision relies on a free interpretation of what makes objects similar to be considered as a group.

For the purpose of an automatic grouping or identification, it is difficult to determine proper features, i.e. mathematically encoded particular properties of the shapes, that would precisely discriminate between different fish, yet emphasize the similarity between resembling examples. An alternative is to compare the shapes by matching them as well as possible and determining the remaining differences. Such a match is found with respect to a specified measure of dissimilarity. This measure should take on small values for objects that are alike and large values for distinct objects.

There are many ways of comparing two objects, hence there are many dissimilarity measures. In general, a suitability of a measure depends on the problem at hand and should rely on some additional knowledge one has about this problem. Example measures are presented in Fig. 1.2, where two fish shapes are compared. Here, the dissimilarity between two similar fish, A and B, is much smaller than between two different fish, B and H. Which to choose depends on expert knowledge or problem characteristics. If there is no clear preference for one measure over the other, a number of measures can be studied and combined. This may be beneficial, especially when different measures focus on different aspects of the patterns.

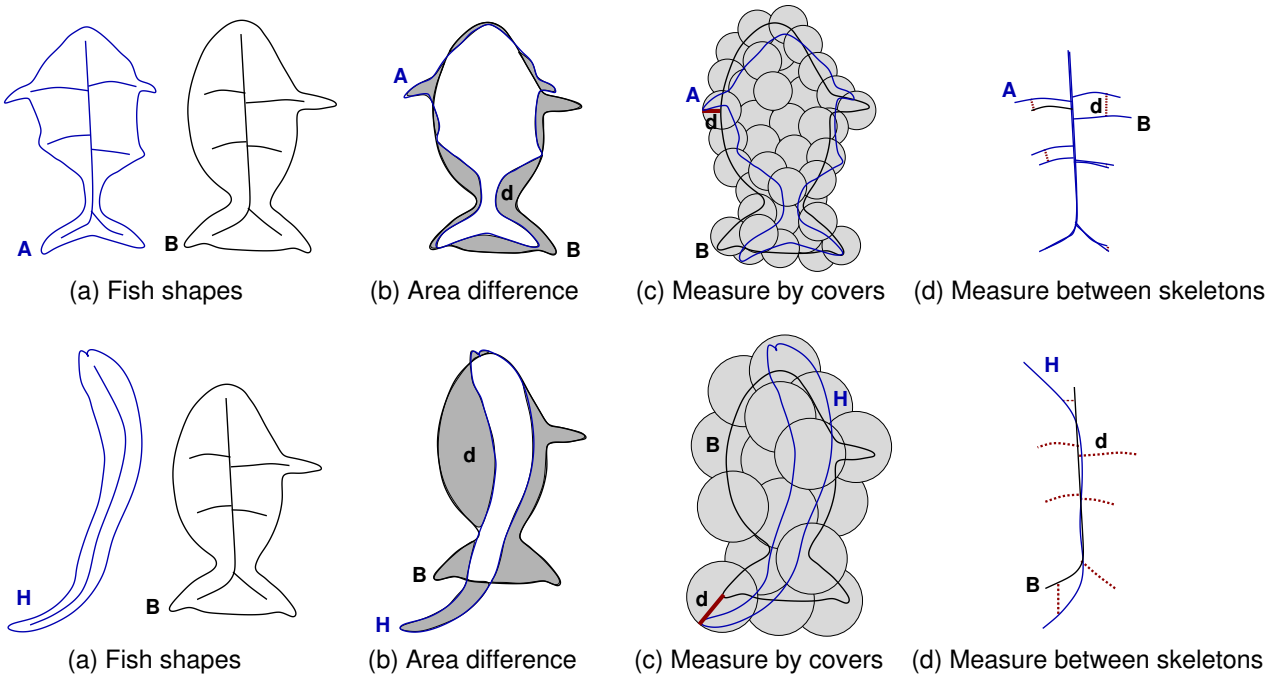


Fig. 1.2: Various dissimilarity measures can be constructed for matching two fish shapes. (b) Area difference: the area of non-overlapping parts is computed. To avoid scale dependency, the measured difference can be expressed relative to the sum of the areas of the shapes. (c) Measure by covers: one shape is covered by identical balls (such that the ball centers belong to it), taking care that the other shape is covered as well. The shapes are exchanged and the radius of the minimal ball is the sought distance. In both cases above, B is covered such that either A or H are also covered. (d) Measure between skeletons: two shape skeletons are compared by summing up the differences between corresponding parts, weighting missing correspondences more heavily.

1.1 Learning from examples

The question how to extract essential knowledge and represent it in a formal way such that a machine could 'learn' a concept of a category, identify objects or classify them, intrigued and provoked many researches. The growing interest inherently led to the establishment of the areas of pattern recognition (PR), machine learning (ML) and artificial intelligence (AI). Researchers in these disciplines try to find ways to mimic the human capacity of using knowledge in an intelligent way. In particular, they try to provide mathematical foundations and develop models and methods that automate the recognition processes by learning from a set of examples. This attempt is inspired by the human ability to recognize e.g. what a tree is, given just a few examples of trees. The idea is that a few examples of objects (and possible relations between them) might be sufficient for extracting suitable knowledge to characterize their class.

After years of research, some practical problems can be now successfully treated in industrial process tasks such as an automatic recognition of damaged products on a conveyor belt, or to fasten the data-handling procedures, like an automatic identification of a person by his/her fingerprint. Yet even after many years of research, the algorithms developed so far are still far from reaching human recognition performance. Although the designed models become more and more complex, yet it seems that to make a step further, one needs to analyze their basic underlying assumptions. An understanding of the recognition process is needed; not only of the learning approaches (inductive or deductive principles), but mainly of the basic notions of class, measurement process and the representation of objects derived from these. The formalized representation of objects (usually in mathematical terms) and the definition of class determine how the act of learning should be mod-

Table 1.1: Basic differences between statistical and structural Pattern Recognition. Distances are a common factor used for discrimination in both approaches.

Properties	Statistical	Structural
Foundation	Well–developed theory of vector spaces	Intuitively appealing: human cognition or perception
Approach	Quantitative	Qualitative: structural and syntactic
Descriptors	Numerical features: vectors of a fixed length	Morphological primitives of a variable size
Characterization	The element position in a vector	The encoding process of primitives
Noise	Easily encoded	Needs regular structures
Learning	Vector-based methods	Based on graphs, decisions trees and grammars
Dissimilarity	Usually a metric distance, often Euclidean	Defined in the matching process
Discrimination	Relies on **distances** or inner products in a feature space	Grammars recognize the membership of valid objects; **distances** often used
Class overlap	Due to improper features and probabilistic models	Due to improper primitives leading to ambiguity in the description

eled. While many researchers are concerned with various algorithmic procedures, we would like to focus on *the issue of representation*. This work is devoted to particular representations, namely dissimilarity representations. Below and in the subsequent sections, we will give some insight into the nature of basic problems in pattern recognition and machine learning, the motivation for the use of dissimilarity representations and the contribution of this dissertation.

While dealing with entities to be compared, we will always refer to them as to objects, elements or instances, regardless of whether they are real or abstract. For instance, images, textures or shapes are called objects, in the same way as apples or chairs. An appropriate representation of objects is based on some data. These are usually obtained by some measurement devices and encoded in a numerical way or given by a set of observations or dependencies, presented in some structural form, e.g. a relational graph. It is assumed that objects can, in general, be grouped together. One hopes to identify a number of groups (clusters) whose existence would support the understanding of not only the data, but also the problem itself. Such a process often serves the purpose of ordering the information and finding suitable or efficient descriptions of the data.

The challenge of automatic object recognition is to develop computer methods which learn to identify whether an object belongs to a specific class or which learn to distinguish between a number of classes. Typically, the system is first presented with a set of labeled objects, the training set, in some convenient representation. Learning consists of finding the class descriptions such that the system can correctly classify novel examples. In practice, the entire system is trained such that the given examples are (mostly) assigned to the correct class. The underlying assumption is that the training examples are representative and sufficient for the problem at hand. This implies that the system can extrapolate well to previously unseen examples, that is it can *generalize* well.

There are two principal directions in PR, statistical and structural [49, 210, 280]. The basic differences are summarized in Table 1.1. Both use features to describe objects, yet they are defined differently. In general, features are functions of the (possibly preprocessed) measurements performed on objects, e.g. particular groups of bits in a binary image summarizing it in a discriminative way. The statistical, i.e. decision-theoretical approach, is (usually) metric and quantitative, while the structural approach is qualitative [49, 280]. This means that in the statistical approach, features are encoded as purely numerical variables. Together these constitute a feature space, usually Euclidean, in which each object is represented as a vector. Learning is then inherently restricted to the mathematical methods that one can apply in such a space. On the contrary, the structural approach tries to describe the structure of the objects that in some intuitive way corresponds to human perception of them [113, 115]. The features become primitives, fundamental structural elements, like strokes,

corners or other morphological elements. Next, the primitives are encoded as syntactic units from which objects are constructed. The strength of the statistical approach relies on the well-developed concepts and learning techniques, while in the structural approach, it is much easier to encode existing knowledge on the objects. See chapter 4 for a more elaborate discussion.

Although our research is grounded in the statistical PR, we recognize the necessity of combining numerical and structural information. Dissimilarity measures as the common factor used for discrimination, Table 1.1, seem to be the natural bridge between these two types.

1.2 Motivation for the use of dissimilarity representations

The notion of similarity plays a pivotal role in class formation, since it might be seen as a natural connection between observations on objects and a judgment on their common nature shared properties. In essence, similar objects can be grouped together to form a class, consequently *a class is a set of similar objects*. There does not exist, however, some general object similarity that can be universally measured or applied. A comparison of two objects is always with respect to a frame of reference, i.e. a particular point of view, context, basic characteristics, type of domain or attributes considered (see also Fig. 1.1). This means that background information, or the existence of other classes will influence the way objects are compared. For instance, two brothers may not appear to resemble each other. However, they may appear much more alike if compared in the presence of their parents. The degree of similarity between two objects should be determined *relative* to a given context or a procedure.

Any measurement of similarity of objects will be based on some assumptions concerning the properties of their relation. Such assumptions come from some model. Similarity can be modeled by a measure of similarity or dissimilarity. These are intimately connected; a small dissimilarity and a large similarity both imply a close resemblance of objects. There exist some ways of changing a similarity value into a dissimilarity value and vice versa, but the interpretation of the measure might be affected. In our work, we mostly concentrate on dissimilarities, which by their construction, focus on the class and object differences. A choice for dissimilarities is supported by the fact that they can be interpreted as distances in suitable vector spaces and in many cases they may be more intuitively appealing. Therefore, we will be mostly concerned with dissimilarities.

In statistical PR, objects are usually encoded by feature values. A feature is a conjunction of measured values for a particular attribute. For instance, if weight is an attribute for the class of apples, then a feature consists of the measured weights for a number of apples. For a set T of N objects, a feature-based representation relying on a set $\mathcal{F}$ of m features is then expressed as an $N \times m$ matrix $A(T, \mathcal{F})$, where each row is a vector describing the feature values for a particular object. Features $\mathcal{F}$ are usually interpreted in a Euclidean feature vector space equipped with the Euclidean metric. This is motivated by the algebraic structure (defined by operations on vectors) being consistent with the geometric (topological) structure defined by the Euclidean distance (which is then defined by the norm). Then all traditional mathematical concepts and methods, such as continuity, convergence or differentiation are applicable. The continuity of algebraic operations makes sure that the local geometry (defined by the Euclidean distance) is preserved throughout the space [224, 278]. Discrimination techniques operating in vector spaces make use of their homogeneity and other properties. Consequently, such spaces require that all the features are treated, up to scaling, in the same way. Moreover, there is no possibility to relate the learning to the geometry defined between the raw representations of the training examples. The geometry is simply *imposed* beforehand by the nature of the Euclidean distance between (reduced) descriptions of objects, i.e. between vectors in a Euclidean space; see also Fig. 1.3. The existence of a well–established theory for Euclidean metric spaces made researchers place the learning paradigm in that context. However, the severe restric-

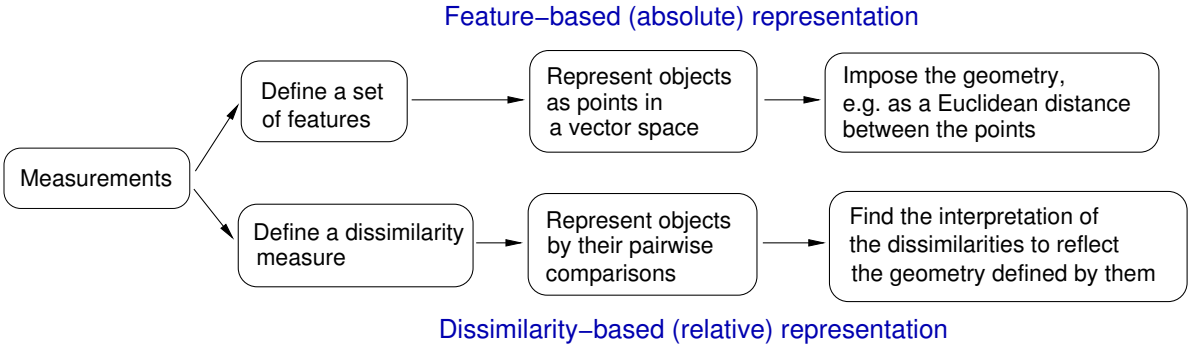


Fig. 1.3: The difference with respect to the geometry between the traditional feature-based representations and dissimilarity-based representations.

tions of such spaces simply do not allow the discovery of structures richer than affine subspaces. From this point of view, the act of learning is very limited.

We argue here that the notion of proximity (similarity or dissimilarity) is more fundamental than that of a feature or a class. According to an intuitive definition of a class (i.e. a set of similar objects), proximity plays a crucial role for class constitution, and not features, which may (or may not) come later. From this point of view, features might be a superfluous step in the description of a class. Surely, proximity might be based on e.g. a weighted combination of features, but the definition of the features should be influenced by the way the proximity of objects will be judged. On the other hand, proximity between objects can be directly found on raw or pre-processed measurements like images. Moreover, in the case of symbolic objects, graphs or grammars, the determination of numerical features might be an intractable problem, while a proximity can be easily defined. This emphasizes that a class of objects is represented by individual examples which are judged similar according to a specified measure. A dissimilarity representation of objects is then based on pairwise comparisons and is expressed e.g. as an $N \times N$ dissimilarity matrix $D(T,T)$, where each entry corresponds to a dissimilarity between pairs of objects; see also Fig. 1.4. Hence, each object x is represented by a vector of proximities $D(x,T)$ to the objects in T (precise definitions will be given in chapter 4). A new example z, represented by $D(z,T)$, is classified to a specific class if it is sufficiently similar to one or more objects within that class.

For a number of years, Goldfarb and colleagues have been trying to establish a new mathematical formalism, which would allow one to describe objects from a metaphysical point of view, that is to learn their structure and characteristics in a process of their construction; see e.g. [153–158]. This projects aims at unifying the geometric learning models (statistical approach with the geometry imposed by a feature space) and symbolic ones (structural approach) using dissimilarity as a natural bridge. A dissimilarity measure is determined in a process of inductive learning realized by so-called evolving transformation systems [153, 157, 160]. Loosely speaking, such a system is composed of a set of primitive structures, basic operations that transform one object into another (or which generate a particular object) and some composition rules which permit the construction of new operations from the existing ones [155–157, 160, 161]. This is the symbolic component of the integrated model. The geometric component is defined by means of a dissimilarity. Since there is a cost connected to each operation, the dissimilarity is determined by the minimal sum of the costs of operations transforming one object into another (or generating this particular object). In this sense, the operations play the role of features and the dissimilarity, dynamically learned in the training process, combines the objects into a class.

In this dissertation, the study of dissimilarity representations has mainly an epistemological character. It focuses on *how* we decide (how we make a model to decide) that an entity belongs to a particular class. Since such a decision builds upon the dissimilarities, we come closer to the na-

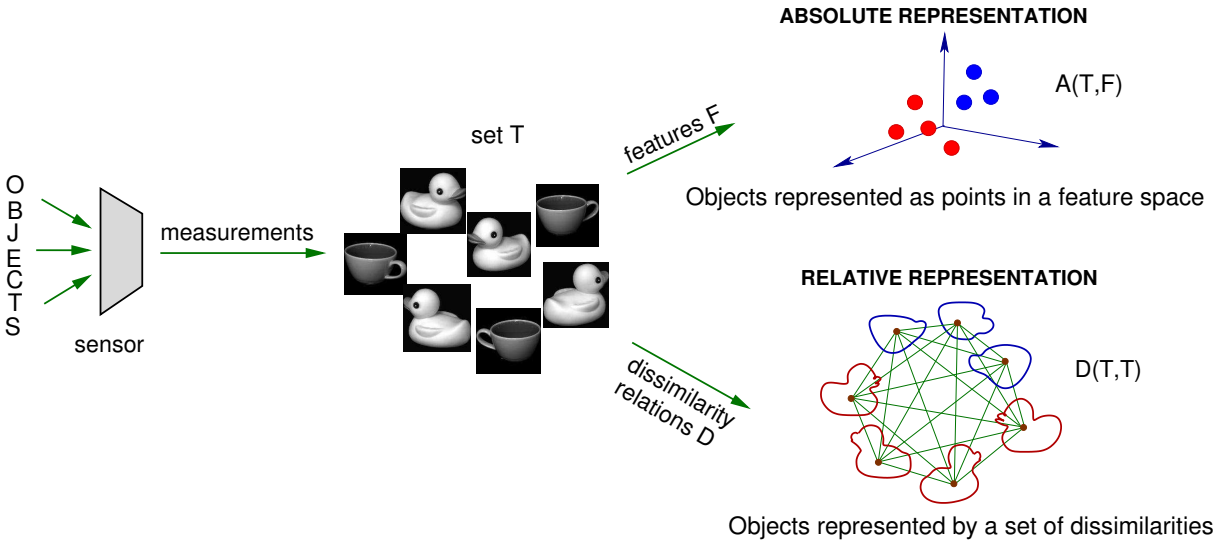


Fig. 1.4: Feature-based (absolute) representation and dissimilarity-based (relative) representation.

ture of *what* a class is, as, we think, it is proximity which defines the class. This approach is much more flexible than the one based on features, since now, the geometry and the structure of a class is defined by the dissimilarity measure, which can reflect the structure of the objects in some space. Note that the reverse holds in a feature space, i.e. a feature space determines the (Euclidean) distance measure, hence the geometry; see also Fig. 1.3. Although, dissimilarity information is further treated in a numerical way, The development of statistical methods dealing with general dissimilarities is the first necessary step towards a unified learning model, as the dissimilarity measure may be developed in a structural approach.

This integrated model may be constructed for objects containing an inherent, identifiable structure or organization, like apples, shapes, spectra, text excerpts etc, yet current research is far from being of a general applicability [160–162, 223]. On the other hand, there are a number of instances or events which are mainly characterized by discontinuous numerical or categorical information, e.g. gender or number of children etc. Therefore, we may have to consider heterogeneous types of information to support decisions in medicine, finance, etc. In such cases, the symbolic learning model cannot be directly utilized, yet a dissimilarity can be defined. This emphasizes the importance of techniques operating on general dissimilarities. The study of proximity representations is the necessary foundation to depart for the journey into alternative inductive learning methodologies, in which the proximity measure, hence a class description will be learned from examples. This is expected to become a part of future research.

1.3 Outline of the thesis

Dissimilarities play a key role in the quest for the integrated statistical–structural learning model, since they are a natural common factor underlying these two approaches, as explained in the previous sections. This is supported by the theory that (dis)similarities can be considered a connection between perception and higher–level knowledge, a crucial factor in the process of human recognition and categorization [115, 166, 418].

Throughout this thesis, all our investigations are devoted to dissimilarity (or similarity) representations. Note, however, that we do not design new measures. Instead, the goal of our work is to study both the methodology and the approaches to learning from dissimilarity representations. We propose novel and advantageous methods, dealing with classification problems in particular. An outline of the thesis is presented in Fig.1.5.

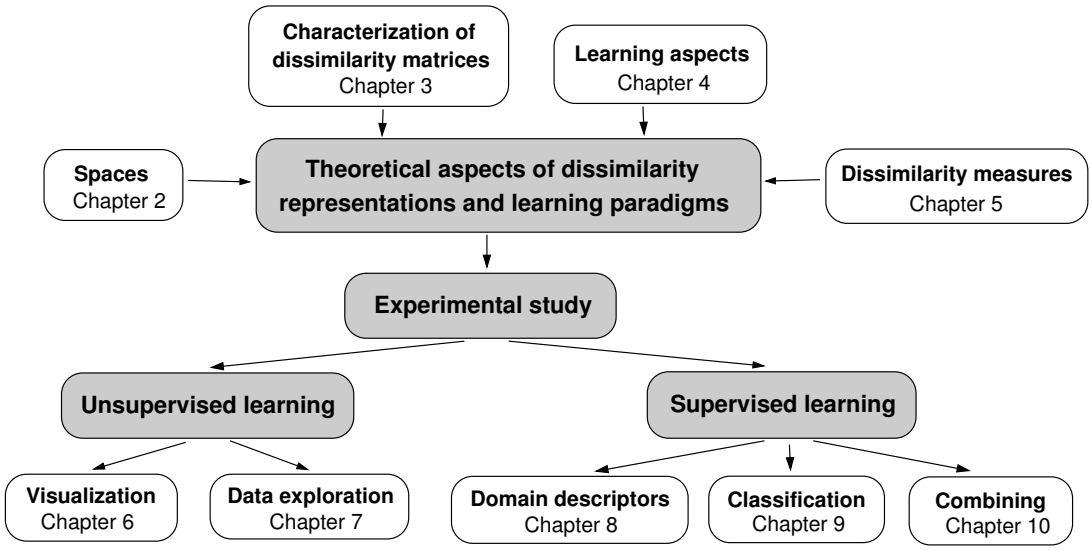


Fig. 1.5: Conceptual outline of the thesis.

The concept of a vector space is fundamental to dissimilarity representations. The dissimilarity value captures the notion of closeness between two objects, which can be interpreted as a *distance* in a suitable space. Since we choose to interpret dissimilarity representations in some vector spaces, our learning methods will ultimately reside there. Therefore, chapter 2 focuses on mathematical characteristics of various spaces, among others (generalized) metric spaces, norm spaces and inner product spaces. These spaces serve later as the context, in which the dissimilarities are interpreted and learning algorithms are designed. Therefore, the understanding of such spaces and their interrelations is needed for a further understanding of learning processes.

In chapter 3, some fundamental issues of dissimilarity measures and generalized metric spaces are discussed. Since a metric distance, particularly the Euclidean distance, is mainly used in statistical learning, its special role is explained and some related theorems are given. The properties of dissimilarity matrices are studied, together with some embeddings, i.e. spatial representations (as vectors in some space found such that the dissimilarities are preserved) of symmetric dissimilarity matrices. This supports the analysis of pairwise dissimilarity data $D(T, T)$ based on a set of examples T.

Chapter 4 starts with a brief introduction to feature-based statistical learning. Then a more detailed description of dissimilarity representations is given. Some discussion on the issue of representation can be found in our publication [105]. The chapter further focuses on possible methods of building classifiers for such representations. Three different approaches are considered. The first one uses dissimilarity values directly by interpreting them as neighborhood relations. The second one interprets them in a space where each dimension is a dissimilarity to a particular object. Finally, the third one relies on embedding (projection algorithms) and building classifiers in the resulting space.

In chapter 5, various types of similarity and dissimilarity measures are described, together with their basic properties. The chapter ends with a brief overview of a number of dissimilarity measures arising from various applications.

Chapters 6 and 7 start from fundamental questions related to exploratory data analysis on dissimilarity data. One of the most basic and crucial points, supporting the process of understanding the relations between data instances, is data visualization. This is discussed in chapter 6. Some of the issues presented are based on the reports that resulted from a project in cooperation with Shell E&P [297–299], a conference article [302] and a book chapter [289]. Other issues related to data exploration and understanding are presented in chapter 7. They reflect upon intrinsic dimensionality of the dissimilarity data or complexity of the data description, information on possible clusters

etc. In other words, this chapter focuses on methods of unsupervised learning, where among others grouping methods are discussed. This chapter has grown from an article [102].

One of the tasks in data analysis is the detection of outliers, i.e. objects with invalid measurements or very specific and rare objects of a class. The removal of such objects is expected to improve the coherence of a class description, so a better model can be designed. Some problems are also naturally characterized by the existence of positive (target) examples, say healthy people, and negative examples, say diseased people, where the occurrence of an illness is a rare event. In such cases, the data are highly unbalanced, yielding a relatively small group of negative instances. A possible approach to such problems is by constructing a domain descriptor. Construction of such descriptors for dissimilarity representations is the topic of chapter 8. Some of these methods have been described in earlier works [301, 306].

Chapter 9 is devoted to classification. Three approaches to learning are examined for some artificial and real data. For recognition, the so-called representation set can be used instead of a complete set of training objects. How to select such a set out of a training set and what the advantages and drawbacks of the approaches are, is discussed. This work is supported by the articles [108, 109] and our earlier publications [103, 106, 290, 291, 293–296, 300, 301, 315].

Chapter 10 discusses some possibilities of combining. This might be achieved e.g. by the combination of either different dissimilarity representations or of different types of classifiers. Additionally, some issues concerning conceptual dissimilarity representations resulting from combining classifiers, one-class classifiers or weak models are briefly discussed. The presented material is based on [235, 236, 292, 303–305].

The overall conclusions and recommendations are summarized in chapter 11.

1.4 Main contributions

In this work, we propose the use of dissimilarity representations for identification and recognition purposes. These are especially advantageous in the following cases:

- for sensor data, such as spectra, digital images, shapes etc;
- when the information on objects is encoded in a structural way, e.g. by trees or strings;
- when vector representations of objects live in a high-dimensional space;
- when the features describing objects are of mixed types;
- as a way of constructing nonlinear classifiers in given feature spaces.

We establish some mathematical foundations for approaching learning problems based on algebra [26, 177, 285], operator theory [112, 327], functional analysis [225, 234], indefinite inner product spaces [3, 34, 204] and general topology [62, 224, 278, 363, 376, 379, 419] as well as the results of Vapnik [403], Schölkopf [345, 346, 348, 350, 351, 356], Goldfarb [151, 152, 154, 163] and other researchers. We present a systematic approach to study dissimilarity representations, which is the principal aim of this work. We propose some novel procedures to learning from such representations, inevitably compared to the nearest neighbor method (NN) [71], which is the one traditionally applied in this context. To our knowledge, although researchers have thoroughly studied the NN method and its variants together with a design of perfect dissimilarity measures (appropriate to the character of the NN rule), little attention has been devoted to alternative approaches. Only recently, in the machine learning community, has the interest arisen for the support vector machines. However, these methods rely on a relatively narrow class of (conditionally) positive definite kernels, which, in our terminology, can be seen as similarity representations [108, 109]. Our methods, on the contrary, are applicable to general (dis)similarity representations and this is where our main contribution is achieved. A more detailed description of the overall contributions is presented below.

Representation of objects. The proximity representation quantitatively encodes the proximity between pairs of objects. It relies on the representation set R, a relatively small collection of objects capturing the variability in the data. Each object is described by a vector of proximities to R. In the beginning, the representation set may consist of all training examples and reduced later in the process of instance selection. Some selection criteria have been proposed and experimentally investigated for different learning frameworks. As such, proximity representations have been developed by us as a first step towards bridging the statistical and structural approaches to pattern recognition. They are successfully used for solving object recognition problems.

Data understanding. To understand data is a difficult task. The main consideration is whether the data sampling is sufficient to describe the problem domain well. Other important questions refer to intrinsic dimensionality, data structure, e.g. in terms of possible clusters and the means of data visualization. Since there exist many algorithms for unsupervised learning, our primal interest lies in the former questions.

In this thesis, three distinct approaches to operate on dissimilarity representations have been proposed. The first one relies on an approximate embedding of dissimilarities into a (pseudo-)Euclidean space. The second approach addresses a dissimilarity representation as a mapping based on the representation set R. As a result, the so-called dissimilarity space is considered, where each dimension corresponds to a dissimilarity to a particular object from R. The third one operates on the given dissimilarities directly. The approaches are introduced, studied and applied in various situations.

Domain description. The problem of describing a class has recently gained a lot of attention, since it can be identified in many applications. The area of interest covers all problems where the specified targets have to be recognized and the anomalies or outlier situations have to be detected. These might be examples of any type of fault detection, abnormal behavior, or rare diseases. The basic assumption that an object belongs to a class if it is similar to examples within this class. The identification procedure can be realized by a proximity function equipped with a threshold, determining whether an instance is a class member or not. This proximity function can be e.g. a distance to a set of selected prototypes. Therefore, the data represented by proximities is more natural for building the concept descriptors, since the proximity function can be directly built on them.

To study this problem, not only some known algorithms have been adopted for dissimilarity representations, but also new methods have been implemented and investigated. Concerning both the efficiency and the performance issues, our methods were found to perform well.

Classification. New methodologies to deal with dissimilarity/similarity data have been proposed. These rely either on the approximate embedding in a pseudo-Euclidean space and constructing the classifiers there or on building the decision rules in a dissimilarity space, or on designing neighborhood-based classifiers, as e.g. the NN rule. In all cases, some foundations have been established, which allow us to handle general dissimilarity measures. Our methods do not require metric constraints, so their applicability is quite universal.

Combining. A possibility to combine various type of information has proved to be useful in practical applications; see e.g. [266, 267]. We argue that combining either significantly different dissimilarity representations or combining classifiers different in nature on the same representation can be beneficial for learning. This may be useful when there is lack of expertise how a well-discrimination dissimilarity measure should be designed. A few measures can be considered, taking into account different characteristics of the data. For instance, when scanned digits should be compared, one measure can focus on the contour information, others on the area or some statistical properties.

Applications. A proximity measure plays an important role in many research problems. Proximity representations have already been used, although indirectly, in many areas. They serve the purpose

of text or image retrieval, data visualization, the learning process from partially labeled sets, etc. Here, a number of other applications is presented, where such measures have been found to be advantageous.

Credits. This thesis contains work that has been published or submitted before. Robert Duin, Carmen Lai, Pavel Paclík, Dick de Ridder, Marina Skurichina and David Tax are acknowledged for the discussions on all types of pattern recognition and machine learning issues, which resulted in common publications. We are also grateful for some dissimilarity data sets to Douglas Zongker, prof. Anil Jain, Simon Günter, prof. Horst Bunke, Volker Roth, Pavel Paclík and Thomas Landgrebe. All the data sets are described in Appendix A.2. Most of the experiments have been conducted using PRTools [101], DD-tools [387] and own routines.

1.5 In summary

> *Progress has not followed a straight ascending line, but a spiral with rhythms of progress and retrogression, of evolution and dissolution.*
>
> JOHANN WOLFGANG VON GOETHE

One of the basic questions in pattern recognition is how to tell the difference between given objects. Two principal approaches can be distinguished to handle this problem. The statistical approach focuses on measuring characteristic numerical features and representing objects as points in a Euclidean feature space. Objects are different if their point representations lie sufficiently far away in this space, which means that the corresponding Euclidean distance is large. The difference between classes of objects is learned by finding a discrimination function in a feature space. It is constructed such that the classes, represented by sets of points, are separated as well as possible.

The structural approach is applicable to objects with some identifiable structural organization. Basic descriptors or primitives, encoded as syntactic units, are then used to characterize objects. Classes of objects are learned either by suitable syntactic grammars or the objects themselves are compared by the cost of some specified match procedure. Such a cost expresses a degree of difference between two objects.

This thesis is concerned with statistical learning methods for dissimilarity representations. These are numerical representations, in which each value captures the degree of commonality between pairs of objects. The goal is to develop and study such learning approaches. Since a dissimilarity measure can be defined on arbitrary data given by collections of sensor measurements, shapes, strings or graphs, or vectors in a feature space, the dissimilarity representation itself becomes very general. The advantages of statistical and structural approaches can be now integrated on this level.

To make the use of statistical learning, dissimilarity representations have to be interpreted in some mathematical frameworks, i.e. in some spaces, in which discrimination functions can be defined. Since general non-Euclidean dissimilarity measures are used in practical applications, a study outside the traditional use of Euclidean spaces was necessary. This led us to more general spaces.

This thesis has some aspects of both mathematical[1] and experimental work. As a result, a necessary trade-off had to be reached to present both theory and practice. Although some foundations are laid down, the work is not completed as it requires years of research to come. We realize that the material may be hard to read due to a variety of issues it discusses. Still, we hope that it will be inspiring and encouraging to think about the presented concepts.

[1] Our theorems, observations and propositions are marked by a star. All the proofs are ours.

PART I

Concepts and theory

”I understand,” said Modi, nodding. ”What I am getting at, however, is the nature of the work. [...] By the nature of your work, I mean...how shall I put it?” He paused, then said, ”Does it deal with aspects of reality perhaps? With areas that go beyond the merely mechanical into zones of, shall we say, more nebulous reality? Where, perhaps, the senses need to be transcended?”

”7 STEPS TO MIDNIGHT”, RICHARD MATHESON

Budowałem na piasku
I zawaliło się.
Budowałem na skale
I zawaliło się.
Teraz budując
Zacznę od dymu z komina.

”Podwaliny”, Leopold Staff

I built on the sand
And it tumbled down.
I built on a rock
And it tumbled down.
Now when I build, I shall begin
With the smoke from the chimney.

”Foundations”, Leopold Staff

2. Spaces

Ring the bells that still can ring
Forget your perfect offering
There is a crack in everything
That's how the light gets in.
"ANTHEM", LEONARD COHEN

The main goal of this thesis is to develop learning methodologies for dissimilarity representations. Although many dissimilarity measures are designed and further used for matching purposes, a general theoretical foundation for learning from examples represented by their dissimilarities to a set of prototype objects is not established yet. Various dissimilarity measures are used for object comparisons in pattern recognition and related fields. Some of the measures are briefly discussed in chapter 5. Different properties of these measures, such as Euclidean behavior, metric or asymmetric properties, may lead to different learning approaches.

Many learning methods exist that make use of (Euclidean) distances in vector spaces. This, however, relies on a feature-based representation of objects, which might not always be feasible to derive for a given problem. Examples are structural descriptions of objects by graphs or strings. The question is, therefore, how a learning task can be performed given a set of examples and their dissimilarity representation. Additionally, sensor measurements or some intermediate description of the considered examples may be also provided.

In order to make use of statistical learning, an appropriate framework for the interpretation of dissimilarity data should be created. The concept of a (vector) space is important for the development of a theoretical foundation, both from representational and algorithmic points of view, since we will rely on numerical procedures and deal with numerical representations of the problems. Dissimilarities quantitatively express the differences between pairs of objects, while learning algorithms usually optimize some error for a chosen numerical model. Dissimilarities have, therefore, a particular meaning within the frame of specified assumptions and models. Spaces possessing different characteristics will allow one for different interpretations of the dissimilarity data, which will lead to different learning algorithms. It is our aim to present various approaches to learning based on properties of various spaces. Therefore, before dissimilarity measures and representations, as well as learning methods are discussed, some essential concepts and properties of spaces are needed.

This chapter is motivated by the lack of a consistent and clearly identified mathematical theory on general dissimilarity measures not only in pattern recognition field, but also in mathematics. In its foundations, such a theory relies on the notion of nearness between two objects. Therefore, the theory of spaces plays a key role, since such a nearness can be easily introduced there. Most of the existing theory deals with norms, which are often used to define metrics. Usually, Euclidean, city block or max-norm distances are considered. Other related issues are spread over various subfields of mathematics, hence they are only partially known in pattern recognition. Our contribution here is to bring together and present a basic theory on spaces in the context of general dissimilarities, both metric and non-metric. The spaces described here will serve as interpretation frameworks of dissimilarity data. The connections will become clear in chapters 3 and 4.

To our knowledge, no book exists yet that explains a theoretical background on general dissimilarity measures and which studies learning problems from such a perspective (although a general study on

pattern theory in this direction was done by Grenander [174–176]). Therefore, this chapter is meant to fill this gap. It not only introduces some spaces with their basic properties, but it also shows the relations between them. Consequently, the concepts are presented from a mathematical point of view and supported, if possible by examples from pattern recognition. This part, although limited, may still seem rather theoretical. Yet, its purpose is clear: to establish the mathematical basis for dissimilarity representations.

In general, a space is a set of elements with an additional structure. From a pattern recognition point of view, a space should posses some particular properties so that a finite representation of objects can be characterized for the learning purpose. This means that some of the characteristics can be induced or/and imposed on the data instances considered. Intuitively, a space should be characterized by some notion of nearness (closeness) between its elements, compatible with the algebraic structure[1] whenever such a structure is available.

A space is often considered to already posses a structure of a high degree, as e.g. linear or metric spaces do have. Usually, more primitive spaces are explained using those high-level concepts. In our case, however, the dissimilarity measure might not satisfy the metric constraints, i.e. reflexivity, definiteness, symmetry and triangle inequality; see Def. 2.30. Therefore, spaces more primitive than metric (or Euclidean) should be considered. A bottom-up approach, starting from e.g. a notion of a neighborhood, is needed. A common approach is either to consider only metric distances or to impose them by a suitable correction of the given dissimilarity measure. (Even stronger, not only metric, but often the Euclidean distance is assumed.) Therefore, in the coming sections 2.1 – 2.4, we will briefly capture the basic concepts of some generalized topological spaces, generalized metric spaces and linear spaces, as well as some of their essential properties. Most of the proofs are omitted as they can be found in standard textbooks.

Now we will briefly mention the spaces to be introduced in the subsequent sections. We start with the notion of a neighborhood[2] (or a closure) which is the basis for the construction of more complex spaces, among others neighborhood spaces, pretopological spaces and topological spaces. For a general illustration of the interrelations between some of the generalized topological spaces, see Fig.2.1. The idea of such a pictorial schema is to present how from a very general space satisfying a few constraints, more specific spaces, possessing more structure, are built. Basically, if one pictorial space is 'encapsulated' by another, it obeys more requirements and possesses more properties than the first one, and, consequently, it is more specific and its structure is richer.

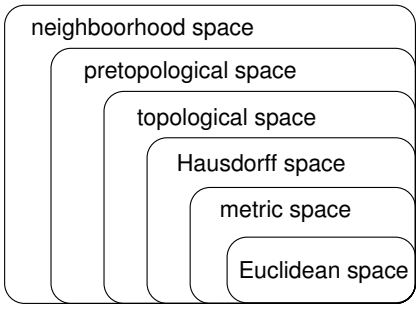
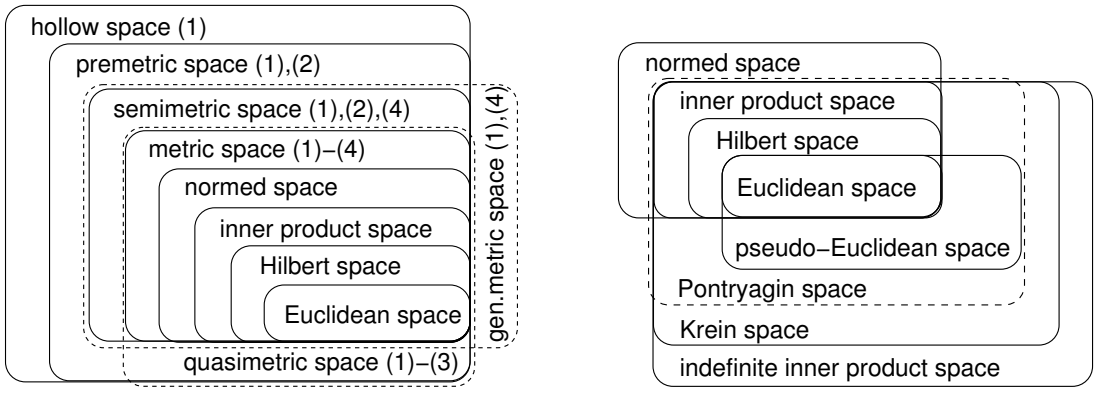


Fig. 2.1: Some generalized topological spaces.

A set with a neighborhood system creates a neighborhood space. Requiring that the intersection of two neighborhoods belongs to the neighborhood system leads to a pretopological space. Adding further the concept of a 'proper' boundary, i.e. an idempotent closure operator, gives rise to a neighborhood basis consisting of open sets. As a results, one gets a topological space. Imposing the existence of disjoint neighborhoods for distinct elements (which implies that the sequences of elements have at most one limit) yields a Hausdorff space. By requiring more and more separation axioms (by the means of topological operations) between disjoint sets and distinct elements, more

[1] For instance, the structure of a vector space is based on the operations of addition and multiplication by a scalar, which e.g. lead to the construction of a linear combination, and consequently to a hyperplane. In a Banach space, algebraic operations are continuous with respect tot he introduced norm.

[2] Even more primitive concepts can be used, like filter, convergence or nearness [62, 141, 375].



(a) Generalized metric spaces (b) Generalized inner product spaces

Fig. 2.2: A schematic diagram of relations between some classes of spaces. The numbers in (a) correspond to the conditions of Def. 2.30. See sections 2.2–2.4 for details.

advanced spaces are obtained, finally leading to a metric space[3]. By providing the Euclidean distance to a vector space, a Euclidean space is obtained. This brief presentation shows that a Euclidean space possesses a structure of a high degree.

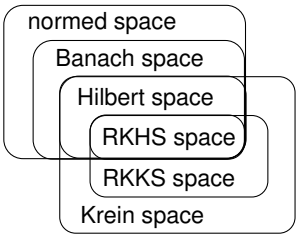


Fig. 2.3: Some inner product spaces. RKHS and RKKS stand for reproducing kernel Hilbert and Kreĭn spaces, respectively.

Having introduced generalized topological spaces, a linear space will be considered as the foundation for more complex spaces. The following spaces are briefly discussed: normed and (indefinite) inner product spaces with their relations to a metric space. Our attention is specifically devoted to Euclidean (Hilbert) and pseudo-Euclidean (Kreĭn) spaces. Since the inner product and metric are essential concepts for the description of relations between object representations, the dependencies between some classes of spaces are considered from these two perspectives; see Fig.2.2 for a schematic diagram. In this pictorial schema, if one space is 'embraced' by another, it is either more restricted or a special case of the first one. For instance, a (finite-dimensional) Euclidean space is a particular case of a Hilbert space, which, in turn, is an inner product space and a special case of a Banach space. The latter is an example of normed spaces, which, if metric is defined, can be considered as metric spaces. If the metric requirements are weakened, then more general spaces, like quasimetric or premetric spaces are obtained. See sections 2.2 – 2.4 for details.

2.1 Generalized topological spaces

I am always doing that which I cannot do, in order that I may learn how to do it.

PABLO PICASSO

Standard textbooks on topology define a topology on a set X by the means of a collection of open sets. Open set is the basic notion of topology. For instance, in application to digital image processing, they are used to construct a new digital topology on the 'integer plane' $\mathbb{Z} \times \mathbb{Z}$; see [214, 215, 221, 222]. When topology is discussed in normed vector spaces, the norm defines a metric distance, which is used to construct open ball neighborhoods $B_\varepsilon(x) = \{y \in X : d(x, y) < \varepsilon\}$, for $\varepsilon > 0$ [278, 327]. These open sets determine the natural topology in metric spaces. The concept of neighborhood is, however, more fundamental than the concept of distance, since a metric (normed) space is already a high-level construction with a high degree of geometric structure; see

[3] This ordering of spaces from extremely general to very specific is by no means unique. One may arrive at metric structure from uniform structure and proximity structure [224, 419].

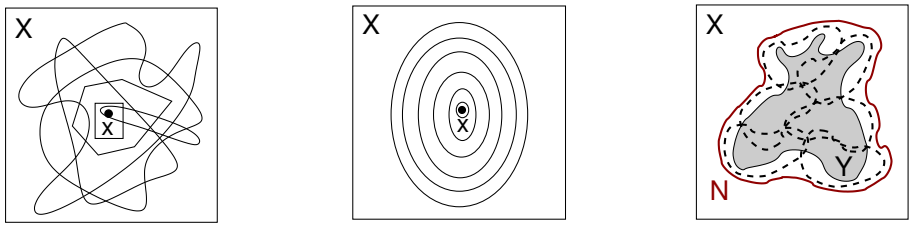


Fig. 2.4: Illustration on neighborhoods. (a) Examples of neighborhoods of x from the set X. (b) A nested neighborhood basis of X. (c) A neighborhood N of a set $Y \subset X$; dashed ovals correspond to neighborhoods of some elements $y \in Y$.

Fig. 2.1 and 2.2. Topology can be derived in a bottom-up way, where the notion of a distance is not yet available. This can be achieved by the use of neighborhoods or generalized closure operators. For an introduction to standard topology, see e.g. [278] and for more general topics, see books of Gastl and Hammer [141], Köthe [224], Sierpiński [363] and Willard [419], as well as the articles [148, 149, 376, 378, 379].

In this thesis we want to point out that neighborhoods or generalized closure operators can be considered as basic concepts to build a (pre)topological space and to express the relations between objects. This can be especially advantageous when one directly works with a representation domain of objects, such as a collection of strings. Since our analysis starts from dissimilarity relations between a set of examples, the neighborhoods will be defined here by the use of dissimilarities in generalized metric spaces.

One of the most crucial characteristics a space should reflect is the notion of nearness, i.e. being able to tell whether two elements are near or not. Note that at the most basic level it might be impossible to distinguish that an element x is nearer to z than to y, although it can be judged that x is near to both y and z. So, the relation of nearness may be based on based on the relations between sets and not yet quantitative. It does not need to be symmetric, i.e. x can be near to y, but not vice versa. (The nearness can also be seen as an asymmetric resemblance relation, e.g. of a child to a parent.) A further study in this direction may lead to the so-called proximity spaces [62, 419].

A possible formalization of the notion of nearness for the set X can be made by defining for each element $x \in X$ a collection of subsets of X, called neighborhoods of x. Intuitively, the basic properties of neighborhoods should be that each element x is contained in all its neighborhoods, any set containing a neighborhood is a neighborhood, so consequently the entire set is the largest neighborhood of each of its points. Below, formal definitions are presented.

Def. 2.1 (Generalized topology via neighborhoods) Let $\mathcal{P}(X)$ be a power set of X, i.e. the set of all subsets of X. The neighborhood function $\mathcal{N}: X \to \mathcal{P}(\mathcal{P}(X))$ assigns to each $x \in X$ the collection $\mathcal{N}(x)$ of all its *neighborhoods* of x such that

(1) Every x belongs to all its neighborhoods: $\forall_{x \in X} \forall_{N \in \mathcal{N}(x)}\ x \in N$.

(2) Any set containing a neighborhood is a neighborhood: $\forall_{N \in \mathcal{N}(x)} \forall_{M \subseteq X}\ (N \subset M \Rightarrow M \in \mathcal{N}(x))$.

(3) The intersection of two neighborhoods is a neighborhood: $\forall_{N,M \in \mathcal{N}(x)}\ N \cap M \in \mathcal{N}(x)$.

(4) For any neighborhood of x, there exists a neighborhood of x that is a neighborhood of each of its elements: $\forall_{N \in \mathcal{N}(x)} \exists_{M \in \mathcal{N}(x)} \forall_{y \in M}\ M \in \mathcal{N}(y)$.

The pair $(X, \mathcal{N})$ with $\mathcal{N}$ satisfying the first two requirements is a *neighborhood space* [141]. The pair $(X, \mathcal{N})$, obeying conditions (1) – (3) is called a *pretopological space*. If all conditions are satisfied, then $(X, \mathcal{N})$ becomes a *topological space*.

Def. 2.2 (Neighborhood basis) A subfamily $\mathcal{N}_B(x)$ of the neighborhood system $\mathcal{N}(x)$ is a *neighborhood basis* (or a local basis) at x if the following conditions are fulfilled:

(1) $\forall_{N\in\mathcal{N}_B(x)}\ x\in N$.

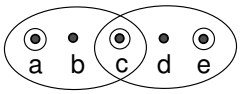
(2) $\forall_{N,N'\in\mathcal{N}_B(x)}\ \exists_{M\in\mathcal{N}_B(x)}\ M\subseteq N\cap N'$.

A neighborhood basis uniquely describes a pretopological space. This follows, since a neighborhood system satisfying the conditions (1) – (3) of Def. 2.1 is built by taking all subsets of X larger than the basis neighborhoods, i.e. $\mathcal{N}(x)=\{M\subseteq X:\exists_{N\in\mathcal{N}_B(x)}\ N\subset M\}$. (Note that a pretopological space may have many bases, each of them capable of describing the entire space.) Therefore, instead of considering a complete neighborhood system, only a neighborhood basis can be used for the definition of pretopology.

Neighborhood systems represent the knowledge on relations between the elements of a set X. In general, a neighborhood of an element x is somewhat similar to x, however its elements are not noticeably distinguishable from x. Therefore, the notion of a neighborhood system provides a general tool for describing relations between elements of X. For instance, neighborhoods can be defined by the use of binary relations, similarity and dissimilarity measures or hierarchic systems. See also Fig. 2.4 for an illustration on neighborhoods.

Example 2.3 (Neighborhood bases)

1. Let $X=\{a,b,c,d,e\}$. The neighborhood basis emphasizes particular relations between the elements. For instance, for the relations on the right side below, it is defined as:
   - $\mathcal{N}_B(a)=\{\{a\},\{a,c\}\}$.
   - $\mathcal{N}_B(b)=\{\{a,b,c\}\}$.
   - $\mathcal{N}_B(c)=\{\{c\},\{a,b,c\},\{c,d,e\}\}$.
   - $\mathcal{N}_B(d)=\{\{c,d,e\}\}$.
   - $\mathcal{N}_B(e)=\{\{e\},\{c,d,e\}\}$.

   Extension of the above neighborhood relations to a set of integers is the Khalimsky line, used to define a digital topology [214, 215].
2. Let $\rho\colon X\times X\to\mathbb{R}_+^0$ be a general dissimilarity measure, Def. 2.38, such that $\rho(x,x)=0$. Then, $B_\delta(x)=\{y\in X:\rho(x,y)<\delta\}$ is a neighborhood of x for $\delta>0$. The neighborhood basis is given as $\mathcal{N}_B(x)=\{B_\varepsilon(x):\varepsilon>0\}$.
3. Let X be a set. A hierarchical clustering (see section 7.1) can be seen as a successive top-down decomposition of X, represented by a tree. The root, corresponding to the complete set, is the largest cluster. Its children nodes point to a decomposition of X into a family of pairwise disjoint clusters. Each cluster can be then further decomposed into smaller clusters until the single elements in the leaves. In this way, sequences of nested clusters are created. Now, a neighborhood of x is a cluster C_h at the level h in the subtree containing the leaf x. Then, $\mathcal{N}_B(x)=\{C_h:x\in C_h\}$. Note that the requirement of disjoint clusters at each level is not essential for the definition of $\mathcal{N}_B(x)$.

Def. 2.4 (Neighborhood of a set) Let $(X,\mathcal{N})$ be a pretopological space and let $Y\subseteq X$. Then N is a neighborhood of Y iff N contains a neighborhood N_y for each $y\in Y$. The neighborhood system for Y is then given by $\mathcal{N}(Y)=\bigcap_{y\in Y}\mathcal{N}(y)$. See also Fig. 2.4(c).

Def. 2.5 (Open and closed sets via neighboorhoods) Let X be a set. $A\subseteq X$ is *open* if it is a neighborhood of each of its elements, i.e. $\forall_{x\in A}\ A\in\mathcal{N}(x)$. A is *closed* if $(X\backslash A)$ is open.

A neighborhood function $\mathcal{N}$ defines a generalized topology on the set X, as given in Def. 2.1. Neighborhoods can be further used to define the generalized interior and closure operators, which may define open and closed sets, the basic concepts in a topological space. Since the properties of the

Table 2.1: Equivalent axioms for the neighborhood and generalized closure operators. Axioms (1) – (3) describe neighborhood spaces, axioms (1) – (4) define pretopological spaces and axioms (1) – (5) define topological spaces.

Properties	Closure A^-	Neighborhood system $\mathcal{N}(x)$
(1)	$\emptyset^- = \emptyset$	$\forall_{x\in X}\, X \in \mathcal{N}(x)$
(2) Expansive	$N \subseteq N^-$	$N \in \mathcal{N}(x) \Rightarrow x \in N$
(3) Monotonic	$A \subseteq B \Rightarrow A^- \subseteq B^-$	$(N \in \mathcal{N}(x) \ \wedge\ N \subset M) \Rightarrow M \in \mathcal{N}(x)$
(4) Sublinear	$(A \cup B)^- \subseteq A^- \cup B^-$	$N, M \in \mathcal{N}(x) \Rightarrow N \cap M \in \mathcal{N}(x)$
(5) Idempotent	$A^{--} = A^-$	$\forall_{N\in\mathcal{N}(x)}\, \exists_{M\in\mathcal{N}(x)}\, \forall_{y\in M}\, M \in \mathcal{N}(y)$

neighborhood, closure and interior functions can be translated into each other, they are equivalent constructions on X. Therefore, a generalized closure can be considered as a principal concept to define other operators on sets [141, 377, 379].

Def. 2.6 (Generalized closure) Let $\mathcal{P}(X)$ be a power set of X. A *generalized closure* is a function $\mathcal{P}(X) \to \mathcal{P}(X)$ that assigns to each $A \subseteq X$ a subset A^- of X such that $\emptyset^- = \emptyset$ and $A \subseteq A^-$.

This generalized closure function is not idempotent, in general, i.e. for $A \subset X$, the condition $A^{--} = A^-$ does not necessarily hold, as required for the topological closure. The interior function and neighborhood system $\mathcal{N}$ can be now defined by the generalized closure.

Def. 2.7 (Generalized interior) Let $\mathcal{P}(X)$ be a power set of X. A *generalized interior* is a function $\mathcal{P}(X) \to \mathcal{P}(X)$ that assigns to each $A \subseteq X$ a subset A° of X such that $A^\circ = X\backslash(X\backslash A)^-$. Equivalently, one has $A^- = X\backslash(X\backslash A)^\circ$.

Def. 2.8 (Neighborhood system) The neighborhood $\mathcal{N} : X \to \mathcal{P}(\mathcal{P}(X))$ is a function which assigns to each $x \in X$ the collection of neighborhoods defined as $\mathcal{N}(x) = \{N \in \mathcal{P}(X) : x \notin (X\backslash N)^-\}$. Equivalently, one can write $x \in N^- \Leftrightarrow (X\backslash N) \notin \mathcal{N}(x)$.

Def. 2.9 (Generalized topology via closure) Let $\mathcal{P}(X)$ be the power set of X. Consider a generalized closure $^-\colon \mathcal{P}(X) \to \mathcal{P}(X)$ with the following properties:

(1) $\emptyset^- = \emptyset$.

(2) Expansive: $\forall_{N\subseteq X}\, N \subseteq N^-$.

(3) Monotonic: $\forall_{N,M\subseteq X}\, N \subseteq M \Rightarrow N^- \subseteq M^-$.

(4) Sublinear: $\forall_{N,M\subseteq X}\, (N \cup M)^- \subseteq N^- \cup M^-$.

(5) Idempotent: $\forall_{N\subseteq X}\, N^{--} = N^-$.

If axioms (1) – (3) are fulfilled, then $(X, ^-)$ is a neighborhood space. If axioms (1) – (4) hold, then $(X, ^-)$ is a pretopological space. If all conditions are satisfied, $(X, ^-)$ defines a topological space; see also Table 2.1.

Corollary* 2.10 Axioms given in Table 2.1 are equivalent.

Proof. Let X be a set. Recall that $A \subseteq B \Leftrightarrow (X\backslash B) \subseteq (X\backslash A)$ holds for any $A, B \subseteq X$. We will make use of Def. 2.8, where the generalized closure is defined by the neighborhood system. This means that $x \in N^- \Leftrightarrow (X\backslash N) \notin \mathcal{N}(x)$. The proof follows.

(1) $\emptyset = \emptyset^- \Leftrightarrow \forall_{x\in X}\, x \notin \emptyset^- \Leftrightarrow \forall_{x\in X}\, x \notin (X\backslash X)^- \Leftrightarrow X \in \mathcal{N}(x)$.

(2) $\Rightarrow$ Let $N \in \mathcal{N}(x)$. Since the generalized closure is expansive, then $(X\backslash N) \subseteq (X\backslash N)^-$. It follows that $X\backslash(X\backslash N)^- \subseteq (X\backslash(X\backslash N)) = N$. Hence, $x \notin (X\backslash N)^- \Leftrightarrow x \in X\backslash(X\backslash N)^- \Rightarrow x \in N$. As $N \in \mathcal{N}(x)$, then by Def. 2.8, $x \notin (X\backslash N)^-$. Hence, we have proved that $N \in \mathcal{N}(x) \Rightarrow x \in N$.

$\Leftarrow$ Let us assume that $N \in \mathcal{N}(x) \Rightarrow x \in N$ holds. Then, by Def. 2.8, we have $x \in N \Rightarrow x \notin (X \backslash N) \Rightarrow (X \backslash N) \notin \mathcal{N}(x) \Leftrightarrow x \in N^{-}$. Consequently, $N \subseteq N^{-}$.

(3) Assume that $N \in \mathcal{N}(x)$ and $N \subseteq M \Leftrightarrow (X \backslash M) \subseteq (X \backslash N)$. Since the generalized closure is monotonic, one has $N \subseteq M \Leftrightarrow (X \backslash M) \subseteq (X \backslash N) \Rightarrow (X \backslash M)^{-} \subseteq (X \backslash N)^{-}$ holds for all $N, M \subseteq X$. Hence, $x \in (X \backslash M)^{-} \Rightarrow x \in (X \backslash N)^{-}$, which is equivalent to stating that $x \notin (X \backslash N)^{-} \Rightarrow x \notin (X \backslash M)^{-}$, which, by Def. 2.8, is equivalent to $N \in \mathcal{N}(x) \Rightarrow M \in \mathcal{N}(x)$. Since $N \in \mathcal{N}(x) \wedge N \subseteq M$, then $M \in \mathcal{N}(x)$.

(4) Let $(N \cup M)^{-} \subseteq N^{-} \cup M^{-}$ hold for all $N, M \subseteq X$. Assume that $N, M \in \mathcal{N}(x)$. Replacing N by $(X \backslash N)$ and M by $(X \backslash M)$, one gets: $((X \backslash N) \cup (X \backslash M))^{-} \subseteq (X \backslash N)^{-} \cup (X \backslash M)^{-}$. Hence $x \in ((X \backslash N) \cup (X \backslash M))^{-} \Rightarrow (x \in (X \backslash N)^{-} \vee x \in (X \backslash M)^{-})$, which is equivalent to $\{x \notin (X \backslash N)^{-} \wedge x \notin (X \backslash M)^{-} \Rightarrow x \notin ((X \backslash N) \cup (X \backslash M))^{-}\}$. Since $N, M \in \mathcal{N}(x)$ and from de Morgan's law $(X \backslash N) \cup (X \backslash M) = X \backslash (N \cap M)$, the latter implication is equivalent to $(N \in \mathcal{N}(x) \wedge M \in \mathcal{N}(x)) \Rightarrow (N \cap M) \in \mathcal{N}(x)$ by Def. 2.8.

(5) Let $N \in \mathcal{N}(x)$. Assume that the generalized closure is idempotent for all subsets of X. Then, $(X \backslash N)^{-} = (X \backslash N)^{--}$. Based on Def. 2.8, we have $N \in \mathcal{N}(x) \Leftrightarrow x \notin (X \backslash N)^{-} \Leftrightarrow x \notin (X \backslash N)^{--} \Leftrightarrow (X \backslash (X \backslash N)^{-}) \in \mathcal{N}(x)$. Let $M = X \backslash (X \backslash N)^{-}$. Then, $M \in \mathcal{N}(x)$ by the reasoning above. For all y, one has $y \in M \Leftrightarrow y \notin (X \backslash M) \Leftrightarrow y \notin (X \backslash N)^{-} \Leftrightarrow y \notin (X \backslash N)^{--} \Leftrightarrow y \notin X \backslash (X \backslash (X \backslash N)^{-})^{-} \Leftrightarrow y \notin (X \backslash M)^{-} \Leftrightarrow M \in \mathcal{N}(y)$, by Def. 2.8. Hence, we have shown that $\forall_{N \in \mathcal{N}(x)} \exists_{M = (X \backslash (X \backslash N)^{-}) \in \mathcal{N}(x)} \forall_{y \in M} M \in \mathcal{N}(y)$. ■

The difference between pretopological and topological spaces lies in the notion of a closure operator. In a topological space, the closure of any set A is closed, $A^{--} = A^{-}$, and the interior of any set is open, $(A^{\circ})^{\circ} = A^{\circ}$. In a pretopological space, this is not necessarily true, so the basis neighborhoods are not open. Here, the closure operator expresses the growth phenomenon, where the composition of several closures results in successive augmentations, i.e. $A \subseteq A^{-} \subseteq A^{--} \subseteq \ldots$.

Example 2.11 (Pretopological and topological spaces)

1. Let X be any set and let $S : X \times X \to \mathcal{P}(X)$ be a symmetric relation, i.e. $S(x, y) = S(y, x)$. Let a generalized closure of $A \subseteq X$ be defined as $A^{-} = \bigcup_{x,y \in A} S(x, y)$. Then $(X, {}^{-})$ is a neighborhood space, since the generalized closure obeys conditions (1) – (3) of Def. 2.9.
2. Let X be a finite set and (X, E) be a directed graph. Let $F(x)$ be a set of the forward neighbors of x, i.e. $F(x) := \{y \in X : (x, y) \in E\}$. Let $A \subseteq X$. It is straightforward to show by axioms of Def. 2.9 that the closure $A^{-} = \bigcup_{x \in A} (F(x) \cup \{x\})$ defines a pretopological space $(X, {}^{-})$.
3. Let $\mathcal{N}_B(x) = \{y \in \mathbb{R} : |x - y| < \varepsilon \wedge \varepsilon > 0\}$. Then $(\mathbb{R}, \mathcal{N}_B)$ defines a topological space.
4. Let $\mathcal{N}_B(x) = \{(a, \infty) : a \in \mathbb{R} \wedge x \in (a, \infty)\}$. Then $(\mathbb{R}, \mathcal{N}_B)$ defines a topological space.

Corollary* 2.12 (Open and closed sets) Let $(X, {}^{-})$ be a neighborhood space defined by the generalized closure, i.e. conditions (1) – (3) of Def. 2.9 hold. $A \subseteq X$ is *open* if $A^{\circ} = A$. A is *closed* if $A^{-} = A$; see also Table 2.2. Therefore, the following holds:

(1) $\forall_{x \in A} A \in \mathcal{N}(x) \Leftrightarrow A = X \backslash (X \backslash A)^{-}$.

(2) $A = A^{\circ} \Leftrightarrow A = X \backslash (X \backslash A)^{-}$.

Proof.

(1) Assume that $\forall_{x \in A} A \in \mathcal{N}(x)$ holds. By Def. 2.8, $\forall_{x \in A} A \in \mathcal{N}(x) \Leftrightarrow \forall_{x \in A} x \notin (X \backslash A)^{-} \Leftrightarrow \forall_{x \in A} x \in X \backslash (X \backslash A)^{-}$. Hence $A = X \backslash (X \backslash A)^{-}$.

(2) $A = A^{\circ} = X \backslash (X \backslash A)^{-}$ by Def. 2.7. ■

Lemma* 2.13 Let $(X, \mathcal{N})$ be a neighborhood space. The assertions below are equivalent:

(1) $\forall_{N \in \mathcal{N}(x)} \exists_{M \in \mathcal{N}(x)} \forall_{y \in M} M \in \mathcal{N}(y)$.

(2) $N \in \mathcal{N}(x) \Leftrightarrow N^{\circ} \in \mathcal{N}(x)$.

Table 2.2: Equivalent definitions of open sets. Note that A is closed, iff $(X \backslash A)$ is open.

$A \subseteq X$	Neighborhood $\mathcal{N}(x)$	Closure A^-	Interior A°
A is open	$\forall_{x \in A}\ A \in \mathcal{N}(x)$	$A = X \backslash (X \backslash A)^-$	$A = A^\circ$

Proof. The proof of Corollary 2.10 (5) shows that $\forall_{N \in \mathcal{N}(x)}\ \exists_{M=(X\backslash(X\backslash N)^-) \in \mathcal{N}(x)}\ \forall_{y \in M}\ M \in \mathcal{N}(y)$. Since $M := N^\circ$ by Def. 2.7, then $N^\circ \in \mathcal{N}(x)$. ■

A collection of open sets containing x constitutes a neighborhood basis in a topological space, which can be proved by Lemma 2.13. Equivalently, since the closure operator is dual to the interior operator, a neighborhood basis in a topological space can be built by a collection of closed sets containing x.

Lemma 2.14 Let $(X, \mathcal{N}_B)$ be a pretopological space. If all neighborhoods of $\mathcal{N}_B$ are open sets or, $\mathcal{N}_B(x) := \{N \subseteq X : x \in N \ \wedge\ N = N^\circ\}$ for all $x \in X$, then $(X, \mathcal{N}_B)$ is a topological space.

Corollary⋆ 2.15 (Closure on neighborhoods) Let $(X, ^-)$ be a neighborhood space. Then the function $\mathcal{P}(X) \to \mathcal{P}(X)$ defined as $\mathrm{gcl}(A) := \{x \in X : \forall_{N \in \mathcal{N}(x)}\ A \cap N \neq \emptyset\}$, is a generalized closure operator. Moreover, $\mathrm{gcl}(A) = A^-$.

Proof. To prove that $\mathrm{gcl}(A) = A^-$ for every $A \subseteq X$, we will equivalently prove that $x \notin \mathrm{gcl}(A) \Leftrightarrow x \notin A^-$ holds for all $x \in X$.

$\Rightarrow$ $x \notin \mathrm{gcl}(A) \Rightarrow \exists_{N \in \mathcal{N}(x)}\ N \cap A = \emptyset$. By Def. 2.8, this is equivalent to $x \notin (X \backslash N)^-\ \wedge\ N \cap A = \emptyset$. Since $N \cap A = \emptyset \Rightarrow A \subseteq X \backslash N$, then by the monotonic property of $^-$, one has $A^- \subseteq (X \backslash N)^-$. Since $x \notin (X \backslash N)^-$, then $x \notin A^-$.

$\Leftarrow$ $x \notin A^- \Rightarrow (X \backslash A) \notin \mathcal{N}(x)$ by Def. 2.8. Since $(X \backslash A) \cap A = \emptyset$, then $x \notin \mathrm{gcl}(A)$. ■

Def. 2.16 (Limit element) Let $(X, \mathcal{N})$ be a neighborhood space. An element $y \in X$ is a *limit* of $A \subseteq X$ iff for every neighborhood $N \in \mathcal{N}(y)$, N intersects $A \backslash \{y\}$. The set of all limits points $\mathrm{der}(A) := \{y \in X : \forall_{N \in \mathcal{N}(y)}\ (A \backslash \{y\}) \cap N \neq \emptyset\}$ is called the *derived* set [363].

Corollary 2.17 In a neighborhood space, $\mathrm{der}(A) \subseteq A^-$. A closed set contains all its limit elements and conversely.

The notions of both convergence and continuity are important in neighborhood spaces. Convergance is usually defined by the use of filters.

Def. 2.18 (Filter and convergance) A *filter* on a set X is a collection $\mathcal{F}$ of subsets of X such that

(1) $\forall_{F \in \mathcal{F}}\ F \neq \emptyset$.

(2) $\forall_{F, F' \in \mathcal{F}}\ \exists_{F'' \in \mathcal{F}}\ F'' \subseteq (F \cap F')$.

(3) $\forall_{F \in \mathcal{F}}\ \forall_{F'}\ F \subseteq F' \Rightarrow F' \in \mathcal{F}$.

Let $(X, \mathcal{N})$ be a neighborhood space. A filter $\mathcal{F}$ converges to $x \in X$, $\mathcal{F} \to x$, if $\forall_{N \in \mathcal{N}(x)}\ \exists_{F \in \mathcal{F}}\ F \subseteq N$.

Def. 2.19 (Continuity of a function) Let f: $(X, \mathcal{N}) \to (Y, \mathcal{M})$ be a function between two neighborhood spaces. f is *continuous* if for each $x \in X$ $\forall_{M \in \mathcal{M}(f(x))}\ \exists_{N \in \mathcal{N}(x)}\ f(N) \subseteq M$.

Theorem 2.20 (On continuous functions) Let f: $(X, \mathcal{N}) \to (Y, \mathcal{M})$ be a function between two neighborhood spaces. The following assertions are equivalent to the continuity of f [150, 278]:

1. For all $x \in X$, $B \in \mathcal{M}(f(x)) \Rightarrow f^{-1}(B) \in \mathcal{N}(x)$.
2. For every set $A \in \mathcal{P}(X)$, $f(A^-) \subseteq (f(A))^-$.
3. For every set $B \in \mathcal{P}(Y)$, $(f^{-1}(B))^- \subseteq f^{-1}(B^-)$.

4. For every set $B \in \mathcal{P}(Y)$, $f^{-1}(B^\circ) \subseteq (f^{-1}(B))^\circ$.

Note that in topological spaces, continuity of a function translates to the fact that the pre-image of an open (closed) set is an open (closed) set.

Def. 2.21 (Cover, compact space)

1. Let X be a set. A collection of subsets $\omega \subseteq X$ is a *cover* of X if $X = \bigcup \omega$. A cover is finite if finitely many sets belong to it. If ω and ω' are covers of X then, ω' is a *subcover* if $\omega' \subset \omega$.
2. A topological space X is *compact* if every open cover has a finite subcover[4].
3. A topological space is *locally compact* if every element has a compact neighborhood.

2.2 Generalized metric spaces

We have just introduced generalized topological spaces defined on sets. The necessity, however, arises to consider sets on which the two operations: addition of elements and scalar multiplication are defined. In particular, *vector spaces* are important. Most of the information presented here can be found in the following books [33, 112, 177, 195, 224, 419].

Def. 2.22 (Vector space) A *vector* (*linear*) space X over Γ ($\mathbb{R}$ or $\mathbb{C}$) is a set of elements, called *vectors*, with the following algebraic structure[5]

1. X is an additive Abelian group, i.e. there is a function $X \times X \to X$, mapping (x, y) to $x + y$ such that the following conditions are satisfied for all $x, y, z \in X$:
   a. associative law: $(x + y) + z = x + (y + z)$.
   b. commutative law: $x + y = y + x$.
   c. the existence of the zero vector θ: $x + \theta = \theta + x = x$.
   d. the existence of an opposite vector $-x$ for each vector x: $x + (-x) = \theta$.
2. There is a mapping $\Gamma \times X \to X$ of (λ, x) to λx such that the following conditions are satisfied for all $x, y \in X$ and all $\lambda, \mu \in \Gamma$:
   a. associative law: $(\lambda\,\mu)\,x = \lambda\,(\mu\,x)$.
   b. distributive laws: $\lambda\,(x + y) = \lambda\,x + \lambda\,y$, and $(\lambda + \mu)\,x = \lambda\,x + \mu\,x$.
   c. the existence of a unit element $1 \in \Gamma$: $1\,x = x$.

Def. 2.23 (Linear combination, span and linear independence) Let X be a vector space over Γ. The vector x is a *linear combination* of $x_1, x_2, \ldots, x_n \in X$ if there exist $\alpha_1, \alpha_2, \ldots, \alpha_n \in \Gamma$ such that $x = \sum_{j=1}^{n} \alpha_j x_j$. The set span$\{x_1, x_2, \ldots, x_n\}$ is the collection of all their linear combinations. A finite set of vectors $x_1, x_2, \ldots, x_n \in X$ is *linearly independent* if $\sum_{j=1}^{n} \alpha_j x_j = 0$ implies that all $\alpha_j = 0$. Otherwise, the set is *linearly dependent*. An infinite set is *linearly independent* if every finite subset is linearly independent.

Def. 2.24 (Basis of a vector space) Let X be a vector space. A set of vectors $B := \{x_i\}$ from X forms a Hamel *basis* of X if B is linearly independent and each vector x is in the span of $F := \{x_j\}$ for some finite subset F of B. The dimension of X is the cardinality of B.

Def. 2.25 (Subspace) A *subspace* V of a vector space X is a subset of X, closed for the operations of vector additions and scalar multiplication.

Example 2.26 Examples of vector spaces:

[4] In a vector space $\mathbb{R}^m$, a set X is compact if it is closed and bounded.

[5] Γ is in fact a field, i.e. a set with the binary operations of addition $+$ and multiplication $*$ such that both $+$ and $*$ are associative and communitative, there exists an additive identity) and a multiplicative identity, different from 0 and for every element, there exist additive na multiplicative inverses and $*$ is distributive over the operation $+$. Usually, Γ is $\mathbb{R}$ or $\mathbb{C}$.

1. Real and complex numbers, $\mathbb{R}$ and $\mathbb{C}$, respectively, with usual operations of scalar addition and multiplication. $\mathbb{R}^m$ and $\mathbb{C}^m$ are m-dimensional vector spaces.
2. All matrices $A^{n\times m}$ with the matrix addition and multiplication by a scalar.
3. The set $\mathcal{F}(\Omega)$ of all functions defined on a closed and bounded set Ω, with the pointwise addition $(f+g)(x)=f(x)+g(x)$ and the scalar multiplication $(cf)(x)=cf(x)$.
4. The set $\mathcal{C}(\Omega)$ of continuous functions on Ω and the set $\mathcal{M}(\Omega)$ of classes of functions measurable in the Lebesgue sense[6] are infinite dimensional vector spaces and subspaces of $\mathcal{F}(\Omega)$.
5. $L_p^{\mathcal{C}}=\{f\in\mathcal{C}(\Omega):(\int_\Omega |f(x)|^p\,dx)^{1/p}<\infty\}$ for $p\geq 1$ is an infinite dimensional vector space and a subspace of $\mathcal{F}(\Omega)$.

Def. 2.27 (Algebraic dual space)[7] Given a vector space X over Γ ($\mathbb{R}$ or $\mathbb{C}$), the *dual* space X^* is a set of all linear functions $f: X\to\Gamma$, called also *linear functionals* and also denoted by $\mathcal{L}(X,\Gamma)$. X^* itself becomes a vector space over Γ under the pointwise addition $(f+g)(x)=f(x)+g(x)$ and scalar multiplication $(cf)(x)=cf(x)$ for all $f,g\in X^*$, $c\in\Gamma$ and $x\in X$.

If X is finite-dimensional, then both X and X^* have the same dimension. Moreover, X is isomorphic[8] to X^*. If X is infinite-dimensional, then the dimensionality of X^* is strictly larger than that of X [224]. The spaces X and X^* are dual with respect to a bilinear function $X^*\times X\to\Gamma$, called a *scalar product*, and denoted as $\langle\cdot,\cdot\rangle$. For instance, if $X=\mathbb{R}^n=X^*$ ($\mathbb{R}^n$ is self-dual), then $\langle x^*,x\rangle=\sum_{i=1}^n x_i^*x_i$ for $x^*\in X^*$ and $x\in X$. Since X is a vector space, then $X^*=\mathcal{L}(X,\Gamma)$. Therefore, for the evaluation functional δ_x, $\delta_x f=f(x)$, one has that $\delta_x f=\langle f,x\rangle$ for $f\in X^*$ and $x\in X$. Any isomorphism $\phi: X\to X^*$ defines a unique non-degenerate bilinear product on X by $\langle x,y\rangle=\phi(x)(y)$ for $x,y\in X$. Now, for the fixed x, $\phi(x): X\to\Gamma$.

Def. 2.28 (Topological vector space) A vector space X over Γ ($\mathbb{R}$ or $\mathbb{C}$) is a *topological vector space* if there exists a neighborhood system $\mathcal{N}$ such that $(X,\mathcal{N})$ is a topological space and the vector space operations of addition $(x,y)\to x+y$ of $X\times X\to X$ and multiplication by a scalar $(\lambda,x)\to\lambda x$ of $\Gamma\times X\to X$ are continuous.

Def. 2.29 (Continuous dual space) The continuous dual $\mathcal{L}_c(X,\Gamma)$ of a topological vector space X is a subspace of the dual space $X^*=\mathcal{L}(X,\Gamma)$ consisting of all continuous linear functionals[9].

Def. 2.30 (Metric space) A *metric* space is a pair (X,d), where X is a set and d is a distance function $d: X\times X\to\mathbb{R}_+^0$ such that the following conditions are fulfilled for all $x,y,z\in X$:

(1) Reflexivity: $d(x,x)=0$.
(2) Symmetry: $d(x,y)=d(y,x)$.
(3) Definiteness: $d(x,y)=0\Rightarrow x=y$.
(4) Triangle inequality: $d(x,y)+d(y,z)\geq d(x,z)$.

For instance, X can be $\mathbb{R}^m$, $\mathbb{Z}^m$, $[a,b]^m$, or a collection of all (bounded) subsets of e.g. $[a,b]^m$. If X is a finite set, e.g. $X\equiv\{x_1,x_2,\ldots,x_n\}$, then d is specified by an $n\times n$ dissimilarity matrix $D=(d_{ij})$, $i,j=1,\ldots,n$ such that $d_{ij}=d(x_i,x_j)$. Consequently, the matrix D is nonnegative, symmetric and has a zero diagonal.

[6] Two functions are in the same equivalence class if they agree almost everywhere, i.e. if they disagree on a set of a measure zero. From now on $\mathcal{M}(\Omega)$ refers to such classes of functions measurable in the Lebesgue sense.

[7] The notion of the dual space is useful for inner product and normed spaces; see section 2.3.

[8] Isomorphism f is a bijective map (one-to-one and onto) such that both f and its inverse f^{-1} are linear maps.

[9] For any finite-dimensional normed vector space (to be defined in section 2.3) or any topological vector space, such as Euclidean space, the continuous dual and the algebraic dual coincide. $\mathcal{L}_c(X)$ is then a normed vector space, where the norm $||f||$ of a continuous linear functional f on X is defined as $||f||=\sup\{|f(x)|:||x||\leq 1\}$.

Example 2.31 Examples of metric spaces:

1. Let X be any set. For $x, y \in X$, the discrete metric on X is given by $d(x,y) = \mathcal{I}(x \neq y)$, where $\mathcal{I}$ is the indicator function. If X is a finite set, then all the pairwise distances can be realized by points lying on an equilateral polytope (an extension of equilateral triangle and an extension of tetrahedron).
2. Metrics in a vector space $\mathbb{R}^m$ (to emphasize that a vector $\mathbf{x} \in \mathbb{R}^m$, we will mark it in bold):
   - $d_p(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^m |x_i - y_i|^p)^{1/p}$ with $p \geq 1$, a general Minkowski distance.
   - $d_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^m |x_i - y_i|$, the city block distance.
   - $d_2 \equiv d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}$, the Euclidean distance.
   - $d_\infty \equiv d_{\max}(\mathbf{x}, \mathbf{y}) = \max_{1 \leq i \leq m} |x_i - y_i|$, the max-norm distance.
3. Let $\mathcal{F}(\Omega)$ be of all functions defined on a bounded and closed set Ω, $\mathcal{C}(\Omega)$ be a set of continuous functions on Ω and $L_p^{\mathcal{C}} = \{f \in \mathcal{C}(\Omega) : (\int_\Omega |f(x)|^p \, dx)^{1/p} < \infty\}$ for $p \geq 1$. Metrics on the space of functions $\mathcal{F}$:
   - $\mathcal{F} := L_p^{\mathcal{C}}$, $d_p(f,g) = \left(\int_\Omega (f(t) - g(t))^p \, dt\right)^{1/p}$.
   - $\mathcal{F} := \mathcal{C}(\Omega)$, $d_\infty(f,g) = \sup_{t \in \Omega} |f(t) - g(t)|$.

Theorem 2.32 (Backward triangle inequality) Let (X,d) be a metric space. Then, for all $x, y, z \in X$, the backward triangle inequality $|d(x,z) - d(y,z)| \leq d(x,y)$ holds.

Theorem 2.33 (Natural topology in metric spaces) Every metric space (X,d) with a suitable neighborhood basis is a topological Hausdorff space.

Proof. Let $B_\varepsilon(x) := \{y \in X : d(x,y) < \varepsilon\}$ be an open ball. To show that (X,d) is a topological space, it is sufficient to prove that the neighborhood basis $\mathcal{N}_B(x) := \{B_\varepsilon(x) : \varepsilon > 0\}$ defines a topology on X. Below, we show that indeed $\mathcal{N}_B(x)$ is a neighborhood basis, i.e. the axioms of Def. 2.2 are fulfilled.

(1) Obviously, $d(x,x) = 0 < \varepsilon$, which means that $x \in B_\varepsilon(x)$ for any $\varepsilon > 0$, so axiom (1) is satisfied.

(2) Consider any $B_\varepsilon(x)$ and $B_\eta(x)$ for $\varepsilon, \eta > 0$. Let $\zeta \leq \min\{\varepsilon, \eta\}$. Then $d(x,y) < \zeta \leq \min\{\varepsilon, \eta\}$, which means that if $y \in B_\zeta$, then $y \in B_\varepsilon(x) \cap B_\eta(x)$, hence the inclusion $B_\zeta(x) \subseteq B_\varepsilon(x) \cap B_\eta(x)$ holds. Therefore, axiom (2) is fulfilled and the ε-balls are the local basis.

By Lemma 2.14 we need to prove that the ε-balls are open sets. By Def. 2.5, $B_\varepsilon(x)$ is an open set iff for all $y \in B_\varepsilon(x) \Rightarrow B_\varepsilon(x) \in \mathcal{N}(y)$. We will first show that for each $y \in B_\varepsilon(x)$ there exists η such that $B_\eta(y) \subseteq B_\varepsilon(x)$. Let $y \in B_\varepsilon(x)$. Let η be such that $0 < \eta \leq \varepsilon - d(x,y)$. Let $z \in B_\eta(y)$. This means that $d(y,z) < \eta \leq \varepsilon - d(x,y)$, which leads to $d(x,y) + d(y,z) < \varepsilon$. By the triangle inequality, we have $d(x,z) < \varepsilon$, which stands for $z \in B_\varepsilon(x)$. Hence, we show that $z \in B_\eta(y) \Rightarrow z \in B_\varepsilon(x)$, hence $B_\eta(y) \subseteq B_\varepsilon(x)$. By axiom (2) of Def. 2.1, any set enclosing $B_\eta(y)$ belongs to $\mathcal{N}(y)$, hence $B_\varepsilon(x) \in \mathcal{N}(y)$ for all $y \in B_\varepsilon(x)$. Therefore, $\mathcal{N}_B(x)$ consists of open sets. Consequently, by Lemma 2.14, $\mathcal{N}_B(x)$ defines a topological space.

The fact that every metric space (X,d) is Hausdorff (see also Def. 5.3) can be shown as follows. Let $x, y \in X$ and $\varepsilon = d(x,y)/2$. Then, the open balls are disjoint, i.e. $B_\varepsilon(x) \cap B_\varepsilon(y) = \emptyset$ and $x \in B_\varepsilon(x)$ and $y \in B_\varepsilon(y)$. ■

Since a metric space is Hausdorff, every sequence has at most one limit and every subsequence is convergent to the same limit. This has an impact on applications. Solutions to many practical problems can be expressed as iterated function systems in some metric space. These properties ensure that if such systems are convergent, they are convergent to a unique solution. In practice, however, an additional property of completeness, Def. 2.35, must be required, which takes care that the limit exists in the domain of interest.

Def. 2.34 (Convergence) Let (X,d) be a metric space. Then the sequence x_n converges to $x \in X$, $\lim_{n \to \infty} x_n = x$, if $\lim_{n \to \infty} d(x_n, x) = 0$ or, equivalently, iff the open ball $B_\varepsilon(x) := \{y \in X : d(x,y) < \varepsilon\}$ contains a tail of x_n.

Def. 2.35 (Cauchy sequence, completeness) Let (X, d) be a metric space.

1. A sequence $x_n \in X$ is *Cauchy* iff $\lim_{n,m\to\infty} d(x_n, x_m) = 0$.
2. A space (X, d) is *complete* if every Cauchy sequence converges in X.

Theorem 2.36 (On metric spaces)

1. In a metric space every convergent sequence is Cauchy, but not conversely; see Example 2.37.
2. In a metric space, the distance d is continuous, i.e. the convergence of any two sequences $x_n, y_n \in X$ to x and y, respectively, implies that $\lim_{n\to\infty} d(x_n, y_n) = d(x, y)$.
3. A closed subset of a complete metric space is complete under the induced metric.

Example 2.37 Examples of complete and non-complete spaces:

1. $((0, 1], d_2)$ is not complete. The sequence $x_n = \frac{1}{n} \to 0$ is Cauchy, but not convergent in $(0, 1]$.
2. $(\mathbb{R}^m, d_2)$ is complete.
3. Let Ω be a closed and bounded set in $\mathbb{R}^m$ and $\mathcal{C}(\Omega)$ be a set of continuous functions on Ω. $(\mathcal{C}(\Omega), d_\infty)$ is complete.
4. Let Ω be a closed and bounded set in $\mathbb{R}^m$. $(\mathcal{C}(\Omega), d_p)$ for $1 \leq p < \infty$ is not complete, since some of the Cauchy sequences converge to discontinuous functions [234]. If $\mathcal{M}(\Omega)$ is a set of classes of functions measurable in the Lebesgue sense, then $(\mathcal{M}(\Omega), d_p)$ is complete.
5. Since every metric space is a subset of a complete space [278, 363], it can be completed by adding the limits of all convergent sequences. E.g. $((0, 1], d_2)$ can be completed to $([0, 1], d_2)$.

Def. 2.38 (Generalized metric spaces) Let X be a set and $\rho : X \times X \to \mathbb{R}^0_+$ be a dissimilarity function. If the requirements of Def. 2.30 hold, then ρ is a distance function. If these requirements are weakened, spaces with less constraints[10] are considered; see also Fig. 2.2:

1. *hollow space* - a space (X, d) obeying the reflexivity condition.
2. *premetric space* - a hollow space (X, ρ) obeying the symmetry constraint.
3. *quasimetric space* - a premetric space (X, ρ) obeying the definiteness constraint.
4. *semimetric space* - a premetric space (X, ρ) satisfying the triangle inequality.
5. A hollow space (X, ρ) satisfying the triangle inequality [35].

Example 2.39 Examples of generalized metric spaces:

1. Let $X = \{\mathcal{N}(\mu, \sigma)\}$ be a space of one-dimensional normal distributions. The Mahalanobis distance between them, $d_M(\mathcal{N}(\mu_1, \sigma_1), \mathcal{N}(\mu_2, \sigma_2)) = \frac{|\mu_1 - \mu_2|}{(\sigma_1^2 + \sigma_2^2)^{1/2}}$ is premetric; see Fig. 2.5.
2. Let $X = \mathbb{R}^m$ and k, $1 \geq k \geq m$ is a fixed integer. Then, the distance d_{k-rank} measuring the absolute difference along the k-th dimension, $d_{k-rank}(\mathbf{x}, \mathbf{y}) = |x_k - y_k|$ is semimetric.
3. Let $(\Omega, \mathcal{A}, \mu)$ be a measurable space, i.e. Ω is a set, $\mathcal{A}$ is a σ-algebra of subsets on Ω and μ is a measure. Then $d_\mu(A, B) = \mu(A \triangle B)$ is semimetric [419], where $A \triangle B := (A \cup B) \setminus (A \cap B)$.
4. Let X be a set of closed subsets of $\mathbb{R}^m$. Similarly, as above, the m-dimensional volume symmetric difference $d_{vol}(A, B) = \text{vol}(A \triangle B)$ is semimetric. The definiteness condition is not fulfilled, since $d_{vol}(A, B) = 0$ for finite collections of points A and B. In the pattern recognition area, this dissimilarity can be computed between two matched shapes as the area of non-overlapping parts; see also Fig. 1.2(b).
5. Let (X, ρ) be a semimetric space. Let the equivalence relation $\backsim$ be defined as $x \backsim y$ iff $\rho(x, y) = 0$. If $X^\backsim$ is the set of equivalence classes $[x]$ in X under this relation, then $\rho_\backsim$ defined on $X^\backsim$ such that $\rho_\backsim([x], [y]) = \rho(x, y)$ is a metric on $X^\backsim$ [419].
6. The space $(\mathbb{R}^m, d_p)$, where $d_p(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^m |x_i - y_i|^p)^{1/p}$ and $p \in (0, 1)$ is quasimetric.

[10] Terminology is not unified, it varies between authors and contexts.

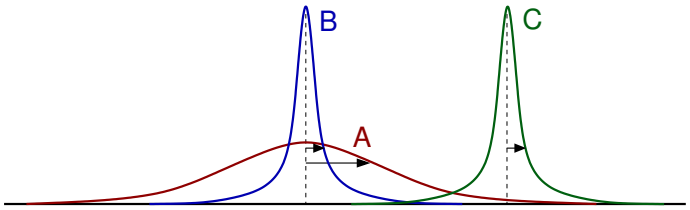


Fig. 2.5: Mahalanobis distance between one-dimensional normal distributions is premetric. The reflexivity and symmetry conditions are satisfied, but the definiteness and triangle inequality are not. Although A and B are different, $d(A,B)=0$. Let $\sigma := \sigma_B = \sigma_C$ and $|\mu_B - \mu_C| = a$. Then $d_M(B,C) = a/(\sqrt{2}\,\sigma)$ and $d_M(C,A) = a/(\sigma^2+\sigma_A^2)^{1/2}$. Since $\sigma_A > \sigma$, $d_M(C,A)+d_M(A,B) < d_M(C,B)$.

> **Proof.** To prove that the triangle inequality does not hold, let $m=2$ and $A=[0,1]^T$, $B=[0,0]^T$ and $C=[1,0]^T$. Then, $d_p(A,B)=d_p(B,C)=1$ and $d_p(C,A)=2^{1/p}$. Finally, $d_p(A,B)+d_p(B,C)=2<2^{1/p}=d_p(C,A)$, since $p<1$. Hence, the triangle inequality is violated. ■

In generalized metric spaces, the definitions of convergence and of a Cauchy sequence are adopted from the metric case, Def. 2.34 and Def. 2.35.

Def. 2.40 (Convergence) Let (X,ρ) be a quasimetric space. An element $x\in X$ is called a limit of an infinite sequence x_n, $\lim_{n\to\infty} x_n = x$, if $\lim_{n\to\infty}\rho(x_n,x)=0$.

Def. 2.41 (Continuity of a dissimilarity) Let (X,ρ) be a generalized metric space. The dissimilarity $\rho\colon X\times X\to\mathbb{R}^0_+$ is continuous at x and y, if for any two sequences $x_n, y_n\in X$, $\lim_{n\to\infty}x_n = x$ and $\lim_{n\to\infty}y_n = y$ implies that $\lim_{n\to\infty}\rho(x_n,y_n)=\rho(x,y)$. Moreover, ρ is continuous in X if it is continuous for each pair from X.

Note that all 'nice' properties of a dissimilarity measure, such as continuity, convergence of a sequence to one limit, Cauchy convergent sequences can be considered for the metric only. A generalized metric space may not fulfill these conditions.

Example 2.42 Let (X,ρ) be a quasimetric space.

1. X is not necessarily a Hausdorff space. A sequence may have more than one limit.
   Proof. Consider a quasimetric space $([0,1],\rho)$ such that $\rho(x,y)=|y-x|$ if $x,y\in[0,1)$, $\rho(x,1):=\rho(x,0)$ if $x\in(0,1)$, $\rho(1,0)=\rho(0,1)=1$ and $\rho(1,1)=0$. Then, the sequence $x_n=\frac{1}{n}$ converges to both 0 and 1, since both $\rho(\frac{1}{n},0)$ and $\rho(\frac{1}{n},1)$ have the limit zero if $n\to\infty$. ■
2. The dissimilarity ρ is not necessarily continuous.
   Proof. Consider a quasimetric space $([0,1],\rho)$ such that $\rho(x,y)=2$ if $x,y\in\{0,1\}$ and $x\neq y$, and $\rho(x,y)=|x-y|$, otherwise. Then, ρ is discontinuous for the pair $(0,1)$, since for $x_n=\frac{1}{n}$ and $y_n=1-\frac{1}{n}$, we have $\lim_{n\to\infty}x_n=0$ and $\lim_{n\to\infty}y_n=1$, but $\lim_{n\to\infty}\rho(x_n,y_n)=1$, while $\rho(x,y)=2$. ■
3. An infinite sequence of elements from X might be convergent without being Cauchy.
   Proof. Consider a space $([0,1],\rho)$, such that $\rho(x,y)=1$ if $x=\frac{1}{n}$, $y=\frac{1}{m}$ and $n\neq m$, and $\rho(x,y)=|x-y|$, otherwise. Then, for $x_n=\frac{1}{n}$, $\lim n\to\infty x_n=0$. So, x_n is convergent, but not Cauchy, since $\lim_{n\to\infty}\rho(\frac{1}{n},\frac{1}{m})=1$. ■

Theorem 2.43 If (X,ρ) is a quasimetric space with a continuous dissimilarity function ρ, then for all $x\in X$ and all $\varepsilon>0$, $B_\varepsilon(x):=\{y\in X\colon \rho(x,y)<\varepsilon\}$ is an open set [363].

Proof. We will use Corollary 2.17 stating that a closed set contains all its limit elements and conversely. To prove that $B_\varepsilon(x)$ is open, we will show that the complementary set $Y:=X\backslash B_\varepsilon(x)=\{y\in X\colon\rho(x,y)\geq\varepsilon\}$ is closed. Let z be a limit element of Y, which means that there exist elements $z_n\in Y$ such that z_n converges to z. From continuity of ρ, $\rho(z_n,z)\to 0$. Since $z_n\in Y$, then for any $x\in X$, one has $\rho(x,z_n)\geq\varepsilon$. From continuity of ρ, it follows that $\rho(x,z)=\lim_{n\to\infty}\rho(x,z_n)\geq\varepsilon$. This proves that $z\in Y$. Consequently, Y is

a closed set, as it contains its limit elements. Hence, $B_\varepsilon(x)$ is open. ■

Theorem⋆ 2.44 (Generalized metric spaces with ball neighboorhoods are pretopological)

(1) A hollow space is pretopological.
(2) A premetric space is pretopological.
(3) A quasimetric space (X, ρ) with a continuous dissimilarity ρ is topological.
(4) A semimetric space is topological.
(5) A hollow space (X, ρ) satisfying the triangle inequality is topological.

Proof. To show that hollow, premetric and quasimetric spaces are pretopological, one needs to prove that the ε-balls define a neighborhood basis. Such a proof directly follows the proof given in the metric case by Theorem 2.33. A continuous dissimilarity measure in a quasimetric space assures that the ε-balls are open sets by Theorem 2.43, hence the axioms of the topological space are fulfilled. The proof that a semimetric space is topological follows the same reasoning as in the metric case; see the proof of Theorem 2.33. The proof that a hollow space satisfying the triangle inequality is topological is given in [35]. ■

Since generalized metric spaces are pretopological, the continuous functions between such spaces can be defined adequately; see Def. 2.19 and also Corollary 2.20. Making use of neighborhood balls, we have:

Def. 2.45 (Continuity of a function) Let (X, d) and (Y, ρ) be generalized metric spaces. A function $f : X \to Y$ is continuous at $x \in X$ if $\forall_{\varepsilon>0}\, \exists_{\delta>0}\, y \in B_\delta(x) \Rightarrow f(y) \in B_\varepsilon(f(x))$, where the neighborhood balls are defined as $B_\delta(x) = \{z : d(x,z) < \delta\}$ and $B_\varepsilon(f(x)) = \{f(z) : \rho(f(z), f(x)) < \varepsilon\}$, respectively. In case of the metrics, the ε- and δ-balls are open sets. The function f is continuous if it is continuous at every $x \in X$.

Corollary 2.46 (On continuous functions) Let (X, d) and (Y, ρ) be metric spaces (or generalized metric spaces with continuous dissimilarity measures). The following assertions are equivalent:

1. f is continuous at x.
2. For every neighborhood M of $f(x) \in Y$, $f^{-1}(M)$ is a neighborhood of $x \in X$.
3. If $\lim_{n\to\infty} x_n = x$, then $\lim_{n\to\infty} f(x_n) = f(x)$.

Corollary 2.47 (Continuity of a composed mapping) Let (X, d_X), (Y, d_Y) and (Z, d_Z) be generalized metric spaces with continuous dissimilarity measures and let $f : X \to Y$, $g : Y \to Z$ and $h : X \to Z$ be mappings. The composed mapping of f and g is denoted by $h = g \circ f$ such that $h(x) = g(f(x))$. If f and g are continuous, then h is continuous as well.

Sketch of proof. The proof follows directly from considering the equivalence between the continuity and the convergence of a sequence based on Corollary 2.46.

Direct product spaces can be used for a construction of a new space by combining two (or more) spaces. In the context of generalized metric spaces, if the measures describe the same set of objects, a new dissimilarity measure can be created, as a result (e.g. by their summation).

Def. 2.48 (Product space) Let (X, d_X) and (Y, d_Y) be generalized metric spaces. Then, a product generalized metric space $X \times Y$ with a dissimilarity d can be defined as $(X \times Y, d_X \bullet d_Y)$, where $\bullet$ is the sum or max operator. This means that $(d_X \bullet d_Y)((x_1, y_1), (x_2, y_2)) = d_X(x_1, x_2) + d_Y(y_1, y_2)$ or $(d_X \bullet d_Y)((x_1, y_1), (x_2, y_2)) = \max\{d_X(x_1, x_2), d_Y(y_1, y_2)\}$ for $x_1, x_2 \in X$ and $y_1, y_2 \in Y$.

The extension of neighborhoods, convergence and continuity to the product space is straightforward. For instance, U is a neighborhood of the pair (x, y) if there exist a neighborhood N of $x \in X$ and a neighborhood M of $y \in Y$ such that $N \times M \subseteq U$. Also, the convergence of a sequence $(x_n, y_n) \in X \times Y$ is equivalent to the convergence of sequences $x_n \in X$ and $y_n \in Y$.

2.3 Normed and inner product spaces

Metric spaces are already richer in structure than topological spaces, still more structure can be introduced; see Fig. 2.1 and Fig. 2.2. Normed and inner product spaces are special cases of metric spaces, where metric is defined either by a norm or an inner product. The algebraic and geometric structures of such spaces are richer than those of metric spaces only. Inner product spaces are important, since there exists a well-developed mathematical theory which places the pattern description and learning in their context. Most of theory presented here can be found in [112, 224, 327].

Def. 2.49 (Normed space) Let X be a vector space. A norm on X is a function $||\cdot||: X \to \mathbb{R}^0_+$ satisfying for all $x, y \in X$ and all $\alpha \in \mathbb{C}$ the following conditions:

(1) Nonnegative definiteness: $||x|| \geq 0$.
(2) Non-degeneration $||x|| = 0$ iff x is a zero vector.
(3) Homogeneity: $||\alpha x|| = |\alpha|\,||x||$.
(4) Triangle inequality: $||x + y|| \leq ||x|| + ||y||$.

A vector space with a norm, $(X, ||\cdot||)$, is called a *normed* space. If only the axioms (1), (3) and (4) are satisfied, then $||\cdot||$ becomes *seminorm* and $(X, ||\cdot||)$ a *seminormed* space.

Example 2.50 Examples of seminormed spaces:

1. $(\mathcal{F}([-1,1], ||\cdot||)$ with $||f|| := |f(0)|$ is a seminormed space.
2. $(\mathbb{R}^m, ||\cdot||_p)$, with $p \geq 1$, where $||\mathbf{x}||_p = (\sum_{i=1}^m |x_i|^p)^{1/p}$ is a normed space.
3. $(\mathbb{R}^m, ||\cdot||_\infty)$, where $||\mathbf{x}||_\infty = \max_{i=1,\ldots,m} |x_i|$, is a normed space.
4. Let $\mathcal{C}(\Omega)$ be a set of continuous functions on a closed and bounded set $\Omega \subset \mathbb{R}^m$. $(\mathcal{C}(\Omega), ||\cdot||_p)$, where $||f||_p = (\int_a^b |f(x)|^p\, dx)^{1/p}$ and $p \geq 1$, is a normed space.

Lemma 2.51 (On seminormed spaces)

1. The (semi)norm is a continuous function, i.e. if $\lim_{n\to\infty} x_n = x$, then $\lim_{n\to\infty} ||x_n|| = ||x||$.
2. Every (semi)normed space is a (semi)metric space with the distance $d(x, y) = ||x - y||$.
3. A (semi)normed space is a topological vector space, where $d(x, y) = ||x - y||$ defines open ball neighborhoods.
4. Not every metric space is a normed space.
   Sketch of proof. Let $X = \mathbb{R}$ and $d(x, y) = \mathcal{I}(x \neq y)$. Suppose that $d(x, y) = ||x - y||$ is true. Then for all $\alpha \in \mathbb{R}$ and $z \in \mathbb{R}$, $||\alpha z|| = |\alpha|||z||$ should hold. Let $z := x - y$, then $||z|| = 1$. Consider $\alpha = 2$. Then, we have $2 = |\alpha|||z|| = ||\alpha z|| = 1$, hence a contradiction. Consequently, there is no norm that generates this metric. ■

Def. 2.52 (Banach space) A normed space for which the associated metric induced by the norm is complete (i.e. every Cauchy sequence converges there) is called a *Banach* space.

Example 2.53 Examples of Banach spaces:

1. $(\mathbb{R}^m, ||\cdot||_2)$ is a Banach space.
2. Let l_p^∞, $p \geq 1$, be a vector space of real sequences $x = (x_1, x_2, \ldots)$ such that $\sum_{i=1}^\infty |x_i|^p < \infty$ with the norm given by $||x||_p = (\sum_{i=1}^\infty |x_i|^p)^{1/p}$. This norm induces the Minkowski metric d_p. Consequently, l_p^∞ and $l_p^m := (\mathbb{R}^m, d_p)$ are Banach spaces.
3. Let l_∞^∞ be a vector space, where each element is a sequence $x = (x_1, x_2, \ldots)$ with norm given by $||x||_\infty = \sup_i |x_i|$. This norm induces the metric d_∞. Therefore, l_∞^∞ is a Banach space. Consequently, the space $l_\infty^m := (\mathbb{R}^m, d_\infty)$ is Banach, as well.

Def. 2.54 (Inner product space) Let X be a vector space. An *inner product* $\langle\cdot,\cdot\rangle$ is a bilinear mapping $X \times X \to \mathbb{C}$ such that for all $x, y, z \in X$ and $\alpha, \beta \in \mathbb{C}$, one has:

(1) Nonnegative definiteness: $\langle x, x\rangle \geq 0$.

(2) Non-degeneration: $\langle x, x\rangle = 0$ iff x is a zero vector.

(3) Symmetry: $\langle x, y\rangle = \langle y, x\rangle^{\dagger}$, where $\dagger$ stands for a complex conjugate.

(4) Linearity in X and over $\mathbb{C}$: $\langle \alpha\, x + \beta y, z\rangle = \alpha\, \langle x, z\rangle + \beta\, \langle y, z\rangle$.

Lemma 2.55 (On inner products)

1. The inner product in an inner product space is a continuous function.
2. Every inner product space is a normed space with the norm defined as $||x|| = \langle x, x\rangle^{1/2}$.
3. The *parallelogram law*: $||x+y||^2 + ||x-y||^2 = 2\,||x||^2 + 2\,||y||^2$ holds for the norm $||x|| = \langle x, x\rangle^{1/2}$.
4. *Polarization identity.* The real inner product $\langle \cdot, \cdot \rangle$ can be determined from the corresponding norm as: $\langle x, y\rangle = \frac{1}{2}(||x + y||^2 - ||x||^2 - ||y||^2)$. The complex inner product can be determined as $\langle x, y\rangle = \frac{1}{4}(||x + y||^2 + ||x - y||^2 - i||x + iy||^2 - i||x - iy||^2)$, where $i^2 = -1$.
5. In a complex inner product space $\langle z, \alpha\, x + \beta y\rangle = \alpha^{\dagger}\, \langle x, z\rangle + \beta^{\dagger}\, \langle y, z\rangle$.

Theorem 2.56 (Cauchy-Bunyakovski-Schwarz inequality) Let X be an inner product space. For all $x, y \in X$, one has $|\, \langle x, y\rangle\, | \leq \langle x, x\rangle^{1/2}\, \langle y, y\rangle^{1/2}$ and the equality holds iff $y = \alpha x$ for some $\alpha \in \mathbb{C}$.

Def. 2.57 (Hilbert, pre-Hilbert space) An inner product space for which the induced norm gives a complete metric space is a *Hilbert* space. A non-complete inner product space is a *pre-Hilbert* space.

Example 2.58

1. $(\mathbb{R}^m, \langle \cdot, \cdot \rangle)$ with $\langle x, y\rangle = \sum_{i=1}^{m} x_i y_i$ is a Hilbert space.
2. l_2^{∞} is a Hilbert space with an inner product defined by $\langle x, y\rangle = \sum_{i=1}^{\infty}\, x_i\, y_i$. The metric becomes then $d(x, y) = ||x - y|| = (\sum_{i=1}^{\infty} (x_i - y_i)^2)^{1/2}$.
3. The space $L_2^{\mathcal{M}}$ defined on a set $\mathcal{M}(\Omega)$ of Lebesgue measurable classes of functions with $\langle f, g\rangle = (\int_a^b\, f(x)\, g(x)\, \mu(dx))^{1/2}$ is a Hilbert space. Note that $L_2^{\mathcal{C}}$, defined on a set of continuous functions, since not complete, is only a pre-Hilbert space.
4. The space l_p^{∞} (and l_p^m) with $p \neq 2$ is not an inner product space, hence not a Hilbert space.

   Sketch of proof. The proof is based on the contradiction of the parallelogram law for $x = (1, 1, 0, 0, \ldots)$ and $y = (-1, 1, 0, 0, \ldots)$.
5. The space $L_{\infty}^{\mathcal{C}}$ on $\Omega := [a, b]$ with the norm $||f||_{\infty} = \max_{x \in [a,b]} |f(x)|$ is not an inner product space, hence not a Hilbert space.

   Sketch of proof. The proof is based on the contradiction of the parallelogram law for the functions $f(x) = a$ and $g(x) = x - a$ defined on $[a, b]$. ■

Def. 2.59 (Orthogonality, orthogonal complement)

1. Let X be an inner product space. Vectors x and y are *orthogonal* in X, $x \perp y$, if $\langle x, y\rangle = 0$. Hence, a zero vector is orthogonal to every vector. A subspace V of X is orthogonal if all pairs of vectors of V are orthogonal.
2. Let $\mathcal{X}$ be a closed subspace of a Hilbert space $\mathcal{H}$. The closed subspace $\mathcal{X}^{\perp} := \{y \in \mathcal{H} : \forall_{x \in \mathcal{X}}\, \langle y, x\rangle = 0\}$ with the property that $\mathcal{X} \cap \mathcal{X}^{\perp} = \{0\}$ is the *orthogonal complement* of $\mathcal{X}$.

Def. 2.60 (Orthonormal basis) Let $\mathcal{H}$ be a Hilbert space. The set $\{e_i\}$ of elements in $\mathcal{H}$ is a *basis* if every $x \in \mathcal{H}$ can be uniquely written as $x = \sum_{i=1}^{\infty} \alpha_i e_i \Leftrightarrow x = \lim_{N \to \infty} \sum_{i=1}^{N} \alpha_i e_i$ for some $\alpha_i \in \mathbb{C}$ and $\sum_{i=1}^{\infty} |\alpha_i|^2$. If additionally, $\langle e_i, e_j\rangle = \delta_{ij}$, where δ_{ij} is the Kronecker delta, $\delta_{ij} = \mathcal{I}(i = j)$, then the basis is *orthonormal*.

Theorem 2.61 (Orthogonal expansions) Let $\{e_i\}_{i=1}^{\infty}$ be an orthonormal basis in a Hilbert space $\mathcal{H}$. Then for all $x, y \in \mathcal{H}$, we have:

1. *Bessel inequality:* $\sum_{i=1}^{\infty} |\langle x, e_i\rangle| \leq ||x||^2$. It is also valid for a pre-Hilbert space.
2. *Parseval formula:* $||x||^2 = \sum_{i=1}^{\infty} |\langle x, e_i\rangle|$.
3. $\langle x, y\rangle = \sum_{i=1}^{\infty} \langle x, e_i\rangle \langle e_i, y\rangle$.

Theorem 2.62 (Projection theorem) Let $\mathcal{V}$ be a closed subspace of $\mathcal{H}$. Then, for every $x \in \mathcal{H}$, there exist unique $x_v \in \mathcal{V}$ and $x_\perp \in \mathcal{V}^\perp$ such that $x = x_v + x_\perp$. Define $x_v = P\,x$, where P is the orthogonal projection of x onto $\mathcal{V}$. P has the following properties:

1. $P^2 = P$ (idempotent).
2. $\langle Px, y\rangle = \langle x, Py\rangle$ (self-adjoint).
3. $\langle Px, (I-P)\,x\rangle = 0$.
4. $x = Px + (I-P)x$ and $P \perp (I-P)$.

Only the first two conditions are required for P to be a projection.

2.3.1 Reproducing kernel Hilbert spaces

Reproducing kernels are used in a variety of applications like function estimation, function approximation or model building. They uniquely define so-called reproducing kernel Hilbert spaces (RKHS), which are spaces of bounded linear functionals[11]. Reproducing kernels are used in statistical learning theory [403] for the construction of support vector machines; see also chapter 4. Here, we will provide some basic definitions and facts. More details can be found e.g. in [22, 112, 337–339, 412].

Def. 2.63 (Positive definite function or kernel) [22, 412] Let X be a set. A Hermitian function $K: X \times X \to \mathbb{C}$ is positive definite (pd) iff for all $n \in \mathbb{N}$, $\{x_1, \ldots, x_n\} \subseteq X$ and $\{c_1, \ldots, c_n\} \subseteq \mathbb{C}$, one has $\sum_{i,j=1}^{n} c_i c_j^\dagger K(x_i, x_j) > 0$, where $\dagger$ stands for a complex conjugate. Such a Hermitian function is called a *kernel*[12]. Additionally, K is conditionally positive definite (cpd) iff the above condition is satisfied only for $\{c_1, \ldots, c_n\}$ such that $\sum_{j=1}^{n} c_j = 0$. Depending on the sign of $\sum_{i,j=1}^{n} c_i c_j^\dagger K(x_i, x_j)$, also (conditionally) negative, nonnegative and nonpositive functions can be defined.

If X is an n-element finite set, such as $X := R = \{p_1, p_2, \ldots, p_n\}$, then K is pd iff the $n \times n$ matrix $K(R, R)$ is pd. Moreover, if K is pd, then $K(p_i, p_i) \geq 0$ for all $p_i \in R$.

Theorem 2.64 (Riesz representation theorem) For every continuous linear functional f on a Hilbert space $\mathcal{H}$, there is a unique $u \in \mathcal{H}$ such that $f(x) = \langle x, u\rangle$ for all $x \in \mathcal{H}$ [80].

Def. 2.65 (Reproducing kernel Hilbert space) Let X be a set and $\mathbb{C}^X$ denotes a space of functions $f: X \to \mathbb{C}$. Let $\mathcal{H}_K \subset \mathbb{C}^X$ be a Hilbert space of bounded (hence continuous) linear functionals. A bilinear function $K: X \times X \to \mathbb{C}$ is a *reproducing kernel* for $\mathcal{H}_K$ if

1. $K(x, \cdot) \in \mathcal{H}_K$ for all $x \in X$ and
2. $K(x, \cdot)$ is the representer of evaluation at x in $\mathcal{H}_K$, that is $f(x) = \langle f, K(x, \cdot)\rangle_{\mathcal{H}_K}$ for all $f \in \mathcal{H}_K$ and all (fixed) $x \in X$.

$\mathcal{H}_K$ equipped with K is called the reproducing kernel Hilbert space (RKHS).

The reproducing kernel map is realized by $\psi: x \to K(x, \cdot)$, so $\psi(y) = K(x, y)$. Since $K(y, \cdot)$ is the representer of evaluation at y, then $\psi(y) = \langle \psi, K(y, \cdot)\rangle_{\mathcal{H}_K} = \langle K(x, \cdot), K(y, \cdot)\rangle_{\mathcal{H}_K}$ As a result, one gets $K(x, y) = \langle K(x, \cdot), K(y, \cdot)\rangle_{\mathcal{H}_K}$. This means that a pd kernel K can be seen as a Gram operator in

[11] Consider a linear functional $f: X \to \mathbb{C}$ on a linear normed space X. It is bounded if there exists $M \in \mathbb{R}_+$ such that $|f(x)| \leq M\,||x||_X$ for all $x \in X$. It is known that a linear functional is bounded iff it is continuous.

[12] Kernel K originates from the study of integral operators, where $(L_K f)(x) = \int_X K(x, y) f(y) dy$. Then, K is called the kernel of the operator L_K.

$\mathcal{H}_K$, i.e. there exists a function ψ in a Hilbert space $\mathcal{H}_K$ such that the evaluation of the kernel at x and y is equivalent to taking the inner product between $\psi(x)$ and $\psi(y)$.

If X is a set of a finite cardinality, say n, then the functions are evaluated only at a finite number of points. Consequently, the RKHS becomes an n-dimensional space, where the linear functions become n-dimensional vectors. As a result, the reproducing kernel K simplifies to an $n \times n$ Hermitian (or symmetric) pd matrix.

Theorem 2.66 (Mercer theorem) Let $\mathcal{H}_K$ be a Hilbert space of functions $f : X \to \mathbb{C}$ and let $K : X \times X \to \mathbb{C}$ be a pd kernel[13]. If $\langle K(x,\cdot), K(x,\cdot)\rangle_{\mathcal{H}_K} \leq \infty$, then K can be expanded by a countable sequence of orthonormal eigenfunctions ψ_i and real positive eigenvalues λ_i such that the bilinear series $K(x,y) = \sum_{i=1}^{\infty} \lambda_i \psi_i(x) \psi_i(y)^\dagger$ converges uniformly and absolutely[14].

The theorem above means that the eigenfunctions and eigenvalues are found as a solution to the eigenequation $\langle K(x,\cdot), \psi_i(\cdot)\rangle_{\mathcal{H}_K} = \lambda_i \psi_i(x)$ or, in the integral form, $\int_X K(x,y)\psi_i(y)dy = \lambda_i \psi_i(x)$, if K corresponds to an inner product defined by the integral. In practice this requires that X is a compact subset of $\mathbb{R}^m$ or an index set. As the eigenfunctions $\{\psi_i\}_{i=1}^{\infty}$ are linearly independent functions (an orthonormal basis of $\mathcal{H}_K$), then any function f in the space $\mathcal{H}_K$ can be written as $f(x) = \sum_{i=1}^{\infty} a_i \psi_i(x)$. The inner product between f and g in the Hilbert space $\mathcal{H}_K$ is defined as $\langle f(x), g(x)\rangle_{\mathcal{H}_K} = \sum_{i=1}^{\infty} \frac{a_i\, b_i^\dagger}{\lambda_i}$, where $g(x) = \sum_{i=1}^{\infty} b_i \psi_i(x)$. Indeed, such a space of functions with the kernel K is a RKHS, since $\langle f, K(x,\cdot)\rangle_{\mathcal{H}_K} = \langle f(y), K(x,y)\rangle_{\mathcal{H}_K} = \langle f(y), K(y,x)^\dagger\rangle_{\mathcal{H}_K} = \sum_{i=1}^{\infty} \frac{a_i\, ((\lambda_i \psi_i(x))^\dagger)^\dagger}{\lambda_i} = \sum_{i=1}^{\infty} a_i psi_i(x) = f(x)$, because K is Hermitian, i.e. $K(x,y) = K(y,x)^\dagger$. Note that $||f||^2_{\mathcal{H}_K} = \langle f(x), f(x)\rangle^2_{\mathcal{H}_K} = \sum_{i=1}^{\infty} \frac{|a_i|^2}{\lambda_i}$ and $||K||^2_{\mathcal{H}_K} = \langle K(x,\cdot), K(x,\cdot)\rangle^2_{\mathcal{H}_K} = \sum_{i=1}^{\infty} \lambda_i$.

There is an equivalence between choosing a specific $\mathcal{H}_K$, reproducing kernel K and defining the set of λ_i and ψ_i.

Theorem 2.67 (Moore-Aronszajn) [412] For every pd kernel K on $X \times X$ (X is a compact set), there exists a unique RKHS $\mathcal{H}_K$ over X for which K is the reproducing kernel and vice versa.

2.4 Indefinite inner product spaces

Indefinite inner product is a generalization of a (positive definite) inner product $\langle\cdot,\cdot\rangle$, Def. 2.54, by requiring that only the symmetry and linearity conditions hold. The facts presented here are based on books of Alpay et al. [3], Bognár [34], Greub [177] and Iohvidov [204] and the articles [67, 92, 151, 152, 320]. Proofs and propositions are ours.

Def. 2.68 (Indefinite inner product space) Let $\mathcal{V}$ be a vector space. An *indefinite inner product* $\langle\cdot,\cdot\rangle_{\mathcal{V}}$ is a mapping $\mathcal{V} \times \mathcal{V} \to \mathbb{C}$ such that for all $x, y, z \in \mathcal{V}$ and $\alpha, \beta \in \mathbb{C}$, one has:

(1) Symmetry: $\langle x, y\rangle_{\mathcal{V}} = \langle y, x\rangle^\dagger_{\mathcal{V}}$, where $\dagger$ stands for a complex conjugate.

(2) Linearity in X and over $\mathbb{C}$: $\langle \alpha\, x + \beta y, z\rangle_{\mathcal{V}} = \alpha\, \langle x, z\rangle_{\mathcal{V}} + \beta\, \langle y, z\rangle_{\mathcal{V}}$,

Since $\langle x, x\rangle_{\mathcal{V}}$ can have any sign, there is a distinction among positive, negative and neutral vectors and the corresponding subspaces. For the material presented below, $\mathcal{V}$ is assumed to be an indefinite inner product space. We will write $\langle\cdot,\cdot\rangle$ only if the traditional inner product, Def. 2.54, is meant, otherwise, we will write $\langle\cdot,\cdot\rangle_{\mathcal{V}}$ to refer to a vector space $\mathcal{V}$.

[13] In the integral form the positive-definiteness means that $\langle Kf, f\rangle_{\mathcal{H}_K} = \int_{X\times X} K(x,y) f(x) f(y)^\dagger dxdy \geq 0$.

[14] Let $\{u_n\}$ be a set of functions $X \to \mathbb{C}$. A series $\sum_{i=1}^{\infty} u_n(x)$, converges uniformly to $u(x)$ iff for every $\varepsilon > 0$, there exists a natural number N, such that for all $x \in X$ and all $n \geq N$, $|u_n(x) - u(x)| < \varepsilon$. For a fixed x, a series $\sum_i u_i(x)$ converges absolutely if the series $\sum_i |u_i(x)|$ converges.

Def. 2.69 (Positive, negative and neutral vectors) A vector $x \in \mathcal{V}$ is *positive* if $\langle x, x\rangle_{\mathcal{V}} > 0$, *negative* if $\langle x, x\rangle_{\mathcal{V}} < 0$ or *neutral* if $\langle x, x\rangle_{\mathcal{V}} = 0$. A subspace $\mathcal{X} \subset \mathcal{V}$ is called positive, negative or neutral if all its elements are so, respectively. Every indefinite inner product space contains at least one non-zero neutral vector [34].

Def. 2.70 (Orthogonality, isotropic subspace, degenerate subspace)

1. Vectors $x, y \in \mathcal{V}$ are *orthogonal* if $\langle x, y\rangle_{\mathcal{V}} = 0$.
2. A vector $v \in \mathcal{V}$ is *isotropic* if it is a non-zero vector orthogonal to every vector in $\mathcal{V}$.
3. Let $\mathcal{X} \subseteq \mathcal{V}$. Then, $\mathcal{X}^{\perp} = \{y \in \mathcal{V} : \forall_{x\in\mathcal{X}}\ \langle y, x\rangle_{\mathcal{V}} = 0\}$ is an *orthogonal complement* of $\mathcal{X}$.
4. Let *zero* be the zero vector. Let $\mathcal{X} \subseteq \mathcal{V}$. The isotropic subspace $\mathcal{X}_0$ of $\mathcal{X}$ consists of isotropic vectors, i.e. $\mathcal{X}_0 = \mathcal{X} \cap \mathcal{X}^{\perp}$. If $\mathcal{X}_0 \neq \theta$, then $\mathcal{X}$ is *degenerate* and $\langle \cdot, \cdot\rangle_{\mathcal{V}}$ is degenerate on $\mathcal{X}$. The entire space $\mathcal{V}$ is degenerate if $\mathcal{V}^{\perp} \neq \theta$.

Example 2.71 Inner product spaces:

1. Let $\mathcal{V}$ be a vector space of pairs of real numbers. Let $\langle x, y\rangle_{\mathcal{V}} = x_1\, y_1 - x_2 y_2$ for $x = (x_1, x_2)$ and $y = (y_1, y_2)$. Then $(\mathcal{V}, \langle \cdot, \cdot\rangle_{\mathcal{V}})$ is indefinite. Note also that if $\mathcal{X} = \{(x_1, x_2) :\ x_1 + x_2 = 0\}$, then $\mathcal{X}^{\perp} = \mathcal{X}$. Hence $\mathcal{X}$ is a degenerate subspace of $\mathcal{V}$.
2. Let $\mathcal{V}$ be a vector space of number sequences $(v_1, v_2, \ldots)$ satisfying $\sum_{i=1}^{\infty} |\varepsilon_i|\, |v_i|^2 < \infty$. Then, $\langle x, y\rangle_{\mathcal{V}} = \sum_{i=1}^{\infty} \varepsilon_i\, x_i\, y_i^{\dagger}$ defines an inner product. Depending on the signs of ε_i, $\langle x, y\rangle_{\mathcal{V}}$ may be positive, negative or indefinite. If ε_i are of different signs, then $\langle x, y\rangle_{\mathcal{V}}$ is indefinite. Moreover, if there exists at least one zero ε_j, then $\langle x, y\rangle_{\mathcal{V}}$ is degenerate.
3. Let $L([a, b])$ be a vector space of real valued functions that are measurable and square-summable with respect to some function μ. Then $(f, g) = \int_a^b f(x)\, g(x)\, d\mu(x)$ defines an inner product which might be indefinite or definite, depending on the function μ; see also [185].

Def. 2.72 (Fundamental decomposition) Let $(\mathcal{V}, \langle \cdot, \cdot\rangle_{\mathcal{V}})$ be an indefinite product space. If $\mathcal{V}$ is represented as a direct orthogonal decomposition[15] $\mathcal{V} = \mathcal{V}_+ \oplus \mathcal{V}_- \oplus \mathcal{V}_0$, such that $\mathcal{V}_+$, $\mathcal{V}_-$ and $\mathcal{V}_0$ are positive, negative and neutral subspaces, respectively, then such a decomposition is called a *fundamental decomposition* and $\mathcal{V}$ is decomposable.

Not every space $\mathcal{V}$ admits a fundamental decomposition [34], yet, every finite-dimensional inner product space does. Spaces which yield a fundamental decomposition are called Kreĭn spaces and are of our interest. Pseudo-Euclidean spaces are the most simple examples of these. See also Fig. 2.2(b).

Def. 2.73 (Pseudo-Euclidean space and its orthonormal basis) A pseudo-Euclidean space $\mathcal{E} := \mathbb{R}^{(p,q)}$ is a real linear vector space equipped with a non-degenerate, indefinite inner product $\langle \cdot, \cdot\rangle_{\mathcal{E}}$ [177]. $\mathcal{E}$ admits a direct orthogonal decomposition $\mathcal{E} = \mathcal{E}_+ \oplus \mathcal{E}_-$, where $\mathcal{E}_+ = \mathbb{R}^p$ and $\mathcal{E}_- = \mathbb{R}^q$ and the inner product is positive definite on $\mathcal{E}_+$ and negative definite on $\mathcal{E}_-$. The space $\mathcal{E}$ is, therefore, characterized by the *signature* (p, q) [151]. An *orthonormal* basis $\{\mathbf{e}_1, \ldots, \mathbf{e}_{p+q}\}$ in $\mathcal{E}$ is given such that $\langle \mathbf{e}_i, \mathbf{e}_i\rangle = 1$ for $i = 1, \ldots, p$ and $\langle \mathbf{e}_i, \mathbf{e}_i\rangle = -1$ for $i = p+1, \ldots, p+q$ and $\langle \mathbf{e}_i, \mathbf{e}_j\rangle = 0$ for $i \neq j$.

By making use of the standard inner product $\langle \cdot, \cdot\rangle$ in a Euclidean space, the inner product between two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^{(p,q)}$ can be expressed as:

$$\langle \mathbf{x}, \mathbf{y}\rangle_{\mathcal{E}} = \sum_{i=1}^{p} x_i y_i - \sum_{i=p+1}^{p+q} x_i y_i = \mathbf{x}^T \mathcal{J}_{pq} \mathbf{y} = \langle \mathcal{J}_{pq}\, \mathbf{x}, \mathbf{y}\rangle, \quad \mathcal{J}_{pq} = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix}, \tag{2.1}$$

[15] A direct sum $Z = X \oplus Y$ means that every $z \in Z$ can be uniquely decomposed into $x \in X$ and $Y \in Y$ such that $z = x + y$. Here, a direct orthogonal decomposition $\mathcal{V} = \mathcal{V}_+ \oplus \mathcal{V}_- \oplus \mathcal{V}_0$ means that $\mathcal{V}_- = \mathcal{V}_+^{\perp}$ and $\mathcal{V}_0 = (\mathcal{V}_+ \cap \mathcal{V}_+^{\perp})^{\perp}$, i.e. $\mathcal{V}_0 = \mathcal{V}_+ \cap \mathcal{V}_+^{\perp}$ consists of neutral vectors perpendicular to all other vectors in $\mathcal{V}$.

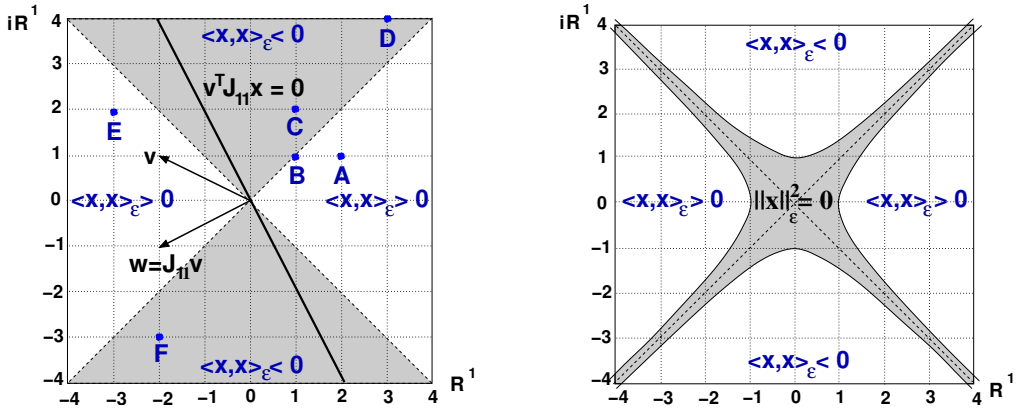


Fig. 2.6: Left: a pseudo-Euclidean space $\mathcal{E} = \mathbb{R}^{(1,1)} := \mathbb{R}^1 \times i\mathbb{R}^1$ with $d^2(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})^T \mathcal{J}_{11}(\mathbf{x}-\mathbf{y})$. Orthogonal vectors are mirrored versus the lines $x_2 = x_1$ or $x_2 = -x_1$, for instance $\langle OA, OC\rangle_\mathcal{E} = 0$. Vector $\mathbf{v}$ defines the plane $0 = \langle \mathbf{v}, \mathbf{x}\rangle_\mathcal{E} = \mathbf{v}^T \mathcal{J}_{11}\mathbf{x}$. Note that the vector $\mathbf{w} = \mathcal{J}_{11}\mathbf{v}$, a 'flipped' version of $\mathbf{v}$, describes the plane as if in a Euclidean space $\mathbb{R}^2$. Therefore, in any pseudo-Euclidean space, the inner product can be interpreted as a Euclidean operation, where one vector is 'flipped' by $\mathcal{J}_{pq}$. The square distances can have any sign, e.g. $d^2(A,C) = 0$, $d^2(A,B) = 1$, $d^2(B,C) = -1$, $d^2(D,A) = -8$, $d^2(F,E) = -24$ and $d^2(E,D) = 32$. Right: A pseudo-sphere $||\mathbf{x}||^2_\mathcal{E} = x_1^2 - x_2^2 = 0$. From the Euclidean point of view, this is an open set between two conjugated hyperbolas. Consequently, the rotation of a point is carried out along them; see also [152].

where $I_{p\times p}$ and $I_{q\times q}$ are the identity matrices. If $\mathbf{x}_+$ and $\mathbf{x}_-$ stand for the orthogonal projections of $\mathbf{x}$ onto $\mathbb{R}^p$ and $\mathbb{R}^q$, respectively, then $\langle \mathbf{x},\mathbf{y}\rangle_\mathcal{E} = \langle \mathbf{x}_+, \mathbf{y}_+\rangle - \langle \mathbf{x}_-, \mathbf{y}_-\rangle$. The pseudo-norm of a non-zero vector $\mathbf{x}$ becomes then $||\mathbf{x}||^2_\mathcal{E} = \langle \mathbf{x},\mathbf{x}\rangle_\mathcal{E} = \mathbf{x}^T \mathcal{J}_{pq}\,\mathbf{x}$, which can be positive, negative or zero. Making use of the inner product, the squared distance can be expressed analogous to the Euclidean case as

$$d^2(\mathbf{x},\mathbf{y}) = ||\mathbf{x}-\mathbf{y}||^2_\mathcal{E} = \langle \mathbf{x}-\mathbf{y},\, \mathbf{x}-\mathbf{y}\rangle_\mathcal{E} = (\mathbf{x}-\mathbf{y})^T\, \mathcal{J}_{pq}\, (\mathbf{x}-\mathbf{y}) \tag{2.2}$$

which can be positive, negative or zero. Therefore, the distance d is either real or in the form of $i\sqrt{|d|}$, where $i^2 = 1$. Note that the distance between distinct $\mathbf{x}$ and $\mathbf{y}$ may also be zero.

Alternatively, a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$ can be represented as a Cartesian product $\mathbb{R}^p \times i\,\mathbb{R}^q$. It is, thereby, a $(p+q)$-dimensional real subspace of the $(p+q)$-dimensional complex space $\mathbb{C}^{p+q}$, obtained by taking the real parts of the first p coordinates and imaginary parts of the remaining q coordinates. This justifies formulas (2.1) and (2.2) and allows one to express the square distance as $d^2(\mathbf{x},\mathbf{y}) = d^2_{\mathbb{R}^p}(\mathbf{x},\mathbf{y}) - d^2_{\mathbb{R}^q}(\mathbf{x},\mathbf{y})$, where the distances on the right side are square Euclidean. A Euclidean space is a special case of the pseudo-Euclidean space, i.e. $\mathbb{R}^p = \mathbb{R}^{(p,0)}$.

The notions of symmetric and orthogonal matrices should be now properly redefined.

Def. 2.74 (Symmetric, orthogonal matrices) Let A be an $n\times n$ matrix in $\mathbb{R}^{(p,q)}$, $n = p+q$.

1. A is *symmetric* or *self-adjoint* if $\mathcal{J}_{pq}\, A^T A = A$.
2. A is *orthogonal* if $(\mathcal{J}_{pq} A^T)\mathcal{J}_{pq}\, A = I$.

The matrix $\mathcal{J}_{pq}$ plays a key role in the definitions above. In general, a symmetric or orthogonal matrix in a pseudo-Euclidean space is *not* symmetric or orthogonal in the Euclidean sense. If, however, $\mathbb{R}^{(p,q)}$ coincides with a Euclidean space, i.e. $q = 0$, then the above definitions simplify to the traditional ones, since $\mathcal{J}_{pq}$ becomes the identity operator I. For instance, by straightforward operations one can check that the matrix $\left[\begin{smallmatrix} 1 & -2 \\ 2 & -1 \end{smallmatrix}\right]$ is symmetric in $\mathbb{R}^{(1,1)}$ with $\mathcal{J}_{pq} = \left[\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right]$, and that $\frac{1}{\sqrt{3}}\left[\begin{smallmatrix} 2 & -1 \\ 1 & -2 \end{smallmatrix}\right]$ is orthogonal in $\mathbb{R}^{(1,1)}$. If we denote $A^* := \mathcal{J}_{pq}\, A^T\, \mathcal{J}_{pq}$, then the conditions above can be reformulated as $A^* = A$ for a symmetric A and as $A^* A = I$ for an orthogonal A. This formulation already suggests that A^* may play a special role. Adjoined operators are discussed below.

Note also that the symmetry and orthogonality of A, Def. 2.74, can be equivalently formulated by treating $\mathbb{R}^{(p,q)}$ as $\mathbb{R}^p \times i\,\mathbb{R}^q$. Let $A = [A_p \;\; A_q]$, where the matrices A_p and A_q of the sizes $n \times p$ and $n \times q$ correspond to the spaces $\mathbb{R}^p$ and $\mathbb{R}^q$, respectively. A is symmetric if $[A_p \;\; iA_q]^\dagger = [A_p \;\; iA_q]$ and A is orthogonal if $[A_p \;\; iA_q]^\dagger [A_p \;\; iA_q] = I$, where $i^2 = 1$.

A further extension of a pseudo-Euclidean space leads to a Kreĭn space, which is a generalization of a Hilbert space as a pseudo-Euclidean space is a generalization of a Euclidean space.

Def. 2.75 (Kreĭn and Pontryagin spaces) A *Kreĭn space* is a linear space $\mathcal{K}$ over $\mathbb{C}$ satisfying:

1. There exists a bilinear form $\langle\cdot,\cdot\rangle_\mathcal{K}$ on $\mathcal{K}$ such that for any $x, y, z \in \mathcal{K}$ and any $\alpha, \beta \in \mathbb{C}$ the following conditions are fulfilled:
   a. Symmetry: $\langle x, y\rangle_\mathcal{K} = \langle y, x\rangle_\mathcal{K}^\dagger$,
   b. Linearity: $\langle \alpha\, x + \beta\, y, z\rangle_\mathcal{K} = \alpha\, \langle x, z\rangle_\mathcal{K} + \beta\, \langle y, z\rangle_\mathcal{K}$.
2. $\mathcal{K}$ admits a direct orthogonal decomposition $\mathcal{K} = \mathcal{K}_+ \oplus \mathcal{K}_-$ such that $(\mathcal{K}_+, \langle\cdot,\cdot\rangle)$ and $(\mathcal{K}_-, -\langle\cdot,\cdot\rangle)$ are Hilbert spaces[16] and $\langle x_+, x_-\rangle_\mathcal{K} = 0$ for any $x_+ \in \mathcal{K}_+$ and $x_- \in \mathcal{K}_-$. $\mathcal{K}_-$ is called also an *antispace* with respect to $\langle\cdot,\cdot\rangle$.

In other words, $\mathcal{K}$ admits a fundamental decomposition with a positive subspace $\mathcal{K}_+$ and a negative subspace $\mathcal{K}_-$. Therefore, $\mathcal{K}_+ = (\mathcal{K}_-)^\perp$. Let $\dim \mathcal{K}_+ = \kappa_+$ and $\dim \mathcal{K}_- = \kappa_-$ be the ranks of positivity and negativity, respectively. Kreĭn spaces with a finite rank of negativity are called *Pontryagin spaces* (in other sources, e.g. [34], the rank of positivity is assumed to be finite). A Pontryagin space with a finite κ_- is denoted by Π_κ. Note that if $\langle\cdot,\cdot\rangle_\mathcal{K}$ is positive definite or zero for zero vectors only, then $\mathcal{K}$ is a Hilbert space.

Example 2.76 (Pseudo-Euclidean, Kreĭn and Pontryagin spaces) Let $\mathcal{V}$ be a vector space of real sequences $(v_1, v_2, \ldots)$ satisfying $\sum_{i=1}^\infty |\varepsilon_i|\,|v_i|^2 < \infty$. Then, $\langle x, y\rangle_\mathcal{V} := \sum_{i=1}^\infty \varepsilon_i\, x_i\, y_i$ defines an inner product. If $\varepsilon_1 = 1$ and $\varepsilon_j = -1$ for all $j > 1$, then the inner product is given as $\langle x, y\rangle_\mathcal{V} = x_1 y_1 - \sum_{i=2}^\infty x_i\, y_i$ and $\mathcal{V}$ becomes a Pontryagin space. If $\varepsilon_{2j} > 0$ and $\varepsilon_{2j-1} < 0$, then $\mathcal{V}$ equipped with $\langle x, y\rangle_\mathcal{V}$ defines a Kreĭn space. If $\mathcal{V}$ is a vector space of finite sequences $(v_1, v_2, \ldots, v_m)$ and all $\varepsilon_j \neq 0$, then $\mathcal{V}$ with $\langle x, y\rangle_\mathcal{V} = \sum_{i=1}^m \varepsilon_i\, x_i\, y_i$ is a pseudo-Euclidean space.

Corollary 2.77 (Indefinite inner product expressed by the traditional one) $\langle x, y\rangle_\mathcal{K} = \langle \mathcal{J}\, x, y\rangle$.

Def. 2.78 (Fundamental symmetry and fundamental projections) Let $\mathcal{K} = \mathcal{K}_+ \oplus \mathcal{K}_-$. The orthogonal projections P_+ and P_- onto $\mathcal{K}_+$ and $\mathcal{K}_-$, respectively, are called *fundamental projections*. Therefore, any $x \in \mathcal{K}$ can be represented as $x = P_+\, x + P_-\, x$ where $I_\mathcal{K} = P_+ + P_-$ is the identity operator in $\mathcal{K}$. The linear operator $\mathcal{J} = P_+ - P_-$ is called a *fundamental symmetry*.

In Hilbert spaces, the classes of symmetric, self-adjoint, isometric and unitary operators are well known [112, 327]. Linear operators, carrying the same names can also be defined in Kreĭn spaces. The definitions are analogous and many results from Hilbert spaces can be generalized to Kreĭn spaces. However, due to indefiniteness of the inner product, the classes of some special properties with respect to the inner product are larger. We will only present the most important (for us) results; see [34, 151, 152, 204, 312] for details.

Def. 2.79 (H-scalar product, H-norm) Let $x, y \in \mathcal{K}$. Then, the *H-scalar product* is defined as $[x, y] = \langle \mathcal{J} x, y\rangle_\mathcal{K}$ and the *H-scalar norm* is $||x||_h = [x, x]^{\frac{1}{2}}$.

Let $x \in \mathcal{K}$ be represented as $x = x_+ + x_-$, where $x_+ \in \mathcal{K}_+$ and $x_- \in \mathcal{K}_-$. Since $[x, y] = \langle \mathcal{J} x, y\rangle_\mathcal{K}$, based on this, we can write $[x, y] = \langle x_+, y_+\rangle_\mathcal{K} - \langle x_-, y_-\rangle_\mathcal{K} = \langle x_+, y_+\rangle - (-\langle x_-, y_-\rangle) = \langle x, y\rangle$. This means that $[x, y]$ is equivalent to the traditional (Hilbertian) inner product and $\mathcal{K}_+$ and $\mathcal{K}_-$ are orthogonal

[16] All Hilbert spaces discussed here are assumed to be separable, i.e. they admit countable bases.

with respect to $[x,y]$. Moreover, the *associated Hilbert space* $\mathcal{H}$ is then such $\mathcal{H}=|\mathcal{K}|:=\mathcal{K}_+\oplus|\mathcal{K}_-|$, where $|\mathcal{K}_-|$ stands for $(\mathcal{K}_-,\langle\cdot,\cdot\rangle)$. Formally, there is a close 'bound' between a Kreĭn space and the associated Hilbert space:

Lemma 2.80 A decomposable, non-degenerate inner product space $\mathcal{K}$ is a Kreĭn space iff for every fundamental symmetry $\mathcal{J}$, the H-scalar product turns it into a Hilbert space [34].

H-scalar product is a Hilbert inner product, therefore $\mathcal{K}$ can be regarded as a complete Hilbert space (Banach space) with the H-scalar product (H-norm). As a result, the (strong) topology of $\mathcal{K}$ is the norm topology of the associated Banach space, i.e. the H-norm topology. This topology does not depend on the choice of fundamental symmetry [34][17]. Therefore, continuity, convergence etc. can be defined for $\mathcal{K}$ with respect to the H-norm.

Def. 2.81 (Convergence, Cauchy sequence)

1. The sequence x_n in $\mathcal{K}$ converges to $x\in\mathcal{K}$ with respect to the H-norm iff $\lim_{n\to\infty}\langle x_n,y\rangle_{\mathcal{K}}=\langle x,y\rangle_{\mathcal{K}}$ for all $y\in\mathcal{K}$ and $\lim_{n\to\infty}\langle x_n,x_n\rangle_{\mathcal{K}}=\langle x,x\rangle_{\mathcal{K}}$.
2. The sequence x_n in $\mathcal{K}$ is Cauchy with respect to the H-norm iff $\langle x_n-x_m,x_n-x_m\rangle_{\mathcal{K}}\to 0$ and $\langle x_n,y\rangle_{\mathcal{K}}$ form a Cauchy sequence for $y\in\mathcal{K}$.

Corollary 2.82 Since $\langle x,y\rangle_{\mathcal{K}}=[x_+,y_+]-[x_-,y_-]$, then $\langle x,y\rangle_{\mathcal{K}}$ is continuous with respect to the H-norm in both x and y.

Theorem 2.83 (Schwarz inequality) For all $x,y\in\mathcal{K}$, we have: $|\langle x,y\rangle_{\mathcal{K}}|\le||x||_h\,||y||_h$.

Proof. $|\langle x,y\rangle_{\mathcal{K}}|^2\le|[x_+,y_+]-[x_-,y_-]|^2\le(||x_+||\,||y_+||+||x_-||\,||y_-||)^2\le(||x_+||^2+||x_-||^2)(||y_+||^2+||y_-||^2)=||x||_h^2||y||_h^2$. ∎

Theorem 2.84 (Orthogonal expansions) If $\mathcal{K}_+$ and $\mathcal{K}_-$ are separable Hilbert spaces, then there exists a countable orthonormal basis $\{e_i\}_{i=1}^{\infty}$ in $\mathcal{K}$ whose span contains any element x of $\mathcal{K}$. This means that $\langle e_i,e_i\rangle_{\mathcal{K}}=1$ if P_+e_i is an orthonormal vector in $\mathcal{K}_+$, $\langle e_i,e_i\rangle_{\mathcal{K}}=-1$ if P_-e_i is an orthonormal vector in $\mathcal{K}_-$, and $\langle e_i,e_j\rangle_{\mathcal{K}}=0$, otherwise. For $x,y\in\mathcal{K}$, we also have [34]:

(1) $\sum_{i=1}^{\infty}|\langle x,e_i\rangle_{\mathcal{K}}|^2<\infty$.
(2) $-\sum_{\langle e_i,e_i\rangle_{\mathcal{K}}=-1}|\langle x,e_i\rangle_{\mathcal{K}}|^2\le\langle x,x\rangle_{\mathcal{K}}\le\sum_{\langle e_i,e_i\rangle_{\mathcal{K}}=1}|\langle x,e_i\rangle_{\mathcal{K}}|^2$.
(3) $\langle x,y\rangle_{\mathcal{K}}\le\sum_{i=1}^{\infty}\langle e_i,e_i\rangle_{\mathcal{K}}\langle x,e_i\rangle_{\mathcal{K}}\langle e_i,y\rangle_{\mathcal{K}}$.

Def. 2.85 (Adjoint operator) Let $\mathcal{L}_c(\mathcal{K},\mathcal{G})$ be a space of continuous linear operators on the Kreĭn space $\mathcal{K}$ onto the Kreĭn space $\mathcal{G}$. If $\mathcal{G}$ is $\mathcal{K}$, then $\mathcal{L}_c(\mathcal{K})$ will be used.

1. $A^*\in\mathcal{L}_c(\mathcal{G},\mathcal{K})$ is a unique *adjoint* of $A\in\mathcal{L}_c(\mathcal{K},\mathcal{G})$ if $\langle Ax,y\rangle_{\mathcal{G}}=\langle x,A^*y\rangle_{\mathcal{K}}$ for $x\in\mathcal{K}$ and $y\in\mathcal{G}$.
2. $A\in\mathcal{L}_c(\mathcal{K})$ is *self-adjoint* (*symmetric*) if $A^*=A$, i.e. $\langle Ax,y\rangle_{\mathcal{K}}=\langle x,Ay\rangle_{\mathcal{K}}$ for all $x,y\in\mathcal{K}$.

Observation 2.86 [34, 204] The fundamental symmetry $\mathcal{J}$ fulfills $\mathcal{J}=\mathcal{J}^*=\mathcal{J}^{-1}$.

Theorem 2.87 (Factorization) [34] Every self-adjoint operator $A\in\mathcal{L}_c(\mathcal{K})$ can be expressed as $A=TT^*$, where $T\in\mathcal{L}_c(\mathcal{V},\mathcal{K})$ for some Kreĭn space $\mathcal{V}$ and $\ker(T)=0$, where $\ker(T):=\{x\in\mathcal{V}\colon T(x)=0\}$.

Def. 2.88 (Isometric and unitary operators) [3, 34] Let $A\in\mathcal{L}_c(\mathcal{K},\mathcal{G})$, then A is *isometric* if $A^*A=I_{\mathcal{G}}$ and *coisometric* if $AA^*=I_{\mathcal{K}}$. $A\in\mathcal{L}_c(\mathcal{K})$ is *unitary* if $\langle Ax,Ay\rangle_{\mathcal{K}}=\langle x,y\rangle_{\mathcal{K}}$ for all $x,y\in\mathcal{K}$, or in other words, if both isometric and coisometric.

[17] In a Kreĭn space, there are infinitely many fundamental decompositions, hence fundamental symmetries and, consequently, infinitely many associated Hilbert spaces. However, the decompositions yield the same ranks of positivity and negativity, the same H-norm topologies; simply, they are isomorphic.

Since a Kreĭn space is inherently connected to the associated Hilbert space, both the adjoint and unitary operators can be expressed through operators in this Hilbert space. Hence, the assertion $\langle Ax, y\rangle_{\mathcal{G}} = \langle x, A^*y\rangle_{\mathcal{K}}$ is equivalent to $\langle \mathcal{J}Ax, g\rangle = \langle \mathcal{J}x, A^*y\rangle$, which is equivalent to $\langle \mathcal{J}Ax, g\rangle = \langle x, \mathcal{J}A^*\rangle$, since $\mathcal{J}$ is self-adjoint also in the associated Hilbert space. This means that in a Hilbert space, the adjoint of $(\mathcal{J}A)$ is $(\mathcal{J}A^*)$. Let $A^\dagger$ be a Hilbert adjoint of A, then $(\mathcal{J}A)^\dagger = A^\dagger\mathcal{J} = \mathcal{J}A^*$ and finally $A^* = \mathcal{J}A^\dagger\mathcal{J}$.

For a unitary operator in a Kreĭn space, we have $\langle Ax, Ay\rangle_{\mathcal{K}} = \langle x, y\rangle_{\mathcal{K}}$, which is equivalent to stating that $\langle \mathcal{J}Ax, Ay\rangle = \langle \mathcal{J}x, y\rangle$. Since $\mathcal{J}$ is self-adjoint, then $\langle (\mathcal{J}A)x, (\mathcal{J}A)y\rangle = \langle x, y\rangle$. So, $(\mathcal{J}A)$ is a unitary operator in a Hilbert space, which means that $(\mathcal{J}A)^\dagger = (\mathcal{J}A)^{-1}$. Then, $A^{-1} = \mathcal{J}A^\dagger\mathcal{J}$, which is equivalent to $A^{-1} = A^*$. Formally, we have:

Corollary 2.89 Let $A \in \mathcal{L}_c(\mathcal{K}, \mathcal{G})$, then $A \in \mathcal{L}_c(|\mathcal{K}|, |\mathcal{G}|)$ for the associated Hilbert spaces $|\mathcal{K}|$ and $|\mathcal{G}|$. If $A^\dagger$ is a Hilbert adjoint of A, then we have $A^* = J_{\mathcal{K}}\, A^\dagger J_{\mathcal{G}}$, where $\mathcal{J}_{\mathcal{K}}$ and $\mathcal{J}_{\mathcal{G}}$ are the fundamental symmetries. Moreover, $||A^*||_h = ||A^\dagger||_h = ||A||_h$.

Def. 2.90 (A Kreĭn subspace) A Kreĭn (regular) subspace of a Kreĭn space $\mathcal{K}$ is a subspace $\mathcal{X}$ which is a Kreĭn space in the inner product of $\mathcal{K}$, i.e.: $\langle x, y\rangle_{\mathcal{X}} = \langle x, y\rangle_{\mathcal{K}}$ for $x, y \in \mathcal{X}$.

Def. 2.91 (Positive, uniformly positive subspaces) A closed or non-closed subspace $\mathcal{V} \in \mathcal{K}$ is *positive*, if $\langle x, x\rangle_{\mathcal{K}} > 0$ for all $x \in \mathcal{V}$ and $\mathcal{V}$ is *uniformly positive* if it is positive and $\langle x, x\rangle_{\mathcal{K}} > \alpha||x||_h^2$ for some positive α depending on $\mathcal{X}$ and the associated H-norm. Similar definitions can be made for *negative*, *uniformly negative*, *nonnegative* etc. subspaces. The term *maximal*, if added, stands for a subspace which is not properly contained in another subspace with the same property.

Every maximal positive (negative) subspace of a Kreĭn space is closed. If $\mathcal{K} = \mathcal{K}_+ \oplus \mathcal{K}_-$ is the fundamental decomposition, then the subspaces $\mathcal{K}_+$ and $\mathcal{K}_-$ are maximal uniformly positive or negative, respectively. Any maximal uniformly positive or negative subspace arises in this way [34].

Def. 2.92 (Positive definite operator) A self-adjoint operator $A \in \mathcal{L}_c(\mathcal{K})$ is *positive definite* (k-pd) in a Kreĭn space if $\langle x, Ax\rangle_{\mathcal{K}} > 0$ for all $x \in \mathcal{K}$. The negative definiteness (k-nd) or semi-definiteness can be defined accordingly.

The above condition is equivalent to stating that $0\langle x, Ax\rangle_{\mathcal{K}} < \langle \mathcal{J}x, Ax\rangle = \langle x, \mathcal{J}Ax\rangle$. This means that A is k-pd if $\mathcal{J}A$ is pd in the associated Hilbert space $|\mathcal{K}|$. For instance, the fundamental symmetry $\mathcal{J}$ is k-pd, since it is self-adjoint and $\mathcal{J}\mathcal{J} = I$.

Theorem 2.93 (Projection theorem) [34, 204] Let $\mathcal{V}$ be a closed, non-degenerate subspace of $\mathcal{K}$. Then, for every $x \in \mathcal{K}$, there exist unique $x_v \in \mathcal{V}$ and $x_\perp \in \mathcal{V}^\perp$ such that $x = x_v + x_\perp$, where $x_v = P\,x$ and P is the orthogonal projection of x onto $\mathcal{V}$. P has the following properties:

1. $P^2 = P$ (idempotent).
2. $\langle Px, y\rangle_{\mathcal{K}} = \langle x, Py\rangle_{\mathcal{K}}$ (self-adjoint).
3. $\langle Px, (I_{\mathcal{K}} - P)\,x\rangle_{\mathcal{K}} = 0$.
4. $x = Px + (I_{\mathcal{K}} - P)\,x$ and $P \perp (I_{\mathcal{K}} - P)$.

Only the first two conditions are required for P to be a projection.

Def. 2.94 (Gram and cross-Gram operators) Let $\mathcal{V}$ be a linear subspace of $\mathcal{K}$ spanned by linearly independent vectors $\{v_1, v_2, \ldots, v_n\}$. The *Gram* operator, i.e. the inner product operator, is defined as $G_{vv} = (\langle v_i, v_j\rangle_{\mathcal{K}})_{i,j=1,\ldots,n}$. Assume further that a subspace $\mathcal{U} \subseteq \mathcal{K}$, spanned by $\{u_1, \ldots, u_t\}$, is given. Then, $G_{vu} = (\langle v_i, u_j\rangle_{\mathcal{K}})_{i=1,\ldots,t;\, j=1,\ldots,n}$ is the *cross-Gram* operator.

Theorem⋆ 2.95 (Projection onto a subspace) Let $\mathcal{V} = \text{span}\{v_1, v_2, \ldots, v_n\}$ be a linear subspace of a Kreĭn space $\mathcal{K}$. Let $V := [v_1, \ldots, v_n]^\dagger$ be the adjoint (conjugate transpose operator) of the corresponding operator in the associated Hilbert space $|\mathcal{V}|$. If the Gram operator $G_{vv} := (\langle v_i, v_j \rangle_\mathcal{K})_{i,j=1,\ldots,n}$ is nonsingular, then the orthogonal projection of $x \in \mathcal{K}$ onto $\mathcal{V}$ is unique and it is given by[18]

$$x_v = V^\dagger G_{vv}^{-1}\, \mathbf{g}_x, \tag{2.3}$$

where $\mathbf{g}_x = [\langle x, v_1 \rangle_\mathcal{K}, \ldots, \langle x, v_n \rangle_\mathcal{K}]^\dagger$. If the Gram operator G_{vv} is singular, then either the projection does not exist or $x_v = V^\dagger \mathbf{z}$, where $\mathbf{z}$ is a solution to the linear system $G_{vv}\, \mathbf{z} = \mathbf{g}_x$.

Proof. Let x_v be the projection of x onto $\mathcal{V}$. Based on Theorem 2.93, $x = x_v + x_\perp$ and $\langle x_\perp, v_i \rangle_\mathcal{K} = 0$. The latter formula allows us to write $\mathbf{g}_x = [\langle x_v, v_1 \rangle_\mathcal{K}, \ldots, \langle x_v, v_n \rangle_\mathcal{K}]^\dagger$. Since $\{v_i\}$ are linearly independent, then there exists $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that $x_v = \sum_{i=1}^n \alpha_i v_i = V^\dagger \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is a column vector. Plugging this into $\mathbf{g}_x$, gives rise to $\mathbf{g}_x = G_{vv}\, \boldsymbol{\alpha}$. If G_{vv} is nonsingular, then $\boldsymbol{\alpha}$ can be determined uniquely as $G_{vv}^{-1}\, \mathbf{g}_x$, hence $x_v = V^\dagger G_{vv}^{-1}\, \mathbf{g}_x$. If G_{vv} is singular then either there is no solution to $\mathbf{g}_x = G_{vv}\, \boldsymbol{\alpha}$ or there are many solutions. ■

In a Hilbert space, the singularity of the Gram operator G_{vv} means that the $\{v_i\}$ are linearly dependent. In case of a Kreĭn space, this means that $\mathcal{V}$ contains an isotropic vector, i.e. there exists a linear combination if $\{v_i\}$ which is orthogonal to every vector in $\mathcal{V}$. In other words, to avoid the singularity of the Gram operator, the subspace $\mathcal{V}$ must be non-degenerate.

Observation⋆ 2.96 Since $\langle x, v_i \rangle_\mathcal{K} = \langle \mathcal{J}x, v_i \rangle = x^\dagger \mathcal{J} v_i$, then by the use of the Hilbert operations only, we can write that $\mathbf{g}_x = V\mathcal{J}x$ and also $G_{vv} = V\mathcal{J}V^\dagger$. As a result, $x_v = V^\dagger (V\mathcal{J}V^\dagger)^{-1} V\mathcal{J}x$ and the projection operator P of x onto $\mathcal{V}$ is expressed as $P = V^\dagger (V\mathcal{J}V^\dagger)^{-1} V\mathcal{J}$.

Corollary⋆ 2.97 Let $\mathcal{V} = \text{span}\{v_1, v_2, \ldots, v_n\}$ and $\mathcal{U} = \text{span}\{u_1, u_2, \ldots, u_t\}$ be linear subspaces of $\mathcal{K}$. Assume the Gram operator G_{vv} and the cross-Gram operator $G_{vu} = (\langle v_i, u_j \rangle_\mathcal{K})_{i=1:t,\, j=1:n}$. If G_{vv} is nonsingular, then by Theorem 2.95 the orthogonal projections of the elements from $\mathcal{K}$ onto $\mathcal{V}$ are given by $Q_\mathcal{V} = G_{vu}\, G_{vv}^{-1}\, V$.

Theorem⋆ 2.98 (Indefinite least-square problem from a Hilbertian perspective)[19]. Let $\mathcal{V}$, spanned by $\{v_1, v_2, \ldots, v_n\}$, be a linear non-degenerate subspace of a Kreĭn space $\mathcal{K}$ and let $V := [v_1, \ldots, v_n]^\dagger$. Then, for $u \in \mathcal{K}$, the function $F(x) := ||u - V^\dagger x||_\mathcal{K}^2$ reaches its minimum iff $G_{vv} := V\mathcal{J}V^\dagger$ is positive definite in a Hilbert sense[20]. Then, the solution is found as $x_s = G_{vv}^{-1} \mathbf{g}_u$, where $\mathbf{g}_u := V\mathcal{J}u$. Otherwise, no solution exists.

Proof. $||u - V^\dagger x||_\mathcal{K}^2 = u^\dagger \mathcal{J} u - 2\, x^\dagger V \mathcal{J} u + x^\dagger V \mathcal{J} V^\dagger x$. From mathematical analysis [28, 125], x_s is a stationary point of $F(x)$ if $\frac{\partial F}{\partial x}\big|_{x=x_s} = 0$. By a straightforward differentiation of F, one gets $2\, V\mathcal{J}V^\dagger x - 2\, V\mathcal{J}u = 0$, hence $V\mathcal{J}V^\dagger x_s = V\mathcal{J}u$. Since V is non-degenerate, then G_{vv}^{-1} exists. Therefore, by Observation 2.96, the solution is then given as $x_s = (V\mathcal{J}V^\dagger)^{-1} V\mathcal{J}u = G_{vv}^{-1} \mathbf{g}_u$. Traditionally, the stationary point x_s is a unique minimum iff the $n \times n$ Hessian H with $H_{ij} (\frac{\partial^2 F}{\partial x_i\, \partial x_j})_{i,j=1}^n |_{x=x_s}$ is positive definite in a Hilbert sense. Indeed,

[18] The same formulation holds for a Hilbert space, provided that the inner product $\langle \cdot, \cdot \rangle$ is used instead of $\langle \cdot, \cdot \rangle_\mathcal{K}$.

[19] For comparison, an equivalent formulation is given for the Hilbert case:
(Least-square problem in a Hilbert space) Let $\mathcal{V} = \text{span}\{v_1, v_2, \ldots, v_n\}$ be a linear subspace of a Hilbert space $\mathcal{H}$ and let $V := [v_1, \ldots, v_n]^\dagger$. Then, for $u \in \mathcal{H}$, the norm $F(x) := ||u - V^\dagger x||^2$ is minimized for x such that $V^\dagger x := u_v$, i.e. the orthogonal projection of u onto $\mathcal{V}$. The unique solution is found as $x_s = G_{vv}^{-1} \mathbf{g}_u$, where G_{vv} if the Gram matrix (in a Hilbert space) and $\mathbf{g}_u = [\langle u, v_1 \rangle, \ldots, \langle u, v_n \rangle]^T$.
Proof. $||u - V^\dagger x||^2 = ||u - u_v + u_v - V^\dagger x||^2 = ||u - u_v||^2 + ||u_v - V^\dagger x||^2$, since $\langle u - u_v, u_v - V^\dagger x \rangle = 0$. From Theorem 2.95, we know that the projection of u onto $\mathcal{V}$ is unique and it is given by $u_v = V^\dagger G_{vv}^{-1}\, \mathbf{g}_u$. $F(x)$ is then minimized for $||u_v - V^\dagger x||^2 = ||V^\dagger G_{vv}^{-1}\, \mathbf{g}_u - V^\dagger x||^2$ being equal to zero, if the sought solution is $x_s = G_{vv}^{-1}\, \mathbf{g}_u$.

[20] From a Hilbertian point of view, the minimum of F cannot be found for an arbitrary indefinite space. Assume, for instance a Kreĭn space $\mathcal{K} := \mathbb{R}^{(1,1)}$ with the pseudo norm $||x||_\mathcal{K}^2 = x_1^2 - x_2^2$. Then, for a particular $x := [1\ x_2]$, the minimum of $||0 - x||_2^\mathcal{K} = 1 - x_2^2$ is reached at $-\infty$.

the Hessian equals $H = 2\,V\mathcal{J}V^\dagger$, so $G_{vv} = V\mathcal{J}V^\dagger$ should be positive definite. If G_{vv} is not positive definite than x_s cannot be considered as the solution to the indefinite least-square problem. ■

Below, we present an interpretation of the indefinite least-square problem, but from the indefinite point of view. The solution does not change, however, the interpretation does:

Theorem⋆ 2.99 (Indefinite least-square problem) Let $\mathcal{V}$, spanned by $\{v_1, v_2, \ldots, v_n\}$, be a linear non-degenerate subspace of a Kreĭn space $\mathcal{K}$ and let $V := [v_1, \ldots, v_n]^\dagger$. Then, for $u \in \mathcal{K}$, the function $F(x) := ||u - V^\dagger x||^2_{\mathcal{K}}$ is minimized in the Kreĭn sense[21] for x such that $V^\dagger x := u_v$, i.e. the orthogonal projection of u onto $\mathcal{V}$. The unique solution is found as $x_s = G_{vv}^{-1}\mathbf{g}_u$.

Proof. Similarly as in the proof above, we have: $||u - V^\dagger x||^2_{\mathcal{K}} = u^\dagger\mathcal{J}u - 2\,x^\dagger V\mathcal{J}u + x^\dagger V\mathcal{J}V^\dagger x$. x_s is a stationary point of $F(x)$ if the $\frac{\partial F}{\partial x}|_{x=x_s} = 0$. This leads to the equation $V\mathcal{J}V^\dagger x_s = V\mathcal{J}u$. By Observation 2.96, the solution is then given as $x_s = G_{vv}^{-1}\mathbf{g}_u$. Traditionally, the stationary point x_s is a unique minimum iff the Hessian is pd in a Hilbert sense. Equivalently, in an indefinite case, we should require that the Hessian, equal to $2\,V\mathcal{J}V^\dagger$, is positive definite in the Kreĭn sense. Indeed, since $\mathcal{J}V\mathcal{J}V^\dagger$ is positive definite in the Hilbert sense[22], then, according to Def. 2.92, $V\mathcal{J}V^\dagger$ is positive definite in a Kreĭn space. Since $\mathcal{J} = P_+ - P_-$, then $VP_+V^\dagger$ is positive definite in a Hilbert space $\mathcal{K}_+$, hence $x_{s,\mathcal{K}_+}$ yields a minimum there and $-VP_-V^\dagger$ is negative definite in a Hilbert space $|\mathcal{K}_-|$, hence $x_{s,\mathcal{K}_-}$ yields a maximum there. ■

Note that the system of linear equations $V\mathcal{J}V^\dagger x = V\mathcal{J}u$ to be solved in an indefinite least-square problem can be expressed as $Q^*Qx = Q^*u$, where $Q = V^\dagger$. This can be seen as a system of normal equations in a Kreĭn space. Consequently, $G_{vv}^{-1}V\mathcal{J}$ can be interpreted as a pseudo-inverse of V.

2.4.1 Reproducing kernel Kreĭn spaces

Reproducing kernel Kreĭn spaces (RKKS) are natural extensions of reproducing kernel Hilbert spaces (RKHS). The basic intuition here relies on the fact that a Kreĭn space is composed as a direct orthogonal sum of two Hilbert spaces, hence the reproducing property of the Hilbert kernels can be extended to a Kreĭn space, basically by constructing two reproducing Hilbert kernels and combining them in a usual way. We will present some facts on reproducing kernel Pontryagin spaces (RKPS), which are Kreĭn spaces with a finite rank of negativity (in other sources, e.g. [34], a rank of positivity is assumed to be finite; then the results have to be converted). Here, we will only present the most important issues, for details and proofs, see the book of Alpay et al. [3] and also the articles [67, 92, 320]. All Hilbert spaces associated to Kreĭn spaces are considered to be separable.

Def. 2.100 (Hermitian kernel) Let X be a closed and bounded set. A function K defined on $X \times X \to \mathbb{C}$ of continuous linear operators is called a *Hermitian kernel* if $K(x,y) = K(x,y)^*$ for all $x, y \in X$. $K(x,y)$ has κ negative squares (κ is a nonnegative integer) if every matrix $\{K(x_i,x_j)\}_{i,j=1}^n$ for $n = 1, 2, \ldots$ and $\{x_1, \ldots, x_n\} \in X$ has at most κ negative eigenvalues and at least one such a matrix that has exactly κ negative eigenvalues.

Lemma 2.101 Let Π_κ be a Pontryagin space and let $x_1, x_2, \ldots, x_n \in \Pi_\kappa$. The Gram operator $(\langle x_i, x_j\rangle_{\Pi_\kappa})_{i,j=1}^n$ can have no more than κ negative eigenvalues. Every total set in Π_κ contains a finite subset whose Gram matrix has exactly κ negative eigenvalues [3].

[21] The minimum here is understood as a special saddle point of F in a Hilbert space such that $F|_{\mathcal{K}_+}$ takes minimum at $u_{s,\mathcal{K}_+}$ and $F|_{\mathcal{K}_-}$ takes the maximum at $u_{s,\mathcal{K}_-}$, where $u_{s,\mathcal{K}_+}$ and $u_{s,\mathcal{K}_-}$ are the fundamental projections of u_s onto either $\mathcal{K}_+$ or $\mathcal{K}_-$, respectively.

[22] This follows from $\mathcal{J} = P_+ - P_-$, where P_+ and P_- are fundamental projections. One has $\mathcal{J}V\mathcal{J}V^\dagger = P_+VP_+V^\dagger + P_-VP_-V^\dagger - P_+VP_-V^\dagger - P_-VP_+V^\dagger$. The last two terms become zero, since for any $u \in \mathcal{K}$, we have $u = u_+ + u_-$, where $u_\pm = P_\pm u$ and $I = P_+ + P_-$ is the identity operator and $u_+^\dagger\mathcal{J}u_- = 0$, see also Def. 2.75. Hence, $\mathcal{J}V\mathcal{J}V^\dagger = IVIV^\dagger = VV^\dagger$, which is positive definite in the Hilbert sense.

Lemma 2.102 Let $x_1, x_2, \ldots, x_n$ belong to an inner product space $\mathcal{K}$. Then, the number of negative eigenvalues of the Gram operator $(\langle x_i, x_j\rangle_{\mathcal{K}})_{i,j=1}^n$ coincides with the dimensionality of the maximal negative subspace of span$\{x_1, \ldots, x_n\}$ [3].

Def. 2.103 (Reproducing kernel Kreĭn space) Let X be some measurable set and $\mathbb{C}^X$ denotes a space of functions $f: X \to \mathbb{C}$. Let $\mathcal{K}_K \subset \mathbb{C}^X$ be a Hilbert space of continuous linear functionals on X. A bilinear function $K: X \times X \to \mathbb{C}$ is a *reproducing kernel* $\mathcal{K}_K$ if

1. $K(x,\cdot) \in \mathcal{K}_K$ for all $x \in X$ and
2. $K(x,\cdot)$ is the representer of evaluation at x in $\mathcal{K}_K$: $f(x) = \langle f, K(x,\cdot)\rangle_{\mathcal{K}_K}$ for all $f \in \mathcal{K}_K$ and all (fixed) $x \in X$.

$\mathcal{K}_K$ equipped with K is called a *reproducing kernel Kreĭn space* (RKKS). If $\mathcal{K}_K$ is a Pontryagin space, then the resulting space of functions is called a *reproducing kernel Pontryagin space* (RKPS).

A reproducing kernel exists if all evaluation mappings $E(x): x \to K(x,\cdot)$ are continuous. This means that $E(x) \in \mathcal{L}_c(\mathcal{K}_K, \mathbb{C})$ for every $x \in X$. Hence, the reproducing kernel is unique and can be written as $K(x,y) = E(x)\,E(y)^*$, where $E(x)^* \in \mathcal{L}_c(\mathbb{C}, \mathcal{K}_K)$ is the adjoint of the evaluation mapping $E(x)$ for any fixed $x \in X$. Similarly to the Hilbert case, one has $\langle K(x,\cdot), K(y,\cdot)\rangle_{\mathcal{K}_K} = K(x,y)$. In case of the Pontryagin space, $K(x,y)$ has at most κ negative squares, Def. 2.100, where κ is the rank of negativity.

Theorem 2.104 (On reproducing kernels) [320] Let $K(x,y)$ be a Hermitian kernel $X \times X \to \mathbb{C}$. The following assertions are equivalent:

1. $K(x,y)$ is a reproducing kernel for some Kreĭn space $\mathcal{K}_K$ of functions on X.
2. $K(x,y)$ has a nonnegative majorant[23] $L(x,y)$ on $X \times X$.
3. $K(x,y) = K_+(x,y) - K_-(x,y)$ for some nonnegative definite kernels K_+ and K_- on $X \times X$.

If the above holds, then for a given nonnegative majorant $L(x,y)$ for $K(x,y)$, there exists a Kreĭn space $\mathcal{K}_K$ with a reproducing kernel $K(x,y)$, which is continuously contained in the Hilbert space $\mathcal{H}_L$ with the reproducing kernel $L(x,y)$.

Note that $L(x,y)$ can be chosen as $K_+(x,y) + K_-(x,y)$. Note also that the consequence of this theorem is that the decomposition $K(x,y) = K_+(x,y) - K_-(x,y)$ can be realized such that K_+ is a reproducing (Hilbert) kernel for $(\mathcal{K}_K)_+$ and K_- is a reproducing (Hilbert) kernel for $(\mathcal{K}_K)_-$ in a fundamental decomposition $\mathcal{K}_K = (\mathcal{K}_K)_+ \oplus (\mathcal{K}_K)_-$. Practically, this means that the $K_\pm$ can be chosen as reproducing kernels for the spaces in a fundamental decomposition.

Theorem 2.105 (On reproducing kernels in RKPS) Suppose that $K_1(x,y)$ and $K_2(x,y)$ are reproducing kernels for Pontryagin spaces Π_{κ_1} and Π_{κ_2} of linear functions on X with the ranks of negativity κ_1 and κ_2, respectively. Then, $K(x,y) = K_1(x,y) + K_2(x,y)$ is the reproducing kernel for a Pontryagin space Π_κ with $\kappa \le \kappa_1 + \kappa_2$. Equality holds iff $\mathcal{H} = \Pi_{\kappa_1} \cap \Pi_{\kappa_2}$ is a Hilbert space with the inner product: $\langle x,y\rangle_{\mathcal{H}} = \langle x,y\rangle_{\Pi_{\kappa_1}} + \langle x,y\rangle_{\Pi_{\kappa_2}}$ for $x, y \in \mathcal{H}$.

2.5 Discussion

Some classes of spaces are briefly described. These are pretopological and topological spaces, generalized metric spaces, normed and various inner product spaces. Normed and metric spaces are topological. A norm can also be induced in an inner product space, hence a topology. Therefore, Euclidean, Hilbert and Banach spaces, as the usual examples of inner product and normed spaces, are topological, as well.

[23] A nonnegative majorant L for K is a nonnegative definite kernel L such that $L - K$ and $L + K$ are nonnegative definite kernels in the 'Hilbert' sense, i.e. according to Def. 2.63.

In practical applications more general spaces are also of interest. However, most of the developed learning methodology deals with feature-based representations of objects either in Euclidean or Hilbert spaces. The reason is that in these spaces inner product, norm (defined by the inner product), metric (defined by the norm) and topology (defined by the metric balls) coincide. Since the learning approaches are well developed there, a natural requirement for dissimilarity data seems to be their metric behavior. As a result, in statistical learning many dissimilarity measures are constructed or corrected to obey this condition. Additionally, other l_p metrics are considered in feature vector spaces, such as the city block or max-norm distance. On the other hand, many general dissimilarity measures are derived for object comparisons in the pattern recognition area, as briefly described in chapter 5. Therefore, there is a need for learning paradigms working with general dissimilarity measures.

Only if a proper mathematical foundation is established for metric and non-metric dissimilarities, more general measures may be used and developed further in pattern recognition and machine learning fields. It is the aim of this dissertation to develop such general learning methods and to apply them to a number of problems. They will be constructed in a mathematical framework relying on generalized metric spaces (e.g. pretopological spaces) and Kreĭn spaces, which reduce to pseudo-Euclidean spaces for finite data representations. Since Kreĭn spaces are a natural extensions of Hilbert spaces, Kreĭn spaces accommodate a more general interpretation of dissimilarity data than Hilbert spaces.

Our starting point is a dissimilarity representation, which can be interpreted in a (finite) generalized metric space. From this point of view, all the relations and close bounds between generalized metric spaces and (indefinite) inner product spaces, as well as, generalized metric spaces and generalized topological spaces are important. The most essential properties have been just discussed. How these spaces are used for learning becomes the topic of chapter 4.

3. Characterization of dissimilarities

A rock pile ceases to be a rock pile the moment a single man contemplates it, bearing within him the image of a cathedral.

"FLIGHT TO ARRAS", ANTOINE DE SAINT-EXUPÉRY

Various spaces in the context of generalized metric spaces were introduced in chapter 2. These are pretopological spaces, as well as normed and (indefinite) inner product spaces. This chapter focuses on theoretical aspects of dissimilarities and the relations between generalized metric spaces and inner product spaces. Semimetric and metric transformations, as well as isometric embeddings are described, since they provide a basic framework, where learning algorithms can be created for dissimilarities. A theory is also presented, which deals with transformations preserving metric properties or which allows one to test whether a particular dissimilarity is Euclidean. In brief, this chapter introduces some tools that check or enhance particular properties of dissimilarity representations. It prepares the ground on which data exploration techniques and learning algorithms, discussed in chapters 6 – 10, will rely on.

In practice, we always deal with finite samples, i.e. a finite collection of numerically represented data entities. This finite representation is used to define a space or a more general framework, where learning algorithms can be applied. For a set R of n objects, the representation is given as an $n \times n$ dissimilarity matrix $D(R, R)$. Each entry d_{ij} of D is a dissimilarity value between the i-th and j-th objects. Consequently, the properties of dissimilarity measures and possible spaces where they can be interpreted are mainly discussed in the context of such finite collections.

Metric dissimilarities have advantageous properties, since many methods work in (Euclidean) metric spaces. Section 3.1 briefly introduces basic aspects of city block and Euclidean embeddings. Such isometric mappings find correspondences between an abstract space defined by a (finite) representation of distances and a chosen metric space. Semimetric and metric transformations are also considered. Next, also tree models for the representation of dissimilarity relations are introduced. Section 3.2 presents basic relations and properties of dissimilarity matrices, especially with respect to the metric and Euclidean behavior.

Many traditional learning methods are designed in a Hilbert space or in a Euclidean space equipped with a Euclidean distance. Therefore, given a distance measure, it is important to check whether it has a Euclidean behavior. For the Euclidean distance, every finite representation D can be perfectly embedded in a Euclidean space. This means that a configuration in a Euclidean space can be found such that the original distances are preserved. If the measure is non-Euclidean, then either it is corrected to become Euclidean or it is used directly. Any premetric non-Euclidean measure, i.e. satisfying the definiteness and symmetry constraints, Def. 2.38, can be interpreted as a distance in a pseudo-Euclidean space (Kreĭn space). Section 3.3 explains how both isometric and approximate embeddings in a pseudo-Euclidean space can be achieved. Such mappings are examples of spatial models of the dissimilarity data. Some other projection techniques are presented in section 3.4. Additionally, spherical embeddings are discussed in section 3.3.8.

3.1 Embeddings, tree models and some transformations

Both embeddings and tree models are means to represent generalized metric spaces in spatial organizations. The main purpose of (isometric) embeddings is to determine whether a given space

(X, d) with a dissimilarity measure d is isometrically equivalent to a predefined space possessing some useful properties. Tree models order the dissimilarity information in terms of organizational aspects, hierarchical and nested structures.

3.1.1 Embeddings

Embeddings are a useful tool in practical problems, where finite dissimilarity representations, i.e. finite (generalized) metric spaces (X, d) defined by the corresponding dissimilarity matrix D, are considered. If an equivalence between such spaces and other *known* spaces is established, the latter, if possessing favorable properties, can be used for the setting and construction of learning paradigms.

Many spaces can be considered in this context, but Hilbert and Euclidean spaces are the most extensively investigated. The reason for their applicability is the fact that they are simultaneously inner product, normed and metric spaces, where the inner product is used to define the norm, which further defines the metric. These properties assure that many theoretical models exist, which are used for the solution of pattern recognition problems formulated in such spaces. Studying the questions related to embeddings allows one for a better characterization of commonly used metric spaces, as well as the relations between them.

The primary work in the area of Euclidean (Hilbert) embeddings was done by Blumenthal [33], Cayley [56], Menger [268] and Schoenberg [341, 343, 344]. Since Euclidean embeddings require a thorough treatment, they become the subject of section 3.2 and partly of section 3.3. Here, we will briefly describe some aspects of the l_p-embeddings, with l_1-embeddings, in particular. The l_1-embeddings rely on the additive property of the l_1 metric, which, on the other hand, can be represented by an additive tree; see also Def. 3.12.

Let us first remind basic definitions (see also Example 2.31). Let $\mathcal{M}([a, b])$ be a set of functions classes on $[a, b]$ measurable in the Lebesgue sense. Formally, one has:

- $l_p^m := (\mathbb{R}^m, d_p)$, where $d_p(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^m |x_i - y_i|^p)^{1/p}$ and $p > 0$.
- $l_\infty^m := (\mathbb{R}^m, d_{\max})$, where $d_{\max}(\mathbf{x}, \mathbf{y}) = \max_i |x_i - y_i|$.
- $L_p^{\mathcal{M}} := (\mathcal{M}([a, b]), d_p)$, where $d_p(f, g) = (\int_a^b |f(x) - g(x)|^p \, dx)^{1/p}$ and $p > 0$.
- $L_\infty^{\mathcal{M}} := (\mathcal{M}([a, b]), d_p)$, where $d_\infty(f, g) = \int_a^b \sup_x |f(x) - g(x)| \, dx$.

l_p^m defines an m-dimensional space, while l_p^∞ describes an infinite dimensional space. For simplicity, we will also write l_p instead of l_p^m, when the dimensionality m is fixed. If $p \geq 1$, then l_p and $L_p^{\mathcal{M}}$ are metric spaces, otherwise, they are quasimetric spaces. l_1 stands for the city block metric, while l_2 is the Euclidean metric.

Def. 3.1 (Isometric embedding) Let (X, d_X) and (Y, d_Y) be metric spaces. Then, (X, d_X) is isometrically embeddable into (Y, d_Y), if there exists an isometry $\phi\colon X \to Y$, i.e. a mapping ϕ such that $d_X(x_1, x_2) = d_Y(\phi(x_1), \phi(x_2))$ for all $x_1, x_2 \in X$.

Def. 3.2 (l_p-embeddability) A metric space (X, d) is l_p-embeddable if (X, d) is isometrically embeddable into the space l_p^m for some integer $m \geq 1$.

Isometries are injective. Two spaces are isometrically isomorphic (see footnote 8 on page 22) if there exists a bijective isometry between them. In this case, the two spaces are essentially identical. Every metric space is isometrically isomorphic to a subset of some normed vector space. Every complete metric space is isometrically isomorphic to a closed subset of some Banach space.

Def. 3.3 (Lipschitz mapping and contraction) Let (X, d_X) and (Y, d_Y) be metric spaces. A mapping $\phi : X \to Y$ is *Lipschitz* continuous if there exists a constant κ such that for all $x_1, x_2 \in X$,

$d_Y(\phi(x_1),\phi(x_2)) \le \kappa\, d_X(x_1,x_2)$ holds. If $\kappa<1$, then ϕ is called a *contraction*.

Lemma 3.4 (On Lipschitz mappings)

(1) Every Lipschitz mapping is continuous.
The reverse is not generally true. E.g. let $X=\mathbb{R}$, $d(x,y)=|x-y|$ and $f(x)=x^2$. f is continuous, but not Lipschitz, since no κ exists that $|x^2-y^2| \le \kappa |x-y|$. To show this, consider $y=0$. Then for $|x|\le 1$, $x^2 \le |x|$, but for $|x|>1$, $x^2>x$, hence a contradiction.

(2) Let (X,d) be a metric space. For every $z\in X$, the mapping $x \to d(x,z)$ is Lipschitz with $\kappa=1$.

Lemma 3.5 [33] A Euclidean space $\mathbb{R}^m$ can be embedded in a Hilbert space. Every finite subset of m elements in a Hilbert space can be embedded in a $\mathbb{R}^{m-1}$.

Not every metric space (X,d) can be embedded in a Hilbert space. A counterexample is a metric space of four elements, i.e. $X=\{I,J,K,L\}$ represented by a distance matrix on the right. From the definition of d, there exist two points J and L which should be the middle points between I and K. However, in a Hilbert space, every pair x and y determines a unique middle point $z=\frac{1}{2}(x+y)$ between them such that $d(x,z)=d(z,y)=\frac{1}{2}d(x,y)$ [33]. Hence, a contradiction.

	I	J	K	L
I	0	1	2	1
J	1	0	1	2
K	2	1	0	1
L	1	2	1	0

Theorem 3.6 (Schoenberg) Let $p\in(0,2]$ and $r\in(0,p/2]$. The spaces $(\mathbb{R}^m,d_p^r)$ and $(\mathcal{M}([a,b]),d_p^r)$, are isometrically embeddable in a Hilbert space.

Proof. See [342] for a proof.

This theorem covers some classes of both non-metric and metric spaces with the dissimilarity d_p, whose distances can be transformed by an appropriate power function such that they become embeddable in a Hilbert space. If a finite collection of points is considered, then Theorem 3.6 refers to a Euclidean embedding. This theorem justifies the validity of a common sense approach, where non-metric or non-Euclidean finite spaces (X,d) are transformed by a power transformation with the power $r\in(0,1)$. This is done in practice, since such a transformation may be capable of making the space metric or even Euclidean; see section 3.1.3. After such a transformation, also a premetric space may become semimetric.

Lemma* 3.7 (On embeddability of metric spaces into max-norm spaces) Any finite metric space (X,d) is l_∞-embeddable.

Proof. Assume that $X=\{x_1,x_2,\ldots,x_n\}$. Then a metric space (X,d) can be embedded in l_∞^n. Let $\phi: X\to\mathbb{R}^n$ be a mapping such that $\phi(x):=[d(x,x_1),d(x,x_2),\ldots d(x,x_n)]^T$. Denote $z_i:=\phi(x_i), i=1,\ldots,n$. Then, one has $d_\infty(\phi(x_i),\phi(x_j))=d_\infty(z_i,z_j)=\max_{1\le k\le n}|z_{ik}-z_{jk}|=\max_{1\le k\le n}|d(x_i,x_k)-d(x_j,x_j)|$. Thanks to the backward triangle inequality, Theorem 2.32, $|d(x_i,x_k)-d(x_j,x_k)|\le d(x_i,x_j)$ holds for any k and $d(x_i,x_j)$ on the right side is attained for $k:=j$. Hence, $\max_{1\le k\le n}|d(x_i,x_k)-d(x_j,x_k)|=d(x_i,x_j)$ and $d_\infty(\phi(x_i),\phi(x_j))=d(x_i,x_j)$. So, ϕ plays the role of an isometric embedding of (X,d) into l_∞^n. ■

Def. 3.8 (Cut semimetric) A partition of a set X into $V\subset X$ and $X\backslash V$ is called a *cut*. Such a cut defines a *cut semimetric* as:

$$\delta_V(x,y)=\mathcal{I}(|V\cap\{x,y\}|=1), \tag{3.1}$$

where $|\cdot|$ stands for the cardinality of the set and $\mathcal{I}$ is the indicator function.

A cut semimetric (metric without the definiteness condition required) serves for a further characterization of the l_1 distance. A distance d can be embedded in l_1^m, if d can be decomposed as a nonnegative linear combination of m cut metrics. Moreover, there exists a nonnegative measure space[1] such that d is the measure of the symmetric difference. Formally, one has:

[1] A measure space is defined as a triple $(\Omega,\mathcal{A},\mu)$, where Ω is a set, $\mathcal{A}$ is a σ-algebra on Ω (a collection of subsets $\mathcal{A}$

Theorem 3.9 (Characterization of l_1) Let $X := \{x_1, \ldots, x_n\}$ and $d_{ij} := d(x_i, x_j)$. Assume (X, d) is a finite metric space. For all $i, j = 1, 2, \ldots, n$, the following assertions are equivalent [88]:

1. $d_{ij} = d(x_i, x_j) = \sum_{V:V \subseteq X} \lambda_V \, \delta_V(x_i, x_j)$, where $\lambda_V \in \mathbb{R}_+^0$.
2. There exists a nonnegative measure (probability) space $(\Omega, \mathcal{A}, \mu)$ and $A_1, A_2, \ldots, A_n \in \mathcal{A}$ such that $d_{ij} = \mu(A_i \triangle A_j)$.
3. (X, d) is l_1^m-embeddable, i.e. there exist vectors $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \in \mathbb{R}^m$ for some integer $m \geq 1$ such that $d_{ij} = \sum_{k=1}^m |v_{ik} - v_{jk}|$.

The fact that the l_1 distance admits a decomposition by cut semimetrics does not simplify the process of determine whether a particular distance is l_1-embeddable or not. The difficulty of devising a polynomial algorithm for an l_1^m-embedding is due to the non-uniqueness of such a decomposition.

Distorted metric embeddings. Not every metric space can be embedded into the l_1 or l_2 spaces. This is, however, possible with some distortions. Let (X, d_X) and (Y, d_Y) be metric spaces. Then, a mapping $\phi : X \to Y$ is a Lipschitz embedding with the distortion $\kappa \geq 1$ if for all $x, y \in X$, one has $1/\kappa \, d_X(x, y) \leq d_Y(\phi(x), \phi(y)) \leq d_X(x, y)$. There has been a lot of research devoted to the problems of distorted embeddings into the l_1, l_2 and l_∞ spaces. For instance, a classical result is that any finite semimetric space of n points can be embedded into l_1 with a distortion of $O(\log_2(n))$ [39]. The problems of Lipschitz embeddings are beyond the scope of this dissertation, but they remain of interest for further study. A brief overview of important results concerning low-distortion embeddings with the algorithmic emphasis can be found e.g. in [203, 264].

3.1.2 Tree models for dissimilarities

A tree is a connected graph[2], where each pair of nodes is connected by a unique path. Representing dissimilarities by trees is an important issue in many scientific fields, like data analysis, mathematical psychology, historical linguistics and evolutionary biology; see e.g. [8, 216, 370]. A tree structure of the dissimilarity matrix allows for a natural interpretation of relations between the objects. It is a useful tool for understanding the data structure, especially for a smaller number of objects, where the results can be presented visually. Further on, tree models support the hierarchical clustering scheme based on proximities; see chapter 6.

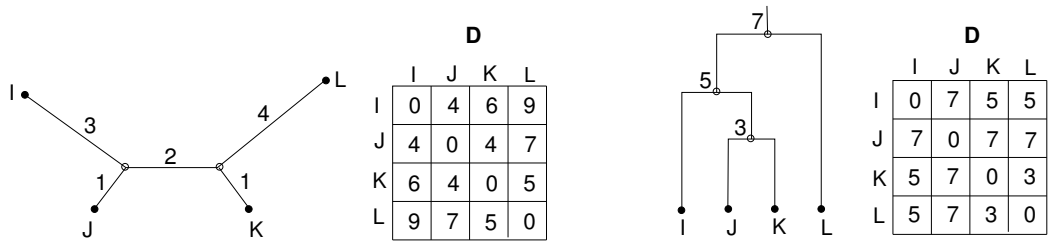
A key model is the additive tree model, which represents objects by nodes of a tree and defines dissimilarities as path lengths between two nodes, computed by the sum of the weights of the edges on the path. A special case of a rooted additive tree is an ultrametric tree, in which the distance from the root to every leaf is identical. The formal definitions follow.

Def. 3.10 Let (X, d) be a metric space. Assume that $X := \{1, x_2, \ldots, x_n\}$. Then d is described by the $n \times n$ distance matrix $D = (d_{ij})$, such that $d_{ij} = d(x_i, x_j)$. Additional constraints can be considered for all $x, y, z, u \in X$:

1. *ultrametric inequality:* $d(x, z) \leq \max\{d(x, y), d(y, z)\}$.
2. *four-point property:* $d(x, y) + d(z, u) \leq \max\{d(x, u) + d(y, z), d(x, z) + d(y, u)\}$.

such that $\Omega \in \mathcal{A}$ and if $A \in \mathcal{A}$, then $\Omega \backslash A \in \mathcal{A}$, and a union of any number of subsets of $\mathcal{A}$ belongs to $\mathcal{A}$, as well).

[2] A graph (V, E) consists of a set of nodes, $V = \{v_1, \ldots, v_n\}$ and a set of edges $E = \{(v_i, v_j) : v_j, v_j \in V\}$ connecting two nodes. In a directed graph, the pairs (v_i, v_j) in E are ordered, in an undirected graph, they are not. The degree of a node v_i is the number of edges incident in v_i. Nodes with degree one are called leaves. Nodes with larger degree are called internal. A path between two nodes v_{i_1} and v_{i_k} is a sequence of connected edges (v_{i_1}, v_{i_2}), $(v_{i_2}, v_{i_3}), \ldots (v_{i_{k-1}}, v_{i_k})$. A graph is connected if there is a path between any of the vertices. In a weighted graph, a nonnegative weight w_{ij} is associated to the edges (v_i, v_j). The length of a path is then a sum of the weights of the connected edges between the nodes v_{i_1} and v_{i_k}. The minimum path distance between any two nodes is defined as the minimum length of the paths connecting them.



(a) Additive dissimilarity tree (b) Ultrametric dissimilarity tree

Fig. 3.1: Tree examples. (a) Additive tree for the given distance matrix D satisfying the four-point property. (b) Ultrametric tree for the given ultrametric distance matrix D.

3. *hypermetric inequality:* $\mathbf{y}^T D\,\mathbf{y} \leq 0$ and $\mathbf{y} \in \mathbb{Z}^n$ such that $\mathbf{y}^T\mathbf{1} = \sum_i y_i = 1$. An infinite metric space X is hypermetric if the inequality holds for every finite subspace of X.
4. *negative type:* $\mathbf{y}^T D\,\mathbf{y} \leq 0$ for $\mathbf{y} \in \mathbb{R}^n$ such that $\mathbf{y}^T\mathbf{1} = 0$. An infinite metric space X is of negative type if the inequality holds for every finite subspace of X.

A metric space satisfying one of the above inequalities is called appropriately, e.g. a hypermetric space. Ultrametric space is also called non-Archimedean. The ultrametric and four-point properties can also be considered for premetric spaces.

The four-point property and ultrametric inequality are inherently connected to a tree structure of dissimilarity data.

Def. 3.11 (Additive, ultrametric trees) An *additive tree* is a connected, undirected graph where each pair of nodes is connected by a unique path. An *ultrametric tree* is an additive tree in which each leaf is equidistant (along the path) from the root.

Let (X, d) be a finite metric space and let D be the corresponding dissimilarity matrix. D defines a unique additive tree iff the four-point inequality holds for any quadruple from X. D defines a unique ultrametric tree iff the ultrametric inequality holds for any triplet from X. See Fig. 3.1 for an example. Formally, one has:

Def. 3.12 (Additive distance tree) [8] Let $D = (d_{ij})$ be an $n \times n$ symmetric distance matrix between the elements of X. Let T_D be an edge weighted tree with at least n nodes, where n distinct nodes of T_D are labeled by the elements of X. T_D is an *additive tree* for the matrix D if for every pair of the labeled nodes (i, j), the path from the node i to the node j has the total weight equal to d_{ij}. See Fig. 3.1(b) for an example.

Note that in an additive tree the root is not determined, and choosing different roots may suggest different interpretations. Basically, the root will distinguish two or three significant groups in the data. The root could be then chosen to enhance the interpretability of the data, but this requires some prior knowledge. Another possibility is to place the root at a node which minimizes the variance of the distances from the root to the leaf nodes, so it splits the data into homogeneous groups.

In a weighted graph, the *path metric* defines the shortest path, judged by the total sum of weights, between two nodes. A metric satisfying the four-point property is the path metric of nonnegative weighted trees.

Theorem 3.13 Every path metric, i.e. the shortest-path metric in a tree, is l_1-embeddable.

Proof. Let $T = (V, E)$ be a tree. Every edge $e = (v_k, v_l)$ introduces a partition of V into two sets V_{kl} and $V_{kl}^c = V \backslash V_{kl}$ such that $v_k \in V_{kl}$ and $v_l \in V_{kl}^c$. Then, the path metric of T can be decomposed as $d_T(v_i, v_j) = \sum_{(v_k,v_l)\in E} \delta_{V_{kl}}(v_k, v_l)$, where $\delta_{V_{kl}}$ is the cut metric, Def. 3.8. In case of a weighted tree, the $\delta_{V_{kl}}(v_k, v_l)$ are multiplied by the weights w_{kl}. By Theorem 3.9, since the path metric can be decomposed as a linear combination of cut metrics, then it is l_1-embeddable. ■

This proposition makes a connection between l_1-embeddability and a path metric of an additive tree. Hence, any dissimilarity matrix D which is l_1-embeddable can be represented by a path metric of an additive tree. This is then a discrete model of dissimilarity relations between the objects.

Def. 3.14 (Ultrametric dissimilarity tree) [8] Let $D=\{d_{ij}\}_{i,j=1}^{n}$ be an $n\times n$ symmetric dissimilarity matrix between the elements of X. An *ultrametric tree* for the matrix D, called also a *dendogram*, is a rooted tree T_D with the following properties (see also Fig. 3.1(a)):

1. T_D contains n leaves, each labeled by a unique element of X.
2. Each internal node is labeled by a dissimilarity values from D such that d_{ij} is the label of the least common ancestor of the leaves i and j.
3. Along any path from the root to a leaf, the numbers labeling internal nodes strictly decrease.
4. Each internal node has at least two children.

In practice, there might be no additive tree to represent the dissimilarity data D, since there might be no path metric coinciding exactly with D. A solution can be offered by finding a tree which models the given dissimilarities as well as possible in terms of the path distances. Such a tree metric D_T should provide the best approximation of D under the error criteria defined e.g. by the norms l_1, l_2 or l_∞. This is a formulation of a numerical taxonomy problem, which has received a great deal of attention over the years; see e.g. [8, 216]. The additive or ultrametric tree fitting problems are known to be NP-hard under the l_1 and l_2 norms [64, 216, 370]. In case of the l_∞ norm, the same holds for an additive tree [1], however the optimal ultrametric tree can be computed in a polynomial time [123]. There exists a number of other methods trying to construct either an ultrametric or additive tree such that the path distance approximates the given distance as well as possible. See [8, 167, 216, 370] for general references or [1, 66, 123, 139, 140, 373, 374], for more specific algorithms. See also section 6.3 for more discussion.

3.1.3 Transformations in semimetric spaces

Some transformations are considered, which either preserve the (semi)metric properties or, in particular cases, change a dissimilarity measure into a metric.

Theorem* 3.15 ((Semi)metric transformation) Let (X,d) be a semimetric space. Then the composition of mappings, $f\circ d$, is also semimetric if $f:\mathbb{R}_+^0\to\mathbb{R}_+^0$ is a non-decreasing and concave function such that $f(0)=0$. If (X,d) is a metric space, then $f\circ d$ is metric, if additionally f is positive on $\mathbb{R}_+$. Such f will be called a *(semi)metric transformation*.

Proof. Here, only the metric space is considered, since the proof for the semimetric space follows directly. Let (X,d) be a metric space. Since $f(x)>0$ for $x>0$ and $f(0)=0$, then $f\circ d$ directly fulfills the positivity, reflexivity and symmetry constraints; see Def. 2.30. Therefore, it suffices to prove the triangle inequality only. Let $d_1=d(x,y)$, $d_2=d(y,z)$ and $d_3=d(z,x)$ for any $x,y,z\in X$. Assume that $d_1+d_2\geq d_3$ holds for each triplet d_1,d_2,d_3. Since $f(d_1+d_2)\geq f(d_3)$, $f\uparrow$, one suffices to show that $f(d_1)+f(d_2)\geq f(d_1+d_2)$. The inequality $f(d_1)+f(d_2)\geq f(d_3)$ is then straightforward. Based on the concavity of f, we have $f(\alpha t+(1-\alpha)u)\geq\alpha f(t)+(1-\alpha)f(u)$ for all $\alpha\in[0,1]$ and all $t,u\geq 0$. Let $\alpha=\frac{d_1}{d_1+d_2}$, $u=d_1+d_2$ and $t=0$. Then, we have:

$$f(d_1)=f(\tfrac{d_1}{d_1+d_2}\,(d_1+d_2)+\tfrac{d_2}{d_1+d_2}\,0)\geq\tfrac{d_1}{d_1+d_2}\,f(d_1+d_2)+\tfrac{d_2}{d_1+d_2}f(0)=\tfrac{d_1}{d_1+d_2}\,f(d_1+d_2)$$

Similarly, $f(d_2)>\frac{d_2}{d_1+d_2}\,f(d_1+d_2)$. Therefore, $f(d_1)+f(d_2)\geq f(d_1+d_2)$, which finishes the proof. ■

Above, we required that f is non-decreasing and concave. Note that instead of concavity, in fact, the subadditive property of f is needed, i.e. $f(x_1+x_2)\leq f(x_1)+f(x_2)$ for all x_1 and x_2 in the domain of f.

Corollary⋆ 3.16 Let (X,d) be a metric space. Then $(X, f\circ d)$ is also metric for $f:\mathbb{R}^0_+ \to \mathbb{R}^0_+$ defined as one of the following functions:

(1) $f_1(x)=c\,x, \quad c>0.$
(2) $f_2(x)=x+c\,\mathcal{I}(x>0), \quad c>0.$
(3) $f_3(x)=\min\{c,x\}, \quad c>0.$
(4) $f_4(x)=x^r, \quad 0<r\leq 1.$
(5) $f_5(x)=\frac{x}{x+c}, \quad c>0.$
(6) $f_6(x)=\text{sigm}\,(x):=\frac{2}{1+\exp(-x/\sigma)}-1$ and $\sigma>0.$
(7) $f_7(x)=\log(1+x)$

Proof. Cases (1) – (3) are trivial. So, we focus on the remaining cases. Let $k>3$. It is straightforward to verify that $f_k(0)=0$ and f_k is monotonically growing. Now, we prove that f_k is concave by showing that the second derivative of f_k is negative for positive x[125]. Thus, we have $f_3''(x)=r\,(r-1)\,x^{r-2}<0$, since $r\in(0,1]$, $f_4''(x)=-2\,c^2/(x+c^2)^3<0$, $f_5''(x)=2/\sigma^2\,(\exp(-2x/\sigma)-\exp(-x/\sigma))/(1+\exp(-x/\sigma^3)<0$, since $\exp(-x)$ is a monotonically decreasing function and $f_6''(x)=-1/(1+x)^2<0$. By Theorem 3.15, $(X, f\circ d)$ is metric. ■

Theorem 3.17 (Blumenthal) Let (X,d) be a metric space and let $f_r(x)=x^r$ be a metric transform with $r\in(0,1/2]$. Then d^r is metric and any four points of (X,d^r) can be isometrically embedded in a Euclidean space.

Proof. See [32] for a proof.

Note that for any metric, every three points can be isometrically embedded in a Euclidean space, which follows from the triangle inequality; see also section 3.2. The above theorem explains that the power transformation $f_r(x)=x^r$ with $0<r\leq 1/2$ makes the metric 'more' Euclidean, since the embeddability holds for any four points.

Corollary⋆ 3.18 Let $p\in(0,1)$. Then, the space $(\mathbb{R}^m, d_p^p)$ is metric.

Proof. Assume that $\mathbf{x},\mathbf{y},\mathbf{z}\in\mathbb{R}^m$. Note that $d_p^p(\mathbf{x},\mathbf{y})=\sum_{i=1}^m |x_i-y_i|^p$. It suffices to show that the triangle inequality holds, since $(\mathbb{R}^m, d_p)$ with $p\in(0,1)$ is quasimetric, as shown in Example 2.39. Based on the Minkowski inequality[3] $\sum_{i=1}^m |a_i+b_i|^p\leq\sum_{i=1}^m |a_i|^p+\sum_{i=1}^m |b_i|^p$, we have $\sum_{i=1}^m |x_i-z_i|^p=\sum_{i=1}^m |(x_i-y_i)+(y_i-z_i)|^p\leq\sum_{i=1}^m |x_i-y_i|^p+\sum_{i=1}^m |y_i-z_i|^p$. The latter inequality is equivalent to $d_p^p(\mathbf{x},\mathbf{z})\leq d_p^p(\mathbf{x},\mathbf{y})+d_p^p(\mathbf{y},\mathbf{z})$. Hence, d_p^p is metric. ■

Observation⋆ 3.19 Note that since $(\mathbb{R}^m, d_p^p)$, with $p\in(0,1)$ is a metric space, then based on Corollary 3.16, $(\mathbb{R}^m, d_p^r)$ with $r\in(0,p]$ is a metric space as well.

Def. 3.20 Let G denote a set of functions $g(x;p)$ of one real variable $x\geq 0$ and one real parameter $p>0$ such that $g(0;p)=0$ and $g(x;p)$ is a continuous strictly increasing function of x. Moreover, g is such that for any $x>0$ and any real $\varepsilon>0$, there exists a such that $p\leq a\Rightarrow |g^2(x;p)-1|\leq\varepsilon$ [70].

Example 3.21 Examples of the functions from G are [70]:

- Power function: $g(x;p)=x^p$ with $\sup x<\infty$. Let $\varepsilon<1$. Then, $a>0$ for $x=1$ and $a=[\log(1+\text{sign}(x-1)\,\varepsilon)]/(2\log(x))$, otherwise.
- Weibull function: $g(x;p)=1-\exp(-x^r/p)$ with $r>0$.
  Let $\varepsilon<1$. Then, $a=-x^r/\log(1-\sqrt{1-\varepsilon})$.

[3] The proof of this inequality relies on the following fact. Let $a\geq b\geq 0$ and $p\in(0,1]$. Consider $g(x)=(1+c^{\frac{1}{x}})^x$ with $c\in(0,1]$ and $x\geq 1$. $g\uparrow$ since $g'(x)=-\frac{c^{\frac{1}{x}}}{x^c}\log(1+c^{\frac{1}{x}})\log(c)g(x)\geq 0$. Therefore, g has the minimum value of $1+c$ at $x=1$. For $x=\frac{1}{p}$ and $c=\frac{b}{a}\leq 1$, we have: $(a^p+b^p)^{\frac{1}{p}}=a\,g(\frac{1}{p})\geq a\,g(1)=a+b$. Hence, $(a+b)^p\leq a^p+b^p$.

Theorem 3.22 (Courrieu) Let (X,ρ) be an n-element finite quasimetric space. Consider a function $g \in G$. Then, the following statements hold:

1. There exists a real $\alpha(X) > 0$ such that for any positive $p \leq \alpha(X)$, the space $(X, g(\cdot;p)\circ\rho)$ is isometrically embeddable in a Euclidean space of a dimensionality $\leq n-1$.
2. There exists a real $\beta(X)$ such that $0 < \beta(X) \leq \alpha(X)$ and for any positive $p \leq \beta(X)$ the space $(X, g(\cdot;p)\circ\rho)$ is isometrically embeddable in a Euclidean space of a dimensionality $n-1$.

Proof. See [70] for a proof.

The above theorem explains that any finite quasimetric space can be transformed into a Euclidean space by a suitable function $g(\cdot;p)$. It does not, however, explain how a proper parameter p can be determined. This depends on the set X and cannot be captured by a general formula. In practice, $p < 1$. For p approaching zero, the quasimetric space resembles more and more a discrete metric space; see Example 2.31. This means that the structure in the data is weakened in the embedded Euclidean space, since the points move towards the corners of an equilateral polytope. Still for any $p > 0$, some structural information is present.

3.1.4 Direct product spaces

Direct product spaces allow one for a construction of a new space by combining two (or more) spaces; see also section 2.2. In the context of generalized metric spaces, this means that given two (or more) such finite spaces describing the same objects, a new dissimilarity measure can be created by their summation or by maximum operator. Now, some conclusions can be drawn for the combined spaces.

Theorem* 3.23 Let (X,d_X) and (Y,d_Y) be metric spaces. The direct product space $(X\times Y, d_X \bullet d_Y)$ defined as $(d_X \bullet d_Y)((x_1,y_1),(x_2,y_2)) = d_X(x_1,x_2) \bullet d_Y(y_1,y_2)$, where $x_i \in X$ and $y_i \in Y$, $i = 1,2$ and $\bullet$ is either the sum or max operator, is metric.

Proof. The proof is straightforward by checking the conditions of Def. 2.30.

Theorem* 3.24 Let (X,ρ_X) and (Y,ρ_Y) be generalized metric spaces. Let the direct product space $(X\times Y, \rho_X \bullet \rho_Y)$ be defined such that $(\rho_X \bullet \rho_Y)((x_1,x_2),(y_1,y_2)) = \rho_X(x_1,x_2) + \rho_Y(y_1,y_2)$ $x_i \in X$ and $y_i \in Y$, $i = 1,2$. Then the space $(X\times Y, \rho_X \bullet \rho_Y)$ is

(1) l_1-embeddable, iff (X,ρ_X) and (Y,ρ_Y) are l_1-embeddable.

(2) is hypermetric, iff (X,ρ_X) and (Y,ρ_Y) are hypermetric.

(3) is of negative type, iff (X,ρ_X) and (Y,ρ_Y) are both of negative type.

Proof. Let $Z = X\times Y$ and $\rho = \rho_X \bullet \rho_Y$.
(1) $\Rightarrow$ Assume that (Z,ρ) is l_1-embeddable. Then $\rho((x_1,y_1),(x_2,y_2)) = \rho_X(x_1,x_2) + 0 + \rho_Y(y_1,y_2) + 0 = \rho_X(x_1,x_2) + \rho_X(x_2,x_2) + \rho_Y(y_1,y_1) + \rho_Y(y_1,y_2) = \rho((x_1,y_1),(y_1,x_2)) + \rho((y_1,x_2),(x_2,y_2))$. Consequently, (Z,ρ) is $(X\times\{x_2\},\rho_X)\times(\{y_1\}\times Y,\rho_Y)$, which is equivalent to $(X,\rho_X)\times(Y,\rho_Y)$. Hence, (X,ρ_X) and (Y,ρ_Y) are l_1-embeddable.

$\Leftarrow$ Assume that (X,ρ_X) and (Y,ρ_Y) are l_1-embeddable. Let ϕ_X and ϕ_Y denote the l_1-embedding of (X,ρ_X), and (Y,ρ_Y), correspondingly. Then, the embedding ϕ of (Z,ρ) into l_1 can be obtained by $\phi(x,y) = [\phi_X(x)\ \ \phi_Y(y)]$. Since $\rho = \rho_X \bullet \rho_Y$, then $\rho((x_1,x_2),(y_1,y_2)) = \rho_X(x_1,x_2) + \rho_Y(y_1,y_2) = ||\phi_X(x_1) - \phi_X(x_2)||_1 + ||\phi_Y(y_1) - \phi_Y(y_2)||_1 = ||[\phi_X(x_1)\ \ \phi_Y(y_1)] - [\phi_X(x_2)\ \ \phi_Y(y_2)||_1 = ||\phi(x_1,y_1) - \phi(x_2,y_2)||_1$. Hence, $(X\times Y,\rho)$ is l_1-embeddable.

(2) $\Rightarrow$ Let (Z,ρ) be hypermetric. This means that for $y_1 \in Y$, $(X\times\{y_1\},\rho)$ is hypermetric as well. Then, $\rho((x_1,y_1),(x_2,y_2)) = \rho_X(x_1,x_2)$, so (X,ρ_X) is hypermetric. The same reasoning holds for (Y,ρ_Y).
$\Leftarrow$ Let (X,ρ_X) and (Y,ρ_Y) be hypermetric spaces. Let $z \in \mathbb{Z}^Z$ satisfy $\sum_{(x_1,y_1)\in Z} z(x_1,y_1) = 1$. De-

fine $u \in \mathbb{Z}^X$ and $\mathbf{v} \in \mathbb{Z}^Y$ such that $u(x_1) := \sum_{y_1 \in Y} z(x_1, y_1)$ and $v(y_1) := \sum_{x_1 \in X} z(x_1, y_1)$. Then, $\sum_{x_1 \in X} u(x_1) = \sum_{y_1 \in Y} v(y_1) = 1$. Then, $\sum_{\{(x_1,x_2) \in X \wedge (y_1,y_2) \in Y\}} z(x_1, y_1)\, z(x_2, y_2)\, \rho((x_1, x_2), (y_1, y_2)) = \sum_{(x_1,x_2) \in X} u(x_1)\, u(y_1)\, \rho_X(x_1, y_1) + \sum_{(y_1,y_2) \in Y} v(y_1)\, v(y_2)\, \rho_Y(y_1, y_2) \leq 0$. Hence (Z, ρ) is hypermetric.

(3) The proof is similar to the one above. ■

If $X = Y$, then the above theorem states that the summation of distances $\rho_X + \rho_Y$ preserves the l_1-embeddability, hypermetric and negative type properties. Moreover, we also know that for (Z, ρ) of negative type, $(Z, \rho^{1/2})$ is l_2-embeddable; see Theorem 3.31. This means that if (Z, ρ_X) and (Z, ρ_Y) are l_2-embeddable, then $(Z, (\rho_X^2 + \rho_Y^2)^{1/2})$ is also l_2-embeddable.

3.2 Properties of dissimilarity matrices

This section discusses some properties of dissimilarity matrices with respect to metric behavior, metric transformations and Euclidean embeddings, as well as the corrections of dissimilarities imposing the metric or Euclidean constraints. This is done explicitly for dissimilarity matrices, since our learning algorithms will be later based on such finite representations. Therefore, the issues discussed here prepare us for the analysis of dissimilarity data. A substantial part of the presented theory comes from Gower [171].

Let us consider an $n \times n$ dissimilarity matrix $D = (d_{ij})$ and an $n \times n$ similarity matrix $S = (s_{ij})$ for $i, j = 1, \ldots, n$. In all discussions below, we assume that both D and S are nonnegative and D has a zero diagonal. Concerning the notation, a few points are important. $\mathbf{e}_j \in \mathbb{R}^n$ is a standard basis vector (a vector of all zeros except for $e_j = 1$), $\mathbf{1}$ stands for a vector of all ones and $*$ denotes the Hadamard (element-wise) operation on matrices. So, $A * B$ denotes the Hadamard matrix product, i.e. if $C = A * B$ and $C = (c_{ij})$, then $c_{ij} = a_{ij} b_{ij}$, $C^{*k} = (c_{ij}^k)$ is the Hadamard power. Moreover, if a vector representation X is mentioned, we follow the convention from pattern recognition, where vectors are placed in *rows* of X, i.e. $X := [\mathbf{x}_1^T; \ldots; \mathbf{x}_n^T]$.

Def. 3.25 (Metric for D) Let D be a symmetric dissimilarity matrix with positive off-diagonal elements. D is metric, if the triangle inequality $d_{ij} + d_{jk} \geq d_{ik}$ holds for all triplets (i, j, k).

Observation* 3.26 Let D be a semimetric, i.e. the definiteness axiom may not hold.

(1) if $d_{ij} = \varepsilon$, then $|d_{ik} - d_{jk}| \leq \varepsilon$ for any k.

(2) if $d_{ij} = 0$, then $d_{ik} = d_{jk}$ for any k.

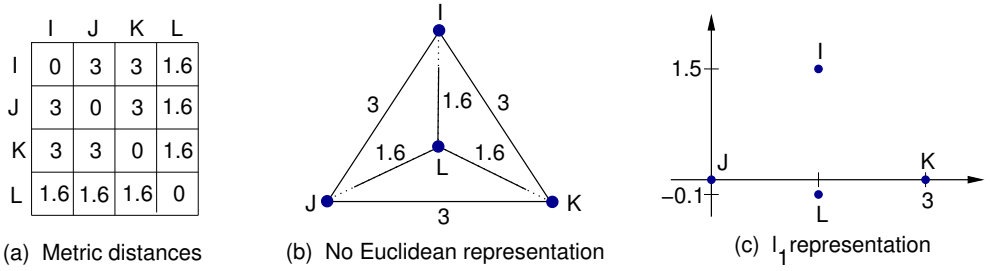
Proof. (1) Making use of the triangle inequality the following inequalities hold: $d_{ik} + d_{ij} \geq d_{kj}$ and $d_{jk} + d_{ji} \geq d_{ki}$ for any k. Based on the symmetry condition and $d_{ij} = \varepsilon$, one obtains: $d_{ik} + \varepsilon \geq d_{jk}$ and $d_{jk} + \varepsilon \geq d_{ik}$, which after a simple transformation, gives $d_{ik} + \varepsilon \geq d_{jk} \geq d_{ik} - \varepsilon$ and finally $|d_{ik} - d_{jk}| \leq \varepsilon$.

(2) Trivial, by the same reasoning as in (1). ■

These properties of metric dissimilarities are important from a practical point of view. Basically, if two objects are similar, i.e. the dissimilarity between them is small (close to zero or equal to zero), then any other object will have a similar relation to them both. That means that one of them can become a prototype to represent both of them. This property enables a construction of the approximate nearest neighbor searches in a Euclidean space; see e.g. [273].

Observation* 3.27 D is a metric if every triplet is Euclidean.

Any metric triplet d_{ij}, d_{ik} and d_{kj} is Euclidean, i.e. it constitutes a Euclidean triangle. However, for $n > 3$, not every $n \times n$ metric distance matrix D has a Euclidean representation. A counterexample is given by a 4×4 matrix D presented in Fig.3.2. There, an l_1^2-embedding can be found such that D can be represented by points in a 2-dimensional space with the city block distances equal to d_{ij}.



(a) Metric distances (b) No Euclidean representation (c) l_1 representation

Fig. 3.2: An example of (a) metric distances with (b) no Euclidean representation and (c) a possible l_1 representation. In order to get a $2D$ or $3D$ Euclidean representation, the distances from the point L to other points should be at least equal to $\frac{2}{3}\frac{3\sqrt{3}}{2}=\sqrt{3}$. They are smaller, so no Euclidean embedding exists.

Corollary 3.28 If D is quasimetric (the triangulation inequality does not hold), then the matrix $D'=D+c\,(\mathbf{1}\mathbf{1}^T-I)$, where $c\geq \max_{p,q,r}|d_{pq}+d_{pr}-d_{qr}|$ is metric.

Proof. It suffices to show that D' fulfills the triangle inequality, since other properties are easily checked. Let (i,j,k) be a triplet for which the triangulation inequality does not hold, i.e. $d_{ij}+d_{jk}<d_{ik}$. Since $c\geq \max_{p,q,r}|d_{pq}+d_{pr}-d_{qr}|$, then $c\geq |d_{ij}+d_{jk}-d_{ik}|$. Now, we should prove that $(d_{ij}+c)+(d_{jk}+c)\geq(d_{ik}+c)$. Note that $d_{ij}+d_{jk}+c\geq d_{ij}+d_{jk}+|d_{ij}+d_{jk}-d_{ik}|=d_{ik}$, since $|z|=-z$ for $z<0$. Because c is nonnegative, then $(d_{ij}+c)+(d_{jk}+c)\geq(d_{ik}+c)$, which finishes the proof. ■

If c is relatively small, then the dissimilarity matrix D is only slightly non-metric. If, however, c is large, then the analysis should take into account its non-metric properties. The triangle inequality is the most burdensome to check; in the worst case, all the triplets needs to be investigated.

Observation* 3.29 An important question refers to transformations of a metric dissimilarity such that the metric properties are preserved. From Theorem 3.15, we already know that if D is metric, then $D_f=(f\,(d_{ij}))$ is a metric as well for f being a non-decreasing and concave function such that $f(0)=0$ and $f(x)>0$ for $x>0$. Consequently, if D is metric, then for $c>0$ the dissimilarity matrices defined as: $(c\,d_{ij})$, $(d_{ij}+c\,(1-\delta_{ij}))$, $(\min\{1,d_{ij}\})$, (d_{ij}^r) with $r\in(0,1]$, $(d_{ij}/(d_{ij}+c))$, $(\mathrm{sigm}\,(d_{ij}))$, $(\log(1+d_{ij}))$ are also metric; see Corollary 3.16.

Below we present some results, mostly related to the Euclidean behavior of a distance matrix and its vector representation. A more thorough explanation can be found in section 3.3.

Def. 3.30 (Euclidean behavior) An $n\times n$ dissimilarity matrix $D=(d_{ij})$ is Euclidean if it can be embedded in a Euclidean space $(\mathbb{R}^m,d_E)$, where $m\leq n$. This means that a configuration $\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_n\}$ can be determined in $\mathbb{R}^m$ such that $||\mathbf{x}_i-\mathbf{x}_j||_2=d_{ij}$.

Theorem 3.31 (Test I for Euclidean behavior) A symmetric $n\times n$ matrix D with a zero diagonal is Euclidean iff $D^{*2}=(d_{ij}^2)$ is conditionally negative definite (cnd), i.e. $\mathbf{z}^TD^{*2}\mathbf{z}\leq 0$ for all vectors $\mathbf{z}\in\mathbb{R}^n$ such that $\mathbf{z}^T\mathbf{1}=0$. Equivalently, a symmetric $n\times n$ matrix D with a zero diagonal is Euclidean iff $-D^{*2}$ is conditionally positive definite (cpd).

Proof. The proof makes use of equivalent transformations. Let D^{*2} be a square Euclidean matrix, i.e. $d_{ij}^2=||\mathbf{x}_i-\mathbf{x}_j||^2$ in a Euclidean space $\mathbb{R}^k$. Let $\mathbf{g}=[||\mathbf{x}_1||^2,||\mathbf{x}_2||^2,\ldots,||\mathbf{x}_n||^2]^T$. Then, $\mathbf{z}^TD^{*2}\mathbf{z}=\sum_{i,j}z_iz_j||\mathbf{x}_i-\mathbf{x}_j||^2=\sum_{i,j}z_iz_j||\mathbf{x}_i||^2-2\sum_{i,j}z_iz_j\langle\mathbf{x}_i,\mathbf{x}_j\rangle+\sum_{i,j}z_iz_j||\mathbf{x}_j||^2=\mathbf{z}^T\mathbf{g}\,\mathbf{1}^T\mathbf{z}-2||\sum_i z_i\mathbf{x}_i||^2+\mathbf{z}^T\mathbf{1}\,\mathbf{g}^T\mathbf{z}=2\,\mathbf{z}^T\mathbf{1}\,\mathbf{g}^T\mathbf{z}-2\,||\mathbf{z}^TX||^2$. Note that $||\mathbf{z}^TX||^2\geq 0$ and $\mathbf{z}^T\mathbf{1}\mathbf{g}^T\mathbf{z}\geq 0$, since $\mathbf{1}\,\mathbf{g}^T$ is a positive semidefinite matrix. Since $\mathbf{z}^TD^{*2}\mathbf{z}=2\,(\mathbf{z}^T\mathbf{1}\,\mathbf{g}^T\mathbf{z}-||\mathbf{z}^TX||^2)$, then to assure that $\mathbf{z}^TD^{*2}\mathbf{z}\leq 0$, one should require that $\mathbf{z}^T\mathbf{1}=0$, which finishes the proof. ■

Theorem 3.32 (Test II for Euclidean behavior) D is Euclidean iff the matrix $D_s=J_sD^{*2}J_s^T$ with $J_s=(I-\mathbf{1}\mathbf{s}^T)$ is negative semidefinite (nsd) for $\mathbf{s}^T\mathbf{1}=1$. Equivalently, D is Euclidean iff the matrix

$S_s = -\frac{1}{2} D_s$ is positive semidefinite (psd) for $\mathbf{s}^T\mathbf{1} = 1$ [171].

Proof. $\Rightarrow$ For any $\mathbf{x} \in \mathbb{R}^{k\times 1}$, $\mathbf{z} = (I - \mathbf{1}\mathbf{s}^T)\,\mathbf{x}$ is orthogonal to $\mathbf{1}$, i.e. $\mathbf{z}^T\mathbf{1} = \mathbf{x}^T(I - \mathbf{s}\mathbf{1}^T)\,\mathbf{1} = \mathbf{x}^T\mathbf{1} - \mathbf{x}^T\mathbf{1}\,\mathbf{s}^T\mathbf{1} = \mathbf{x}^T\mathbf{1} - \mathbf{x}^T\mathbf{1}\,1 = 0$. Then, based on Theorem 3.31, $\mathbf{z}^T D^{*2}\mathbf{z} \le 0$, which yields $\mathbf{x}^T[(I - \mathbf{1}\mathbf{s}^T)\,D^{*2}(I - \mathbf{s}\mathbf{1}^T)]\mathbf{x} \le 0$. This proves that D_s is negative semidefinite.

$\Leftarrow$ Let $\mathbf{z}^T\mathbf{1} = 0$ and D_s be nsd. Then, $0 \ge \mathbf{z}^T D_s\,\mathbf{z} = \mathbf{z}^T[(I - \mathbf{1}\mathbf{s}^T)\,D^{*2}(I - \mathbf{s}\mathbf{1}^T)]\mathbf{z} = \mathbf{z}^T D^{*2}\mathbf{z} - 2\,\mathbf{z}^T\mathbf{1}\mathbf{s}^T D^{*2} + \mathbf{z}^T\mathbf{1}\mathbf{s}^T D^{*2}\mathbf{s}\mathbf{1}^T\mathbf{z} = \mathbf{z}^T D^{*2}\mathbf{z}$. This means that D^{*2} is cnd and by Theorem 3.31, D is Euclidean. ■

Observation⋆ 3.33 D is Euclidean iff $D_c = J\,D^{*2}J$ is nsd. $J := (I - \frac{1}{n}\,\mathbf{1}\mathbf{1}^T)$ is known as the *centering matrix*. This is a special case of Theorem 3.32 for $\mathbf{s} = \frac{1}{n}\mathbf{1}$. Another special case holds for $\mathbf{s} = \mathbf{e}_i$, where $\mathbf{e}_i \in \mathbb{R}^n$ is a standard basis vector.

Observation⋆ 3.34 Let D be an $n\times n$ dissimilarity matrix. If Theorem 3.32 is true for a particular $\mathbf{s}$, e.g. $\mathbf{s} = \frac{1}{n}\mathbf{1}$, then it is true for any $\mathbf{s}$ such that $\mathbf{s}^T\mathbf{1} = 1$. See also section 3.2.1.

Theorem 3.35 (Vector representation) For an $n\times n$ Euclidean distance matrix D, a vector representation of the distances in $\mathbb{R}^m$ is given by the rows of the $n\times m$ matrix X, $m \le n$, where $X\,X^T = -\frac{1}{2}\,(I - \mathbf{1}\mathbf{s}^T)D^{*2}(I - \mathbf{s}\mathbf{1}^T)$ and $\mathbf{s}^T\mathbf{1} = 1$.

Proof. Indirectly, the goal is to prove that the matrix $S_s = -\frac{1}{2}\,(I - \mathbf{1}\mathbf{s}^T)D^{*2}(I - \mathbf{s}\mathbf{1}^T)$ is a matrix of inner products (Gram matrix) of a vector representation X in $\mathbb{R}^m$. Let $\mathbf{h} = D^{*2}\mathbf{s} - \frac{1}{2}\,\mathbf{1}\mathbf{s}^T D^{*2}\mathbf{s}$ (it can be easily check that $\mathbf{h}$ describes a diagonal of S_s; see also section 3.2.1). Then, after straightforward mathematical operations, we can express S_s as follows $S_s = -\frac{1}{2}\,(I - \mathbf{1}\mathbf{s}^T)\,D^{*2}(I - \mathbf{s}\mathbf{1}^T) = -\frac{1}{2}\,(D^{*2} - \mathbf{h}\mathbf{1}^T - \mathbf{1}\mathbf{h}^T)$. Note that since $(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{1} = 0$, then $\mathbf{h}\mathbf{1}^T(\mathbf{e}_i - \mathbf{e}_j) = 0$. Consequently, $(\mathbf{e}_i - \mathbf{e}_j)^T S_s\,(\mathbf{e}_i - \mathbf{e}_j) = -\frac{1}{2}\,(\mathbf{e}_i - \mathbf{e}_j)^T D^{*2}\,(\mathbf{e}_i - \mathbf{e}_j) = -\frac{1}{2}(d_{ii}^2 + d_{jj}^2 - 2\,d_{ij}^2) = d_{ij}^2$, since D^{*2} has a zero diagonal ($d_{ii} = 0$ for any i). On the other hand, according to Def. 3.30, for an Euclidean distance D, there exist a vector configuration $X = [\mathbf{x}_1^T; \mathbf{x}_2^T; \ldots, \mathbf{x}_n^T]$ such that $d_{ij}^2 = ||\mathbf{x}_i - \mathbf{x}_j||_2^2$. So, we can further write $d_{ij}^2 == ||X^T\mathbf{e}_i - X^T\mathbf{e}_j||_2^2 = ||X^T(\mathbf{e}_i - \mathbf{e}_j)||_2^2 = (\mathbf{e}_i - \mathbf{e}_j)^T X X^T(\mathbf{e}_i - \mathbf{e}_j)$. Since, $d_{ij}^2 = (\mathbf{e}_i - \mathbf{e}_j)^T S_s\,(\mathbf{e}_i - \mathbf{e}_j)$, then X can be related to S_s as $S_s = X X^T$. Note that the dimensionality m of X is obtained as the rank of S_s. ■

Theorem⋆ 3.36 (Test III for Euclidean behavior) Let D be an $n\times n$ non-zero symmetric matrix with the zero diagonal. D is Euclidean iff $\begin{bmatrix} -D^{*2} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}$ has exactly one negative eigenvalue.

Proof. This theorem and its proof follows directly from the considerations of Chabrillac and Crouzeix on semidefinitness of quadratic forms [58]. Given an $n\times n$ real symmetric matrix A, they show that requiring that $\mathbf{z}^T A\mathbf{z} \ge 0$ for all $\mathbf{z}$ such that $B^T\mathbf{z} = 0$ is equivalent to stating that the matrix $\begin{bmatrix} A & B \\ B^T & \mathbf{0} \end{bmatrix}$ has exactly $r = rank(B)$ negative eigenvalues. By Theorem 3.31, D is Euclidean iff D^{*2} is cnd, that is $\mathbf{z}^T(-D^{*2})\mathbf{z} \ge 0$ for all $0 = \mathbf{z}^T\mathbf{1} = \mathbf{1}^T\mathbf{z}$. By substituting $A = -D^{*2}$ and $B = \mathbf{1}$, we get that $\begin{bmatrix} -D^{*2} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix}$ has exactly one negative eigenvalue. ■

Observation⋆ 3.37 If an $n\times n$ symmetric matrix D with a zero diagonal is Euclidean, then $-D^{*2}$ has exactly one negative eigenvalue. The reverse does not hold.

Proof. Assume that $\{\lambda_i\}_{i=1}^n$ are the eigenvalues of $-D^{*2}$. Since the trace of $-D^{*2}$ is, on the one hand, a sum of the diagonal elements and, on the other hand, the sum of eigenvalues, then $\sum_{i=1}^n \lambda_i = 0$. It follows that $-D^{*2}$ must have at least one negative eigenvalue. By Theorem 3.31, D is Euclidean iff $-D^{*2}$ is cpd. From Chabrillac and Crouzeix [58] follows that a $n\times n$ cpd matrix has at most one negative eigenvalue. Hence, $-D^{*2}$ has exactly one negative eigenvalue.

To prove that the reverse does not hold, consider a nonnegative symmetric matrix $A = \begin{bmatrix} 0 & 3 & 1 \\ 3 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$. One may found out that the eigenvalues of $-A^{*2}$ are $\{9, 0.2170, -9.2170\}$. Hence, $-A^{*2}$ has exactly one negative eigenvalue. However, A cannot be Euclidean since the triangle inequality is not fulfilled as $1 + 1 < 3$. ■

Theorem* 3.38 (Constructing D from S) Let S be a psd similarity matrix.

(1) For $S := (s_{ij})$ with the elements s_{ij} obeying $0 \leq s_{ij} \leq 1$ and $s_{ii} = 1$, the dissimilarity matrix $D_1 = (\mathbf{1}\mathbf{1}^T - S)^{*\frac{1}{2}}$ is Euclidean. Also the matrix $D_2 = (\mathbf{1}\mathbf{1}^T - S)$ is Euclidean.

(2) The dissimilarity matrix $D = (d_{ij})$ with $d_{ij} = (s_{ii} + s_{jj} - 2s_{ij})^{1/2}$ is Euclidean.

If S is not psd, then the corresponding dissimilarity matrices shown above are not Euclidean, yet they can still be constructed.

Proof. By Theorem 3.31, it is sufficient to prove that D_1^{*2} and D_2^{*2} are cnd for all $\mathbf{z}$ such that $\mathbf{z}^T\mathbf{1} = 0$.

(1) Let $\mathbf{z}^T\mathbf{1} = 0$. Since $D_1^{*2} = \mathbf{1}\mathbf{1}^T - S$, then $\mathbf{z}^T D_1^{*2}\mathbf{z} = \mathbf{z}^T\mathbf{1}\mathbf{1}^T\mathbf{z} - \mathbf{z}^T S\mathbf{z} = 0 - \mathbf{z}^T S\mathbf{z} \leq 0$. The latter inequality holds since S is psd, i.e. $\mathbf{z}^T S\mathbf{z} \geq 0$ for any $\mathbf{z}$. Hence, D_1^{*2} is cnd. Consequently, D_1 is Euclidean. One also has that $D_2^{*2} = \mathbf{1}\mathbf{1}^T - 2S + S^{*2}$. Then, $\mathbf{z}^T D_2^{*2}\mathbf{z} = 0 - \mathbf{z}^T(2S - S^{*2})\,\mathbf{z}$. Now, one needs to require that $\mathbf{z}^T(2S - S^{*2})\mathbf{z} \geq 0$. This holds if $(2S - S^{*2})$ is psd. To show that, we will make use of two theorems. The Schur theorem states that the Hadamard product of psd matrices is psd [22]. The other theorem says that a square matrix A is psd iff it has a dominant diagonal, i.e. $a_{ii} \geq \sum_{j\neq i} a_{ij}$. In our case, since S and S^{*2} are psd (S^{*2} is psd by the Schur theorem), then the following inequalities are true: $1 = s_{ii} \geq \sum_{j\neq i} s_{ij}$ and $1 = s_{ii}^2 \geq \sum_{j\neq i} s_{ij}^2$. Since $s_{ij} \leq 1$, then $s_{ij}^2 \leq s_{ij}$ and consequently, $\sum_{j\neq i} s_{ij} \geq \sum_{j\neq i} s_{ij}^2$. So, $2 = 2\,s_{ii} \geq 2\sum_{j\neq i} s_{ij} \geq \sum_{j\neq i} s_{ij}^2$, as well. This leads to $1 = 2\,s_{ii} - s_{ii}^2 \geq 2\sum_{j\neq i} s_{ij} - \sum_{j\neq i} s_{ij}^2$, which states that $(2S - S^{*2})$ is psd, which finishes the proof.

(2) Let $\mathbf{s} = \mathrm{diag}\,(S)$. Then, $D^{*2} = \mathbf{s}\mathbf{1}^T + \mathbf{1}\mathbf{s}^T - 2S$. Let $\mathbf{z}^T\mathbf{1} = 0$. Consequently, $\mathbf{z}^T D^{*2}\mathbf{z} = \mathbf{z}^T\mathbf{s}\mathbf{1}^T\mathbf{z} + \mathbf{z}^T\mathbf{1}\mathbf{s}^T\mathbf{z} - 2\mathbf{z}^T S\mathbf{z} = 0 + 0 - 2\mathbf{z}^T S\mathbf{z} < 0$, since S is psd. Hence, D^{*2} is cnd. As a result, D is Euclidean. ■

Corollary 3.39 [171] If the matrix $D = (\mathbf{1}\mathbf{1}^T - S)^{*\frac{1}{2}}$ is either non-metric or non-Euclidean metric, then S is not psd. Moreover, if $D = (\mathbf{1}\mathbf{1}^T - S)$ is either non-metric or non-Euclidean metric, then $2S - S^{*2}$ is not psd.

Theorem 3.40 (Correcting D to make it Euclidean) Let D be a non-Euclidean symmetric dissimilarity matrix and let $S(D) = -\frac{1}{2}\,J\,D\,J$, where $J = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$. Denote $\lambda_{\min}$ as the smallest eigenvalue of $S(D^{*2})$ and $\lambda_{\max}$ as the largest eigenvalue of the matrix $\begin{bmatrix} O_{n\times n} & 2\,S(D^{*2}) \\ -I_{n\times n} & -4\,S(D) \end{bmatrix}$, where $O_{n\times n}$ is the zero matrix and $I_{n\times n}$ is the identity matrix. Then, D can be corrected such that the matrices $D_\tau^{(1)}$ and $D_\kappa^{(2)}$ are Euclidean [171][4]:

(1) $D_\tau^{(1)} = [D^{*2} + 2\,\tau\,(\mathbf{1}\mathbf{1}^T - I)]^{*1/2}, \quad \tau \geq -\lambda_{\min}$,

(2) $D_\kappa^{(2)} = D + \kappa\,(\mathbf{1}\mathbf{1}^T - I), \quad \kappa \geq \lambda_{\max}$.

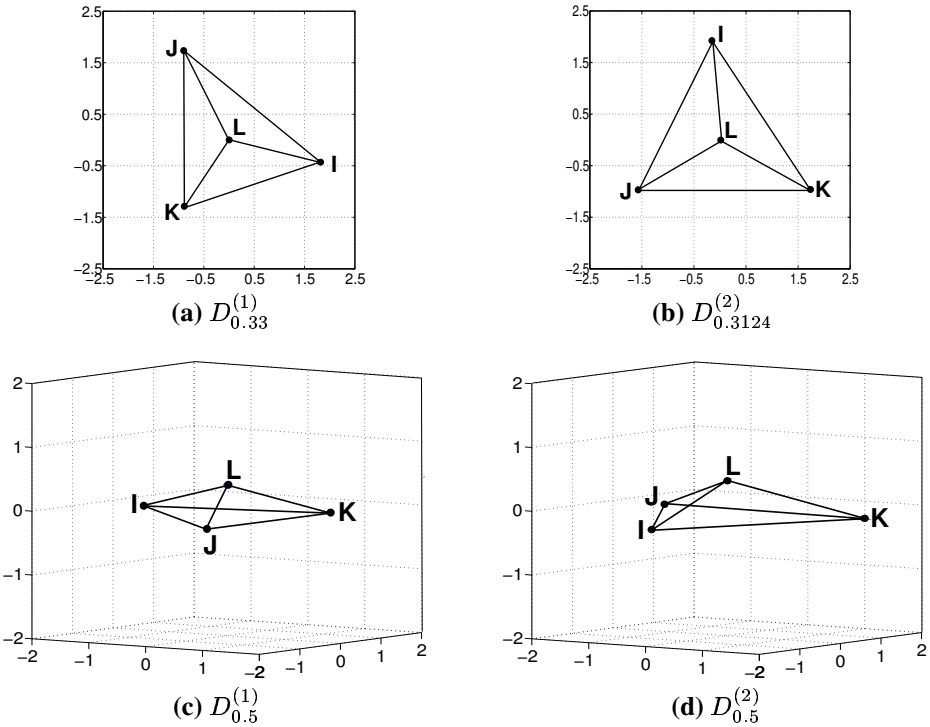
Proof.

(1) Assume that D is a non-Euclidean symmetric dissimilarity matrix. We will use Observation 3.33 to prove that $D_\tau^{(1)}$ is Euclidean. Consider $S_c = -\frac{1}{2}\,(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)\,D^{*2}(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$. Let $\mathbf{h} = \frac{1}{n}D^{*2}\mathbf{1} - \frac{1}{2\,n^2}\,\mathbf{1}\,\mathbf{1}^T D^{*2}\mathbf{1}$. Then, after straightforward mathematical operations, we can express S_c as follows $S_c = -\frac{1}{2}\,(D^{*2} - \mathbf{h}\mathbf{1}^T - \mathbf{1}\mathbf{h}^T)$. Note that $\mathrm{diag}\,(S_c) = -\frac{1}{2}\,[\mathrm{diag}\,(D^{*2}) - \mathrm{diag}\,(\mathbf{h}\mathbf{1}^T) - \mathrm{diag}\,(\mathbf{1}\mathbf{h}^T)] = -\frac{1}{2}\,[\mathbf{0} - \mathbf{h} - \mathbf{h}] = \mathbf{h}$. Therefore, D^{*2} can be expressed as $D^{*2} = \mathbf{h}\mathbf{1}^T + \mathbf{1}\mathbf{h}^T - 2\,S_c = \mathrm{diag}\,(S_c)\,\mathbf{1}^T + \mathbf{1}\,\mathrm{diag}\,(S_c)^T - 2\,S_c$. Let $S_c = (s_{ij})$, then $d_{ij}^2 = s_{ii} + s_{jj} - 2\,s_{ij}$ for all $i, j = 1, \ldots, n$. From the latter equation follows that adding a constant τ to the diagonal of S_c is equivalent to adding 2τ to the off-diagonal elements of D^{*2}.

An eigendecomposition of S_c is given as $S_c = Q\,\Lambda\,Q^T$, where $\Lambda = \mathrm{diag}\,(\lambda_i)$ consists of the eigenvalues in a non-increasing order and Q is an orthogonal matrix of the corresponding eigenvectors. If S_c is psd, then all eigenvalues are nonnegative, hence $\lambda_{\min} \geq 0$. However, since D is non-Euclidean, then S_c is not psd by Theorem 3.32. This means that there exist some negative eigenvalues, thereby $\lambda_{\min} < 0$.

Let $\tau \geq -\lambda_{\min}$, where $\tau > 0$ if S_c is not psd (note that $\tau \geq 0$ if S_c would be psd). Then, $\Lambda + \tau I$ is a matrix

[4] There are mistakes (misprints?) in the formulation of this theorem in [171].



(a) $D^{(1)}_{0.33}$ **(b)** $D^{(2)}_{0.3124}$

(c) $D^{(1)}_{0.5}$ **(d)** $D^{(2)}_{0.5}$

Fig. 3.3: Any non-Euclidean distances can be corrected to become Euclidean. Let D be the dissimilarity matrix from Fig. 3.2. Then, the matrix $D^{(1)}_{\tau} = [D^{*2} + 2\,\tau\,(\mathbf{1}\mathbf{1}^T - I)]^{*1/2}$ is Euclidean for $\tau \geq 0.33$ and the matrix $D^{(2)}_{\kappa} = D + \kappa\,(\mathbf{1}\mathbf{1}^T - I)$ is Euclidean for $\kappa \geq 0.3124$; see Theorem 3.40. The plots present Euclidean embeddings of the corrected distances. Note that by using $\tau = 0.33$ and $\kappa = 0.3124$, 2-dimensional representations are obtained; see plots (a) and (b), while for $\tau = \kappa = 0.5$ the number of dimension increases; see plots (c) and (d).

whose diagonal has nonnegative values. As a result, $S_\tau := Q\,[\Lambda + \tau\,I]\,Q^T$ is psd. Note further that $S_\tau = Q\,\Lambda\,Q^T + Q\,\tau\,I\,Q^T = S + \tau\,Q\,Q^T = S_c + \tau I$ and by the observation above, $S_\tau := S_c + \tau I = D^{*2} + 2\,\tau\,(\mathbf{1}\mathbf{1}^T - I)$. Since S_c is psd, then by Observation 3.33, $[D^{*2} + 2\,\tau\,(\mathbf{1}\mathbf{1}^T - I)]^{*1/2}$ is Euclidean.

(2) Here, the smallest $\kappa > 0$ is sought such that $D^{(2)}_{\kappa} = D + \kappa\,(\mathbf{1}\mathbf{1}^T - I)$ is Euclidean. Based on Observation 3.33, this means that κ should be chosen such that the smallest eigenvalue of $S_\kappa = -\frac{1}{2} J (D^{(2)}_{\kappa})^{*2} J$, where $J = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$, is zero. Let $\mathbf{q}$ be the eigenvector of S_κ corresponding to the zero eigenvalue. Then, one has: $S_\kappa\,\mathbf{q} = 0$. Since $(D^{(2)}_{\kappa})^{*2} = D^{*2} + 2\kappa D + \kappa^2(\mathbf{1}\mathbf{1}^T - I)$ and $JJ = J$ (which is easy to check), then after simple transformations, we obtain $-\frac{1}{2}\,(JD^{*2}J + 2\kappa JDJ - \kappa^2 J)\,\mathbf{q} = 0$. Let $\mathbf{p}$ be a vector such that $\mathbf{p} = -\frac{1}{\kappa} D^{*2} J\mathbf{q}$ or, equivalently, $-\kappa J\mathbf{p} = JD^{*2}J\mathbf{q}$. Then, one gets: $-\kappa J\mathbf{p} + 2\kappa JDJ\mathbf{q} - \kappa^2 J\mathbf{q} = 0$. After the division by κ (remember that $\kappa > 0$), the following equation is obtained: $-J\mathbf{p} + 2JDJ\mathbf{q} = \kappa J\mathbf{q}$. Consequently, by the fact that $JJ = J$, one needs to solve the following system of equations $\begin{cases} -JD^{*2}J\,(J\mathbf{q}) = \kappa\,(J\mathbf{p}) \\ -J\mathbf{p} + 2JDJ\,(J\mathbf{q}) = \kappa\,(J\mathbf{q}), \end{cases}$ which is equivalent to $\begin{bmatrix} O_{n\times n} & -JD^{*2}J \\ -I_{n\times n} & 2JDJ \end{bmatrix} \begin{bmatrix} J\,\mathbf{p} \\ J\,\mathbf{q} \end{bmatrix} = \kappa \begin{bmatrix} J\,\mathbf{p} \\ J\,\mathbf{q} \end{bmatrix}$. This means that κ is the largest eigenvalue of the matrix $\begin{bmatrix} O_{n\times n} & 2\,S(D^{*2}) \\ -I_{n\times n} & -4\,S(D) \end{bmatrix}$, which finishes the proof. ■

Note that both corrections defined above yield different solutions, as illustrated in Fig. 3.3. In the first case, the correction of dissimilarities is linearly related to the corresponding matrix of inner products S_c (see also Theorem 3.35), such that $\tilde{S}_c = S_c + \tau I$. This does not hold in the latter case.

Theorem 3.41 Let (X, d) be a finite metric space with the associated dissimilarity matrix D. Consider the assertions [8, 120, 195, 213]:

(1) (X, d) is ultrametric.
(2) (X, d) possesses the four-point property.
(3) (X, d) is l_2-embeddable.
(4) (X, d) is l_1-embeddable.
(5) (X, d) is hypermetric.
(6) (X, d) is of negative type.
(7) $(X, d^{1/2})$ is l_2-embeddable.

The following implications: (1) $\Rightarrow \begin{matrix}(2)\\(3)\end{matrix} \Rightarrow$ (4) $\Rightarrow$ (5) $\Rightarrow$ (6) $\Rightarrow$ (7) hold.

Proof.
(1) $\Rightarrow$ (2) Ultrametric space is realized by an ultrametric tree, which is additive by Def. 3.11. Hence, the four point property is fulfilled.

(1) $\Rightarrow$ (3) See [243] for a proof.

(2) $\Rightarrow$ (4) See Theorem 3.13 and also [8].

(3) $\Rightarrow$ (4) See [44, 75] for proofs.

(4) $\Rightarrow$ (5) Let $X = \{x_1, x_2, \ldots, x_n\}$. Based on Theorem 3.9, it suffices to show that every cut metric, Def. 3.8, satisfies the hypermetric inequalities, Def. 3.10. Let V and $V^c := X \backslash V$ define the cut. Then, the cut metric $\delta_V(x_i, x_j)$ equals 1 if $|V \cap \{x_i, x_j\}| = 1$ and 0, otherwise. Let $\mathbf{y} \in \mathbb{Z}^n$ such that $\sum_{i=1}^n y_i = 1$. Then $\sum_{i=1}^n \sum_{j=1}^n y_i y_j \delta_V(x_i, x_j) = 2 \sum_{i \in V} \sum_{j \notin V} y_i y_j = 2 \left(\sum_{i \in V} y_i\right)\left(\sum_{j \notin V} y_j\right) = 2 \left(\sum_{i \in V} y_i\right)\left(1 - \sum_{i \in V} y_i\right) \leq 0$. The latter inequality holds, since $\sum_{i \in V} y_i$ is an integer and, therefore, either both $\sum_{i \in V} y_i$ and $1 - \sum_{i \in V} y_i$ have opposite signs or one of them is zero. Hence, the cut metric is hypermetric.

(5) $\Rightarrow$ (6) Let D be hypermetric. Then, by Def. 3.10, $\mathbf{y}^T D\, \mathbf{y} \leq 0$ for $\mathbf{y} \in Z^n$ such that $\mathbf{y}^T \mathbf{1} = 1$. Let $\mathbf{x} \in \mathbb{R}^n$ be any vector. Then, $\mathbf{z} = (I - \mathbf{1}\mathbf{y}^T)\, \mathbf{x} \in \mathbb{R}^n$. Moreover, it is easy to check that $\mathbf{z}^T \mathbf{1} = 0$. Hence, by Def. 3.10, D is of negative type.

(6) $\Rightarrow$ (7) There is an equivalence of d being of negative type and D^{*2} being cnd. This implication is true based on Theorem 3.31. ■

Many dissimilarity measures are constructed by combining the measure applied to all the attribute separately. Given m attributes, the dissimilarity can be expressed in the form of $d(x, y) = \sum_{r=1}^m f(x_r, y_r)$, where $f(x_r, x_r) = 0$ and $f(x_r, y_r) = f(y_r, x_r) \geq 0$ for all r. Then, we have:

Corollary 3.42 Let $x, y \in \mathbb{R}^m$. Then $d(x, y) = \sum_{r=1}^m f(x_r, y_r)$ is metric, iff f is metric in $\mathbb{R}$.

Proof. $\Rightarrow$ Since f is nonnegative, symmetric and $f(u, u) = 0$ for $u \in \mathbb{R}$, then the axioms of reflexivity, symmetry and definiteness are fulfilled. Since d is metric, then $d(x, y) + d(y, z) \geq d(x, z)$ for all x, y, z. Consider x, y, z such that $x_r = c_x$, $y_r = c_y$, $z_r = c_z$ for all r and some constants c_x, c_y and c_z. The triangle inequality for d reduces to $f(x_c, y_c) + f(y_c, z_c) \geq f(x_c, z_c)$, hence f is metric.
$\Leftarrow$ Trivial. ■

Theorem 3.43 $d(x, y) = \sum_{r=1}^m f(x_r, y_r)$ is metric, iff $\rho(x, y) = \left(\sum_{r=1}^m [f(x_r, y_r)]^2\right)^{1/2}$ is metric.

Proof. See [171] for a proof.

Remark* 3.44 Direct product spaces allow us for a construction of a new space by combining two (or more) spaces; see also section 3.1.4. Given a number of square dissimilarity matrices, a new dissimilarity matrix can be created by applying some element-wise operator, such as sum or max, to them. For instance, $D = D_1 + D_2$. In the light of section 3.1.4, this means that finite generalized metric spaces are combined into a new one. The spaces are assumed to be defined on the same finite set X, yet they are distinguished by the dissimilarities measures used. Now, the consequences from Theorems 3.23 and 3.24 and the mathematical induction are:

- The max and sum operators preserve the metric properties.
- A square dissimilarity matrix, resulting from the summation of dissimilarity matrices preserves the l_1-embeddability, hypermetric and negative type properties.
- If ${D^{*2}}_1$ and ${D^{*2}}_2$ are of negative type, then $({D^{*2}}_1 + {D^{*2}}_2)^{*1/2}$ is l_2-embeddable. This follows from the preservation of the negative type property by summation and Theorem 3.41(7).

3.2.1 Relations between square distances and inner products

Assume n vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ are given in a Euclidean space. Based on the definitions of a Euclidean distance and an inner product, one has $d^2(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i - \mathbf{x}_j, \mathbf{x}_i - \mathbf{x}_j \rangle$. Therefore,

$$d^2(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_i \rangle + \langle \mathbf{x}_j, \mathbf{x}_j \rangle - 2\langle \mathbf{x}_i, \mathbf{x}_j \rangle = d^2(\mathbf{x}_i, \mathbf{0}) + d^2(\mathbf{x}_j, \mathbf{0}) - 2\,\langle \mathbf{x}_i, \mathbf{x}_j \rangle, \tag{3.2}$$

where $\mathbf{0}$ is the origin in this space. Consequently, the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ can be expressed as:

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = -\frac{1}{2}\left[d^2(\mathbf{x}_i, \mathbf{x}_j) - d^2(\mathbf{x}_i, \mathbf{0}) - d^2(\mathbf{x}_j, \mathbf{0})\right]. \tag{3.3}$$

Based on well known properties of inner products and formula (3.3), the square distance of $\mathbf{x}_i$ to the mean of the configuration, $d^2(\mathbf{x}_i, \overline{\mathbf{x}})$, can be expressed by distances as follows (see also [152, 398]):

$$\begin{aligned} d^2(\mathbf{x}_i, \overline{\mathbf{x}}) &= ||\mathbf{x}_i - \overline{\mathbf{x}}||^2 = \langle \mathbf{x}_i - \overline{\mathbf{x}},\, \mathbf{x}_i - \overline{\mathbf{x}} \rangle = \langle \mathbf{x}_i,\, \mathbf{x}_i \rangle + \langle \overline{\mathbf{x}},\, \overline{\mathbf{x}} \rangle - 2\,\langle \mathbf{x}_i,\, \overline{\mathbf{x}} \rangle \\ &= d^2(\mathbf{x}_i,\, \mathbf{0}) + \frac{1}{n^2}\sum_{p=1}^{n}\sum_{s=1}^{n}\langle \mathbf{x}_p,\, \mathbf{x}_s \rangle - \frac{2}{n}\sum_{s=1}^{n}\langle \mathbf{x}_i,\, \mathbf{x}_s \rangle \\ &= d^2(\mathbf{x}_i,\, \mathbf{0}) + \frac{1}{2n^2}\sum_{p,s=1}^{n}\left[d^2(\mathbf{x}_p, \mathbf{0}) + d^2(\mathbf{x}_s, \mathbf{0}) - d^2(\mathbf{x}_p,\, \mathbf{x}_s)\right] \\ &- \frac{1}{n}\sum_{s=1}^{n}\left[d^2(\mathbf{x}_i,\, \mathbf{0}) + d^2(\mathbf{x}_s,\, \mathbf{0}) - d^2(\mathbf{x}_i,\, \mathbf{x}_s)\right] \\ &= \frac{1}{n}\sum_{s=1}^{n} d^2(\mathbf{x}_i,\, \mathbf{x}_s) - \frac{1}{2n^2}\sum_{p,s=1}^{n} d^2(\mathbf{x}_p,\, \mathbf{x}_s) = d^2_{i\cdot} - \frac{1}{2}d^2_{\cdot\cdot}, \end{aligned} \tag{3.4}$$

where, abusing somewhat the notation, $d^2_{i\cdot}$ stands for the mean computed over the i-th row of the matrix D^{*2} and $d^2_{\cdot\cdot}$ is the overall mean.

Without loss of generality, let us assume that the mean vector coincides with the origin, i.e. $\overline{\mathbf{x}} = \mathbf{0}$. This implies that $d^2(\mathbf{x}_i, \mathbf{0}) = d^2(\mathbf{x}_i, \overline{\mathbf{x}})$. By applying this into formula (3.3) and by plugging (3.4), one gets the following expression for all $i, j = 1, 2, \ldots, n$:

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = -\frac{1}{2}\left[d^2(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{n}\sum_{s=1}^{n} d^2(\mathbf{x}_i, \mathbf{x}_s) - \frac{1}{n}\sum_{s=1}^{n} d^2(\mathbf{x}_s, \mathbf{x}_j) + \frac{1}{n^2}\sum_{p,s=1}^{n} d^2(\mathbf{x}_p, \mathbf{x}_s)\right]. \tag{3.5}$$

Let $X \in \mathbb{R}^{n\times k}$ be a representation of all vectors ($\mathbf{x}_i^T$ is the i-th row of X) and let G be the matrix of inner products, i.e. $G = X\,X^T$, such that $g_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Then, formula (3.5) simplifies to

$$g_{ij} = -\frac{1}{2}\,(d^2_{ij} - d^2_{i\cdot} - d^2_{\cdot j} + d^2_{\cdot\cdot}). \tag{3.6}$$

Let D^{*2} be an $n\times n$ square Euclidean distance matrix. By using the following substitutions $d^2_{i\cdot} = \frac{1}{n}\,D^{*2}(\mathbf{x}_i, \cdot)\mathbf{1}^T$, $d^2_{\cdot j} = \frac{1}{n}\,D^{*2}(\cdot, \mathbf{x}_j)^T\mathbf{1}^T$ and $d^2_{\cdot\cdot} = \frac{1}{n^2}\,\mathbf{1}\,D^{*2}\mathbf{1}^T$ and after some straightforward mathematical operations, G is determined as

$$G = -\frac{1}{2}J\,D^{*2}\,J, \quad \text{where} \quad J = I - \frac{1}{n}\,\mathbf{1}\mathbf{1}^T. \tag{3.7}$$

Alternatively, by formula (3.2), D^{*2} can be defined by the Gram matrix G as

$$D^{*2} = \mathbf{g}\,\mathbf{1}^T + \mathbf{1}\,\mathbf{g}^T - 2\,G, \tag{3.8}$$

where $\mathbf{g}$ is a vector of the diagonal elements of G, i.e. $\mathbf{g} = \mathrm{diag}\,(G)$. In this way, an explicit linear relation between the Gram matrix G and the matrix of square Euclidean distances D^{*2} is expressed. Although the vectors X are assumed to have a zero mean, this is not essential, since the configuration can be shifted such that the origin coincides with any other vector lying in a convex hull of X. This means that instead of $\overline{\mathbf{x}} := X^T \frac{1}{n}\mathbf{1} = \mathbf{0}$, we require that $X^T\mathbf{s} = \mathbf{0}$ with $\mathbf{s}^T\mathbf{1} = 1$. As a result, J from formula (3.7) becomes $J = I - \mathbf{1}\,\mathbf{s}^T$, so in a bottom-up way, we reached Theorems 3.32 and 3.35, and Observation 3.34.

Note that precisely the same reasoning holds for a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$, since the linear formulation between square distances and inner products in both spaces is the same. Therefore, formula (3.5) is valid for a pseudo-Euclidean space, where instead of $\langle\cdot,\cdot\rangle$, an indefinite inner product $\langle\cdot,\cdot\rangle_{\mathcal{E}}$ defined by (2.1) is meant and instead of the Euclidean distance, a pseudo-Euclidean distance (2.2) is used. This leads to the conclusion that formulas (3.7) and (3.8) remain true for a pseudo-Euclidean space, as well. More discussion follows in section 3.3.3.

3.3 Linear embeddings of dissimilarities

Dissimilarity data can be embedded into a Euclidean space in a number of ways. Since we are interested in a faithful configuration, an embedding is found such that the distances are preserved as well as possible. Here, linear embeddings are considered, first isometric ones and then their approximate variants. Since it is not always possible to isometrically embed the data into a Euclidean space, a pseudo-Euclidean space will be considered. From such a perspective, any finite premetric space can be isometrically embedded into a pseudo-Euclidean space.

3.3.1 Euclidean embedding

Given a set $R = \{p_1, p_2, \ldots, p_n\}$ of n objects[5] and a Euclidean distance matrix $D := D(R,R) \in \mathbb{R}^{n\times n}$ between them, a distance preserving linear mapping into a Euclidean space can be found. Such a projection is known as *classical scaling* (CS) [37, 72, 425]. This means that the dimensionality k, $k \leq n$ and a configuration $X \in \mathbb{R}^{n\times k}$ have to be determined such that the (squared) Euclidean distances are preserved. Note that having found one configuration, another one can be created by a rotation or a translation. Without loss of generality, the mapping is constructed such that the origin coincides with the mean of the configuration X.

To define X, the relations between the Euclidean distances and inner products are used. We know from section 3.2.1 that $D^{*2} = \mathbf{g}\mathbf{1}^T + \mathbf{1}\mathbf{g}^T - 2\,G$, where G is the Gram matrix of the underlying configuration X, $G = XX^T$, and $\mathbf{g} = \mathrm{diag}\,(G)$. G can also be expressed as $G = -\frac{1}{2}\,JD^{*2}J$, where J is the centering matrix $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{n\times n}$. J projects[6] the data such that the final configuration has a zero mean. Then, the factorization of G by its eigendecomposition can be found as

$$G = Q\,\Lambda\,Q^T, \tag{3.9}$$

[5] The set R is a collection of objects. The objects may not be yet represented for the use of computer algorithms, therefore, you may think of R as of an index of objects. X, on the contrary, is a representation of the objects from R in a Euclidean (pseudo-Euclidean) space $\mathbb{R}^k$. Hence, it can be described by the vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$.

[6] A more general projection can be achieved imposing that a weighted mean of X becomes zero; see also section 3.2.1. Then, $J = I - \mathbf{1}\,\mathbf{s}^T$, where $\mathbf{s}$ is such that $\mathbf{s}^T\mathbf{1} = 1$ and $G = -\frac{1}{2}\,J\,D^{*2}J^T$. By choosing a proper $\mathbf{s}$, any arbitrary point of X can be projected at the origin, as well.

where Λ is a diagonal matrix whose diagonal consists of nonnegative eigenvalues (G is positive definite by Theorem 3.32) ranked in descending order and followed by the zero values, and Q is an orthogonal matrix of the corresponding eigenvectors; see Theorem 3.35. As a matrix of inner products, G can be expressed as $G = XX^T$. Hence, for k ($k \leq n$) non-zero eigenvalues, a k-dimensional representation X can be then determined as

$$X = Q_k \, \Lambda_k^{1/2}, \quad \text{where} \quad Q_k \in \mathbb{R}^{n \times k}, \; \Lambda_k^{1/2} \in \mathbb{R}^{k \times k}, \tag{3.10}$$

where Q_k is the matrix of the k leading eigenvectors (i.e. corresponding to the largest eigenvalues) and $\Lambda_k^{1/2}$ contains the square roots of the corresponding eigenvalues. This is the result of classical scaling. Note that X determined in this way is unique up to rotation (the centroid is now fixed), since for any orthogonal matrix P, $XX^T = (XP)(XP)^T$. Also the features of X are uncorrelated, since the columns of Q_k are orthonormal. The estimated covariance matrix of X becomes then

$$C = \frac{1}{n-1} X^T X = \frac{1}{n-1} \Lambda_k^{1/2} Q_k^T Q_k \Lambda_k^{1/2} = \frac{1}{n-1} \Lambda_k. \tag{3.11}$$

This means that the vector configuration X obtained in this way is equivalent to a Principal Component Analysis (PCA) result [97, 138][7]. Moreover, the eigenvalues of G play a key role here as they linearly scale the features and by this they decide which of them are significant and which not. Note that such an uncorrelated vector representation is obtained for $J = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$, which corresponds to $\mathbf{s} = \frac{1}{n}\mathbf{1}$. Only then the vector mean of X is set to the origin. This justifies why this particular $\mathbf{s}$ is in favor. Note also that G can be seen as a reproducing kernel for the Euclidean space $\mathbb{R}^k$; see also section 2.3.1. This also means that the embedding procedure can be performed directly starting from a positive definite matrix G, hence a kernel, which can be treated as a similarity matrix.

3.3.2 Correction of non-Euclidean dissimilarities

The matrix of inner products, $G = -\frac{1}{2} J D^{*2} J$, is positive (semi)definite if the dissimilarity matrix $D \in \mathbb{R}^{n \times n}$ is Euclidean; see Theorem 3.32 and Observation 3.33. Therefore, a finite quasimetric space described by a non-Euclidean D has a Gram matrix G which is not psd. Since G has negative eigenvalues, then a Euclidean representation X cannot be constructed by formula (3.10) as it relies on the square roots of the eigenvalues. However, D can be corrected such that the corresponding G becomes psd. Some possible approaches to address this issue are:

- Only p positive eigenvalues are taken into account, resulting in a p-dimensional Euclidean configuration $X = Q_p \, \Lambda_p^{1/2}$ ($p < k$). Since the actual dissimilarities D are nonnegative, the magnitude of the smallest negative eigenvalue of G is smaller than the largest positive one. Also the sum of the positive eigenvalues is larger than the sum of magnitudes of the negative ones. Hence, after neglecting the negative contributions, the resulting Euclidean distances are overestimated. This might be a justified approach if the negative eigenvalues are relatively small with respect to the positive ones; see section 3.3.6, where the issue of noise influence is discussed. We argue that the distances which are directly measured may be noisy and, therefore, not perfectly Euclidean. This will result in small negative eigenvalues of G. Therefore, by disregarding them, noise is diminished.

[7] Assume a configuration X in a Euclidean space $\mathbb{R}^k$ and the Euclidean distance matrix $D(X, X)$. Let the mean of X lie at the origin. Then, the PCA projection based on the estimated covariance matrix of X, $cov(X) = \frac{1}{n-1} X^T X$, gives the classical scaling result (3.10). To observe it, let $(\lambda_i, \mathbf{q}_i)$ be an eigen-pair of $G = XX^T$ (computed of course as $-\frac{1}{2} JD^{*2}J$). Then, $XX^T\mathbf{q}_i = \lambda_i \mathbf{q}_i$ and further by the multiplication by $\frac{1}{n-1}X^T$, one obtains $\frac{1}{n-1}(X^TX)(X^T\mathbf{q}_i) = \frac{\lambda_i}{n-1}(X^T\mathbf{q}_i)$. It is straightforward to check that the vectors $\mathbf{q}_i^{PCA} := X^T\mathbf{q}_i/\sqrt{\lambda_i}$, i=1,2,...,n, are orthonormal. This means that $(\frac{\lambda_i}{n-1}, \mathbf{q}_i^{PCA})$ is an eigen-pair of $cov(X)$. The solution of the PCA projection, $\mathbf{x}_i^{PCA}$ is given as $\mathbf{x}_i^{PCA} = X\mathbf{q}_i^{PCA} = XX^T\mathbf{q}_i/\sqrt{\lambda_i}$, which is equivalent to $\mathbf{x}_i^{PCA} = \sqrt{\lambda_i}\mathbf{q}_i$. In the matrix notation, $X^{PCA} = Q_k\sqrt{\Lambda_k}$, which is the classical scaling result.

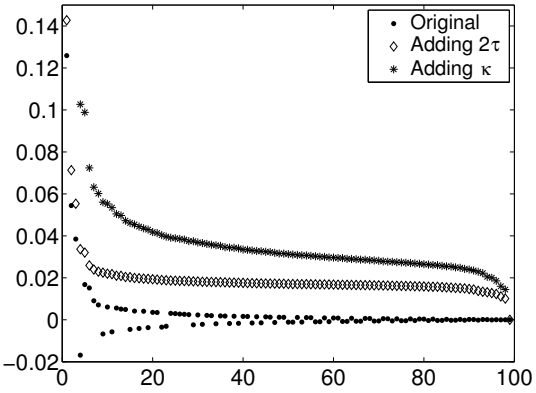


Fig. 3.4: Eigenvalues resulting from the embedding of a 100×100 modified Hausdorff dissimilarity representation D of the NIST-38 digit data (see section A) and the corrected representations $D_{2\tau}$ and D_κ. The eigenvalues are sorted with respect to their magnitudes. Remember that $\tau := |\lambda_{min}|$, where λ_{min} is the smallest eigenvalue. Adding 2τ to the off-diagonal elements of D^{*2} is equivalent to adding τ to all the original eigenvalues (hence the smallest one becomes now zero). Eigenvectors remain the same. The relation between original eigenvalues and eigenvalues of D_κ is nonlinear.

- There exists a positive constant $\tau \geq -\lambda_{\min}$, where $\lambda_{\min}$ is the smallest (negative) eigenvalue of G, such that $D_{2\tau} = [D^{*2} + 2\,\tau\,(\mathbf{1}\mathbf{1}^T - I)]^{*1/2}$ is Euclidean; see Theorem 3.40. This means that the corresponding G_τ is pd. In practice, the eigenvectors of G and G_τ are identical, but the value τ is added to the non-zero eigenvalues, giving rise to a new diagonal eigenvalue matrix $\Lambda_\tau := \Lambda_k + \tau\, I$. This is equivalent to 'regularizing' the covariance matrix of our configuration X by $C = \frac{1}{n-1}\,(\Lambda_k + \tau\, I)$ and changing X respectively. Note that the original dissimilarities are distorted significantly for a large τ.
- There exists a positive constant $\kappa \geq \lambda_{\max}$, where $\lambda_{\max}$ is defined in Theorem 3.40, such that $D_\kappa = D + \kappa\,(\mathbf{1}\mathbf{1}^T - I)$ is Euclidean. After the correction of D, the corresponding Gram matrix G_κ yields eigenvalues and eigenvectors which are different than those of G.
- There exists a parameter p such that the matrix $D_p = (g(d_{ij}; p))$ is Euclidean for g defined as in Def. 3.20; see also Theorem 3.22. In practice, p can be determined only by trial-and-error, although $p < 1$, in general. In principle, an indication how p should be chosen is given by $1-r$, where r is the ratio of the absolute value of the smallest negative eigenvalue to the largest positive one. An algorithm to determine p has also been proposed in [70].

These approaches transform the original dissimilarity data such that a Euclidean configuration can be found. This is especially useful when the negative eigenvalues are relatively small in magnitude, which suggests that the original distance measure is close to Euclidean. In such cases, the negative eigenvalues can be interpreted as noise contributions. If the negative eigenvalues are relatively large (in magnitude), then by neglecting them, important information might have been neglected; see also Fig. 3.4. There is still an open question referring to the consequences on the learning tasks of transforming the considered problem into a Euclidean one, either by neglecting the negative eigenvalues or by directly enlarging D^{*2} by a constant.

In general, a hollow dissimilarity matrix D, Def. 2.38, can be corrected to have the Euclidean behavior. First, to make it definite, any zero dissimilarity between two different objects should change into a small fixed value, depending on the overall distances, e.g. 0.01. Next, to make it symmetric, an operation like averaging of d_{ij} and d_{ji} or taking their maximum value should be performed. Since D has become quasimetric, any of the corrections described above will make it Euclidean.

It is also possible that the corrections applied are less than required for guaranteeing the Euclidean behavior (i.e. by adding a constant to the off-diagonal elements of D^{*2} smaller than the required one). In such cases, the measure is simply made 'more' Euclidean (hence, also 'more' metric),

since the influence of negative eigenvalues will become smaller after the discussed transformations.

3.3.3 Pseudo-Euclidean embedding

For cases where a Euclidean space is not 'large enough' to embed the dissimilarity data (this is due negative eigenvalues of the corresponding Gram matrix), Goldfarb [151, 152] proposed to embed D into a pseudo-Euclidean space. Such a procedure can be applied to any premetric finite representation. A pseudo-Euclidean space is a direct orthogonal decomposition of two Euclidean spaces, for which the inner product operation is positive definite on the first space and negative definite on the second one; see section 2.4 for details. To determine the embedding, the same reasoning as in the Euclidean case is applied here. The essential difference refers to the notion of an inner product and a distance. Now $G = -\frac{1}{2} JD^{*2}J$ is the Gram matrix, but expressed as:

$$G = X\, \mathcal{J}_{pq}\, X^T, \tag{3.12}$$

where $\mathcal{J}_{pq}$ is the fundamental symmetry matrix in a pseudo-Euclidean space. Following [152], we can write (compare also formula 3.9):

$$X\, \mathcal{J}_{pq}\, X^T = G = Q\, \Lambda\, Q^T = Q\, |\Lambda|^{1/2} \begin{bmatrix} \mathcal{J}_{pq} & \\ & 0 \end{bmatrix} |\Lambda|^{1/2} Q^T, \quad \text{where} \quad \mathcal{J}_{pq} = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix} \tag{3.13}$$

and $p+q=k$. Λ is now based on p positive and q negative eigenvalues, presented in the following order: first the positive eigenvalues with decreasing values, then the negative ones with decreasing magnitudes followed by zeros. X can now be expressed in a pseudo-Euclidean space $\mathbb{R}^k = \mathbb{R}^{(p,q)}$ of the signature (p, q) [151] as follows:

$$X = Q_k\, |\Lambda_k|^{1/2}, \tag{3.14}$$

where only k non-zero eigenvalues in Λ_k are taken into account. (Otherwise, additional zero eigenvalues would describe X in a finite indefinite inner product space, but degenerate; see also section 2.4). The estimated pseudo-Euclidean covariance matrix is given as [152]:

$$C = \frac{1}{n-1} X^T X\, \mathcal{J}_{pq} = \frac{1}{n-1} |\Lambda_k| \mathcal{J}_{pq} = \frac{1}{k-1} \Lambda_k, \tag{3.15}$$

Hence X is an uncorrelated representation. Although C is not pd in a Euclidean sense, it is k-pd, hence pd in a pseudo-Euclidean sense; see Def. 2.63. This means that X is a result of a mapping in the spirit of the PCA projection and the whole embedding procedure can also be interpreted as a kernel-PCA [345, 351] approach, where the kernel G is a reproducing kernel for the pseudo-Euclidean feature space; see also section 2.4.1. Note that, similarly to the Euclidean space, such an uncorrelated vector representation is obtained for $J = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)$, i.e. $\mathbf{s} = \frac{1}{n}\mathbf{1}$ only.

Computing square distances in a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$ can be interpreted as computing the square Euclidean distance in a 'positive' space $\mathbb{R}^p$ and subtracting the square Euclidean distance found in a 'negative' space $\mathbb{R}^q$. The distances computed only in the 'positive' space are overestimated, therefore, the purpose of the 'negative' space is to correct them, i.e. make them be non-Euclidean. Since the pseudo-Euclidean spaces that we are going to consider will result from the embedding process of nonnegative dissimilarities, the contribution of the 'negative' space $\mathbb{R}^q$ to the overall distances is smaller than of the space $\mathbb{R}^p$. In such a case, due to the construction of X, in $\mathbb{R}^q$, X takes values (much) smaller than in space $\mathbb{R}^p$. Practice confirms that many summation-based measures are close to the Euclidean distance, giving rise to relatively small negative eigenvalues in the embedding. On the contrary, measures based on operations like minimum or maximum might be completely different.

Note that the proposed embedding is very *general*. Any symmetric dissimilarity matrix can be embedded in a pseudo-Euclidean space. In case of asymmetric matrices, they first need to be corrected to the symmetric ones.

3.3.4 Generalized average variance

In an embedded pseudo-Euclidean space $\mathcal{E}$, the generalized average variance of the configuration X with the vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ can be defined using the trace of the covariance matrix of X, i.e. the sum of variances, as follows:

$$V_d(X) = \frac{1}{n} \sum_{j=1}^{n} ||\mathbf{x}_j||_{\mathcal{E}}^2 - ||\overline{\mathbf{x}}||_{\mathcal{E}}^2 \tag{3.16}$$

Remember that $||.||_{\mathcal{E}} = \langle \cdot, \cdot \rangle_{\mathcal{E}}$. Since X reflects the same geometry as imposed by the square distance matrix $D^{*2}(R,R)$, based on the set $R = \{p_1, p_2, \ldots, p_n\}$, it is possible to express $V_d(X)$ only in terms of such square distances. Formally, one has:

Corollary 3.45 Given the dissimilarity matrix $D := D(R,R)$, the generalized average variance of the embedded pseudo-Euclidean configuration X is given as the average square dissimilarity:

$$V_d(T) = \frac{1}{2n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} d^2(p_j, p_k) \tag{3.17}$$

Proof. We will show now the equivalence of (3.17) and (3.16) by using equivalent transformations. Making use of formula (3.8) and the facts that $\mathbf{1}^T\mathbf{1} = n$, and $\mathbf{1}^T\mathbf{g} = \mathrm{tr}(G) = \sum_{j=1}^{n} ||\mathbf{x}_j||_{\mathcal{E}}^2$, one gets: $V_d(T) = \frac{1}{2n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} d^2(p_j, p_k) = \frac{1}{2n^2} \mathbf{1}^T D^{*2} \mathbf{1} = \frac{1}{2n^2} [\mathbf{1}^T\mathbf{g}\, \mathbf{1}^T\mathbf{1} + \mathbf{1}^T\mathbf{1}\, \mathbf{g}^T\mathbf{1} - 2\, \mathbf{1}^T G\, \mathbf{1}] = \frac{1}{2n} \mathbf{1}^T\mathbf{g} + \frac{1}{2n} \mathbf{g}^T\mathbf{1} - \frac{1}{n^2} \mathbf{1}^T G\, \mathbf{1} = \frac{1}{n} \mathbf{1}^T\mathbf{g} - \frac{1}{n^2} \mathbf{1}^T G\, \mathbf{1} = \frac{1}{n} \sum_{j=1}^{n} ||\mathbf{x}_j||_{\mathcal{E}}^2 - ||\overline{\mathbf{x}}||_{\mathcal{E}}^2$, since $\mathbf{1}^T G\, \mathbf{1} = \mathbf{1}^T X\, \mathcal{J}_{pq}\, X^T \mathbf{1} = \overline{\mathbf{x}}^T\, \mathcal{J}_{pq}\, \overline{\mathbf{x}} = ||\overline{\mathbf{x}}||_{\mathcal{E}}^2$, for $\mathcal{J}_{pq}$ being the matrix of inner products in the space $\mathcal{E}$. Since $\mathcal{J}_{pq} = I$ for the Euclidean space, then the above reasoning remains valid for a Euclidean space, i.e. if D is Euclidean. ■

3.3.5 Projecting new points to an embedded space

Let $X \in \mathbb{R}^{n \times k}$, $k = p+q$, be a configuration in a pseudo-Euclidean space $\mathbb{R}^k = \mathbb{R}^{(p,q)}$ that preserves all pairwise distances expressed by $D(R,R)$. (A Euclidean case is included for $q = 0$.) Given a matrix $D_n \in \mathbb{R}^{t \times n}$, expressing dissimilarities between t new objects and all objects of the set R, new vectors can be projected to an embedded space. Let X_n be the configuration of new objects to be determined. First, the cross-Gram matrix G_n relating all new objects to the objects from R should be found.

Corollary 3.46 Let $D(R,R)$ be isometrically embedded into $(X, \mathbb{R}^k)$, where $\mathbb{R}^k := \mathbb{R}^{(p,q)}$). Let D_n be a dissimilarity matrix between t novel objects and the objects of R. The cross-Gram matrix G_n of indefinite inner products is given as $G_n = -\frac{1}{2}[D_n^{*2}\, J - U\, D^{*2} J]$, where $J = (I - \frac{1}{n}\mathbf{1}\mathbf{1}^T) \in \mathbb{R}^{n \times n}$ and $U = \frac{1}{t}\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{t \times n}$.

Proof. Assume that $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_t\}$ is a vector representation of new objects projected into the space $\mathbb{R}^k$. It follows from formula (3.3) that the inner product between a new vector and the original points is given by $\langle \mathbf{y}_i, \mathbf{x}_j \rangle_{\mathcal{E}} = -\frac{1}{2}[d_n^2(\mathbf{y}_i, \mathbf{x}_j) - d_n^2(\mathbf{y}_i, \mathbf{0}) - d^2(\mathbf{x}_j, \mathbf{0})]$. Making use of formula (3.4) and the fact that the mean coincides with the origin, the indefinite inner product becomes then

$$\begin{aligned} \langle \mathbf{y}_i, \mathbf{x}_j \rangle_{\mathcal{E}} &= -\frac{1}{2}[d_n^2(\mathbf{y}_i, \mathbf{x}_j) - \frac{1}{n} \sum_{s=1}^{n} d_n^2(\mathbf{y}_i, \mathbf{x}_s) - \frac{1}{n} \sum_{s=1}^{n} d^2(\mathbf{x}_s, \mathbf{x}_j) + \frac{1}{n^2} \sum_{p,s=1}^{n} d^2(\mathbf{x}_p, \mathbf{x}_s)] \\ &= -\frac{1}{2}\,(d_n^2(\mathbf{y}_i, \mathbf{x}_j) - (d_n^2)_{i\,\cdot} - d_{\cdot\,j}^2 + d_{\cdot\,\cdot}^2), \end{aligned} \tag{3.18}$$

where, $(d_n^2)_{i\,\cdot}$ stands for the mean computed on the i-th row of the dissimilarity matrix D_n^{*2}, $d_{\cdot\,j}^2$ stands for the mean computed over the j-th row of the matrix D^{*2} and $d_{\cdot\,\cdot}^2$ is the overall mean.
Let $G_n \in \mathbb{R}^{t \times n}$ be the matrix of inner products between t new vectors and n original ones. Using elementary

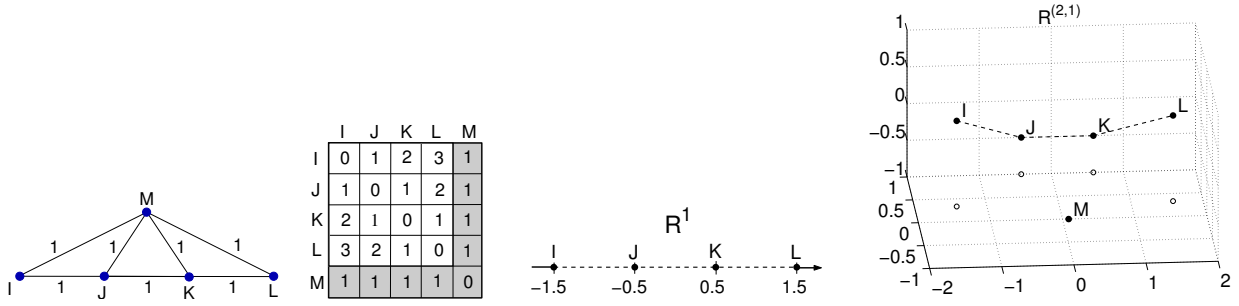


Fig. 3.5: By adding one point to the set R, the dimensionality k of the vector representation of $D(R,R)$ in a pseudo-Euclidean space might increase by more than one. The points of $R=\{I,J,K,L\}$ lie on a line, but the point M does not (the leftmost plot). The embedding of $D(R,R)$ reveals a 1D configuration (third plot). After enlarging R by M, a pseudo-Euclidean configuration in $\mathbb{R}^{(2,1)}$ is obtained (the rightmost plot, where the z-axis describes the 'negative' contribution), increasing the dimensionality by 2. The circles in the rightmost plot correspond to the points I – L projected into a plane, parallel to the xy-plane, on which M lies.

matrix operations, formula (3.18) can be rewritten as: $G_n = -\frac{1}{2}(D_n^{*2}\,J - U\,D^{*2}J)$, where J is the centering matrix and $U=\frac{1}{t}\,\mathbf{1}\mathbf{1}^T \in \mathbb{R}^{t\times n}$. ■

Consequently, the cross-Gram matrix G_n becomes now $G_n = -\frac{1}{2}(D_n^{*2}\,J - U\,D^{*2}J)$. On the other hand, G_n is the matrix of indefinite inner products and, thereby, it can be expressed as:

$$G_n = X_n\,\mathcal{J}_{pq}\,X^T \quad \text{with} \quad \mathcal{J}_{pq} = \begin{cases} I \in \mathbb{R}^{k\times k}, & \text{if } \mathbb{R}^k \text{ is Euclidean.} \\ \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix} \in \mathbb{R}^{k\times k}, & \text{if } \mathbb{R}^k \text{ is pseudo-Euclidean.} \end{cases} \tag{3.19}$$

Therefore, X_n can be found as the indefinite least-square solution to $X_n\,\mathcal{J}_{pq}\,X^T = G_n$, i.e. $X_n = G_n\,X\,(X^TX)^{-1}\,\mathcal{J}_{pq}$; see Theorem 2.95, Observation 2.96 and Corollary 2.97 for justification.

Knowing that $X^TX = |\Lambda|$ and $X = Q_k\,|\Lambda_k|^{1/2}$, X_n is alternatively expressed as

$$X_n = G_n\,X\,|\Lambda|^{-1}\,\mathcal{J}_{pq} \quad \text{or} \quad X_n = G_n\,Q_k\,|\Lambda_k|^{-\frac{1}{2}}\,\mathcal{J}_{pq}. \tag{3.20}$$

Hence, assuming that $J - U\,D^{*2}J$ is pre-computed, the computational complexity of determine the cross-Gram values of G_n for a single object is $\mathcal{O}(n)$. Since $X\,|\Lambda|^{-1}\,\mathcal{J}_{pq}$ can be pre-computed as well, then $\mathcal{O}(nk)$ operations are required for a projection of a new object.

3.3.6 Reduction of dimensionality

By enlarging the set R by one object, in practice, one point is added to a finite pseudo-Euclidean space, but the dimensionality k of the vector representation resulting from the enlarged D might increase by more than one, contrary to the Euclidean case [152]; see Fig. 3.5 for an illustration. This means that both outliers and noise can significantly contribute to the resulting dimensionality k. In practice, when new points are added, they are projected into the space determined by the starting configuration X. Therefore, the reliability of X, i.e. whether $D(R,R)$ is sufficiently well sampled, plays an essential role in the process of representing new data, and consequently, the performance of learning algorithms applied further on.

Originally, the pseudo-Euclidean configuration X is found such that the distances are preserved exactly and the dimensionality of X is determined by the number of non-zero eigenvalues of G. However, there might be many relatively small non-zero eigenvalues as compared to the large ones. Knowing that dissimilarities are noisy measurements, the small eigenvalues correspond to non-significant directions of X. In such a framework, neglecting small eigenvalues stands for reducing

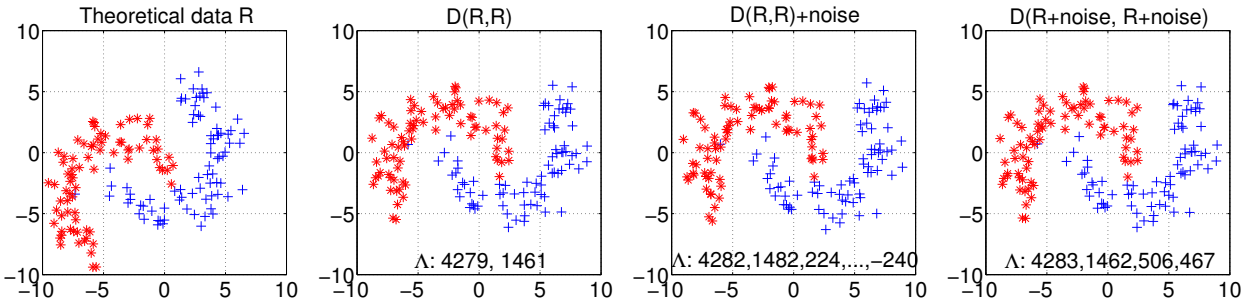


Fig. 3.6: Noise influence on the eigenvalues of G. The first, leftmost plot presents a theoretical banana data of 160 points, for which the Euclidean distance matrix D has been computed. The second plot shows the embedding of D into a 2D space (note that the retrieved configuration is exact up to a rotation). The third plot presents the projection into the first 2 dimensions of the 159D data obtained via embedding of the distorted distances $\tilde{D}$, (where $\tilde{d}_{ij} = d_{ij} + \varepsilon_{ij}$ for $i \neq j$ and $\varepsilon_{ij} \sim N(0,1)$) (taking care that $\min_{i\neq j} |\varepsilon_{ij}| < \min_{i\neq j} d_{ij}$, i.e. no negative distances arise), which become non-Euclidean. The average distortion is 0.8, while the average Euclidean distance is 7.57. The rightmost plot presents the projection into the first 2 dimensions of the 4D data obtained via embedding of $D(\tilde{R})$, where $\tilde{R}$ consists of the theoretical data R to which 2 noisy features were added giving rise to the average distortion of 0.9. Note that the first 2 largest eigenvalues, as presented in the plots, are relatively the same for the non-distorted as well as distorted data, which practically gives the same results in all the cases. Therefore, by neglecting relatively small eigenvalues, noise is diminished.

noise contribution (see Fig.3.6 for an illustration) or for determining a representation with the intrinsic dimension. In both cases, the distances will be preserved approximately. One has, however, a control over the dimensionality of the reduced vector representation. Basically, the dimensionality reduction can be achieved by the orthogonal projection, governed by the PCA. The particular construction of $X = Q_k |\Lambda_k|^{1/2}$ and the fact that X is an uncorrelated vector representation, i.e. the covariance matrix $C = \frac{1}{n-1}\Lambda_k$ is a diagonal matrix, stand for X being given in the form of the orthogonal PCA projection; see formula (3.15). It means that the reduction of dimensionality is performed in a simple way by neglecting directions corresponding to eigenvalues small in magnitude. The reduced representation is then determined by p' significant positive eigenvalues and q' significant (in magnitude) negative eigenvalues[8]. Therefore, $X_{red} \in \mathbb{R}^{n\times m}$, $m < k$, is found as $X_{red} = Q_m |\Lambda_m|^{1/2}$, where $m = p' + q'$ and Λ_m is a diagonal matrix of first, decreasing positive eigenvalues and then increasing negative eigenvalues, and Q_m is the matrix of the corresponding eigenvectors.

3.3.7 Reduction of complexity

Reduction of dimensionality described above is useful for data representation, since both noise and non-significant information are neglected. Still, the reduced configuration in $\mathbb{R}^m = \mathbb{R}^{(p',q')}$ is determined by all n objects. Yet, for the definition of an m-dimensional pseudo-Euclidean space, only $m+1$ objects are in principle necessary: one being the origin and m objects corresponding to the basis vectors. The question now arises how, given X_{red} w.r.t. to the principal axes, to choose a reduced set R_{red} of $r = m+1$ (or more) objects such that the projection defined by R_{red} gives a good approximation of the configuration X_{red}. To avoid an intractable search over all possible subsets, an error measure between the reduced and approximated configurations can be defined and then minimized, e.g. in a greedy approach. Such criteria are proposed and analyzed in an experimental study in section 9.3.

[8] Remember that X is an uncorrelated representation only if the origin coincides with the mean of X, i.e. obtained by using the centering matrix $J = (I - \mathbf{1}\mathbf{s}^T)$ with $\mathbf{s} = \frac{1}{n}\mathbf{1}$ in formula (3.7); see also Theorem 3.32. If some other $\mathbf{s}$ is used, then the PCA should be performed in a pseudo-Euclidean space.

3.3.8 Spherical embeddings

Nonlinear projection methods can rely on the geodesic distance, i.e. the shortest path between two points on a manifold; see e.g. [241, 242, 393]. Euclidean distance is the geodesic distance on a hyperplane. Since there exists a natural connection between the spherical geodesic distance and a linear Euclidean embedding, also spherical embeddings are here considered.

Def. 3.47 (Spherical distance) Let $\mathcal{S}_r^m \subset \mathbb{R}^{m+1}$ be an m-dimensional spherical space, such that $\sum_{i=1}^{m+1} x_i^2 = r^2$. We assume that the center of the sphere lies at the origin. The spherical distance d_s between two vectors $\mathbf{x}, \mathbf{y} \in \mathcal{S}_r^m$ is $d_s(\mathbf{x}, \mathbf{y}) = r \arccos \frac{\mathbf{x}^T\mathbf{y}}{r^2}$. This is the geodesic distance on the sphere, which coincides with the angle between the vectors.

Given an $n \times n$ dissimilarity matrix D and a positive r, a question arises whether there exist points: $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ on a sphere $\mathcal{S}_r^m$ such that the spherical distance $d_s(\mathbf{x}_i, \mathbf{x}_j) = d_{ij}$.

Theorem 3.48 (Schoenberg) Let D be an $n \times n$ dissimilarity matrix. D can be embedded into a spherical space $\mathcal{S}_r^m$ for a positive r iff $d_{ij} \leq \pi\, r$ for all $i, j = 1, 2, \ldots, n$ and $G = (\cos(d_{ij}/r))$ is psd [342]. Then the smallest m such that D embeds into a spherical space $\mathcal{S}_r^m$ is $m = \text{rank}(G) - 1$. The solution is undefined for $\text{rank}(G) = 1$.

Proof. Although the proof can be found in [342], we present it here since a problem of the embedding of D into the spherical space $\mathcal{S}_r^m$ can be transformed into the problem of embedding of some matrix into a Euclidean space. The requirement of $d_{ij} \leq \pi\, r$ is obvious, since no distance on the sphere with the radius r can exceed $\pi\, r$. Suppose that $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ lie on a sphere $\mathcal{S}_r^m$. Let $\mathbf{x}_0$ be the center of $\mathcal{S}_r^m$. Then, all $\mathbf{x}_i$ form an n-simplex in $\mathbb{R}^{m+1}$, whose edges have the lengths:[9] $\rho_{0i} := \overline{\mathbf{x}_0\, \mathbf{x}_i} = r$ and $\rho_{ij} := \overline{\mathbf{x}_i\, \mathbf{x}_j} = 2\, r\, (\sin \frac{d_{ij}}{2r})$ for $i, j == 1, \ldots, n$. Consequently, we have to prove that the distance matrix $D_\rho = (\rho_{ij})$ is Euclidean (or equivalently, that D_ρ embeds into a Euclidean space). Based on Theorems 3.32 and 3.35, D_ρ is Euclidean if $S = -\frac{1}{2}\,(I - \mathbf{1}\mathbf{s}^T)\, D_\rho^{*2}\, (I - \mathbf{s}\mathbf{1}^T)$ for $\mathbf{s}^T\mathbf{1} = 1$ is psd. Since in our case, $\mathbf{x}_0$ should become the origin (i.e. the center of the sphere), then we choose $\mathbf{s} = \mathbf{e}_1$. Since $D_\rho^{*2}\mathbf{e}_1 = r^2(\mathbf{1} - \mathbf{e}_1)$, then after straightforward basic transformations we get $S = r^2((\mathbf{1} - \mathbf{e}_1)\mathbf{1}^T - \frac{1}{2}D_\rho^{*2})$. Since $\mathbf{x}_0 := \mathbf{0}$, it is, therefore, sufficient to consider a matrix G which is S without the first row and the first column. Then, $G = r^2\, (\mathbf{1}\mathbf{1}^T - 2\, \sin^2(d_{ij}/(2r)) = r^2\, (\cos(d_{ij}/r))$. The condition that S is psd, imposes also that G should be psd, which finishes the proof. ■

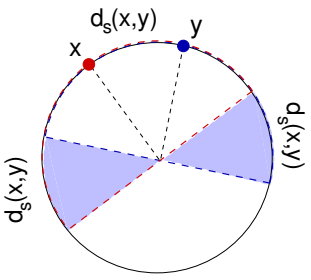
Consequently, we have also proved the following:

Theorem 3.49 An $n \times n$ dissimilarity matrix D can be embedded into a spherical space iff $D_\rho = (\rho_{ij})$, $i, j = 0, 1, \ldots, n$, with $\rho_{0j} = r$ and $\rho_{ij} = 2\, r\, (\sin(d_{ij}/(2r))$, $i = 1, \ldots n$ is Euclidean.

Note that the embedded points are found by applying Theorem 3.35 (with $\mathbf{s} = \mathbf{e}_1$) to D_ρ.

Corollary 3.50 Spherical distances $(\mathcal{S}_r^m, d_s)$ are l_1-embeddable.

Sketch of proof. Based on Theorem 3.9, it is sufficient to show the existence of a nonnegative measure space such that d_s is the measure of the symmetric difference. Let $\mathcal{S}_1^m$ be an m-dimensional sphere with the radius $r = 1$. Define the measure μ on $\mathcal{S}_1^m$ as the fraction of m-dimensional hypervolume such that $\mu(A) := \frac{vol(A)}{vol(\mathcal{S}_1^m)} \in [0, 1]$ for $A \subseteq \mathcal{S}_1^m$. Consequently, $(\mathcal{S}_1^m, \mathcal{A}, \mu)$ for $\mathcal{A}$ being a collection of subsets of $\mathcal{S}_1^m$ is a probability space. Consider further a hemisphere $H_1^m(\mathbf{x}) := \{\mathbf{y} \in \mathcal{S}_1^m : d_s(\mathbf{x}, \mathbf{y}) \leq \frac{1}{2}\}$ centered at x. Then, $\mu\,(H_1^m(\mathbf{x}) \triangle H_1^m(\mathbf{y})) = 2\, \frac{1}{vol(\mathcal{S}_1^m)}\, \frac{\arccos(\mathbf{x}^T\mathbf{y})}{2\pi}\, vol(\mathcal{S}_1^m) = \frac{1}{\pi}\, d_s(\mathbf{x}, \mathbf{y})$. On the pictorial illustration on the right, $\mu\,(H_1^m(\mathbf{x}) \triangle H_1^m(\mathbf{y}))$ is the volume of the shaded regions. Based on Theorem 3.9, the space $(\mathcal{S}_1^m, d_s)$ is l_1-embeddable.



[9] From geometry $\rho_{ij}^2 = \rho_{0i}^2 + \rho_{0j}^2 - 2\, \rho_{0i}\, \rho_{0j}\, \cos \frac{d_{ij}}{r}$, which after basic transformation gives $\rho_{ij} = 2\, \sin \frac{d_{ij}}{2r}$.

3.4 Spatial representation of dissimilarities

Given dissimilarity data[10] $D(R,R)$, a spatial representation of D is a configuration of points representing the objects in a space. Usually, a Euclidean (pseudo-Euclidean) space is considered or, alternatively, $\mathbb{R}^m$ equipped with an l_p metric. Spatial representations are in fact approximate embeddings, which should reflect the dissimilarity relations between the objects. Hence, they are often used as a visualization tool. Such spatial representations are visually appealing and often allow for a better interpretation of the data. The configurations are believed to reflect significant characteristics, as well as 'hidden structures' of the data. Therefore, objects judged to be similar result in points being close to each other in a low-dimensional space. The larger the dissimilarity between two objects, the further apart they should be in the resulting map of points. More generally, spatial representations are interpreted as (possibly reduced in complexity) feature-space configurations of the overall dissimilarity structure in the data and they are further utilized in clustering, classification or data-mining algorithms and techniques.

We already know from the previous section that (approximate) linear embeddings of the dissimilarities are methods for obtaining spatial representations. In this section, we will discuss two more techniques, namely a linear projection FastMap [122] and nonlinear multidimensional scaling (MDS); see e.g. [37, 72, 228]. One crucial thing to realize about such spatial maps is that the axes are, in themselves, meaningless. What is important is the relative positions of the objects. In case of a Euclidean space, additionally, the orientation of the projection is arbitrary, since any rotation of the configuration does not change the distances (the same is valid for pseudo-Euclidean spaces, however, in terms of a rotation defined appropriately there). See Fig.3.8 for an illustration of the basic spatial models on a theoretical banana data.

A number of other techniques exists for obtaining spatial representations. These will be briefly introduced in chapter 7, where also some practical aspects of spatial models are considered.

3.4.1 FastMap

FastMap was introduced in [122] in the Data Mining community and it is meant for vectorial data accompanied by a distance measure. Assume that a set $R := \{p_1, \ldots, p_n\}$ and an $n \times n$ Euclidean distance matrix $D(R,R)$ are given. Then, there exists an m-dimensional Euclidean space, $m \leq n$ such that the distances are preserved perfectly; see section 3.3.1. The idea is to project the data on m mutually orthogonal directions. This will be achieved in an incremented manner, starting from the first dimension. The basic principle is to orthogonally project p_i into a line in $\mathbb{R}^m$ determined by two *pivot* objects, r_1 and r_2. Pivot objects should be the ones which yield the largest distance. The projection x_i of the object p_i into this line can be determined from the cosine law of the Euclidean geometry (as illustrated in Fig. 3.7) as:

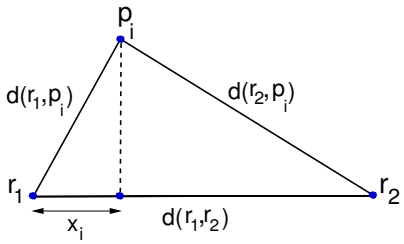
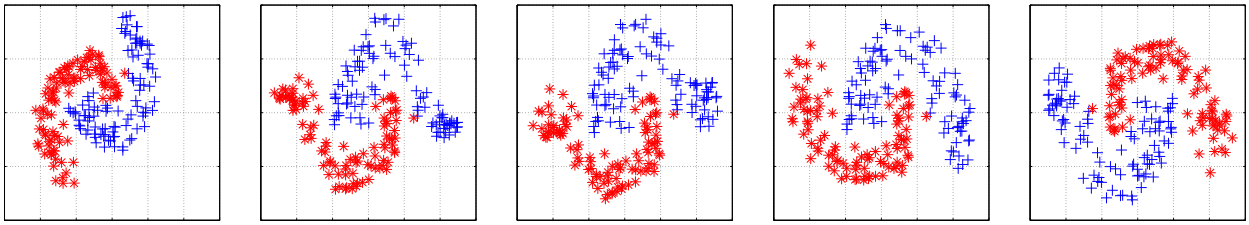


Fig. 3.7: Cosine law.

$$x_i = \frac{d(r_1,r_2)^2 + d(r_1,t_i)^2 - d(r_2,t_i)^2}{2\,d(r_1,r_2)}. \tag{3.21}$$

Since the objects will lie in a Euclidean space $\mathbb{R}^m$, the projection method can be extended as follows. Let H be an $(m-1)$-dimensional hyperplane perpendicular to the line defined by r_1 and r_2 (or the remaining $\mathbb{R}^{m-1}$ space). Then, after mapping all the objects on this hyperplane (or in fact

[10] We assume that we have an access to a set of raw data examples R, e.g. typed words, shapes, digitized voice excerpts, and a dissimilarity measure provided by an expert. Since the computation of dissimilarities is usually costly, in practical applications, R can be assumed to be a relatively small set of objects e.g. chosen from a larger set T.



Theoretical data | Classical Scaling | FastMap | Sammon map: S_{-1} | Sammon map: S_1

Fig. 3.8: Spatial 2D maps of 200×200 dissimilarity matrix D for theoretical banana data (leftmost subplot). The dissimilarity used is the city block distance. Since D is non-Euclidean, Theorem 3.41, the 2D maps are only approximations. The scale is the same for all subplots.

updating the distances appropriately), the problem to be faced is identical to the original one, but with a dimensionality $m-1$, instead. Hence, the solution can be found recursively using formula (3.21) to determine the coordinates of the dimension of interest. Since the square Euclidean distances are additive, the distances of the objects projected into the hyperplane H become then [122]: $d_H(p_i,p_j)^2 = d(p_i,p_j)^2 - (x_i - x_j)^2$. In the next step $D := D_H$, defining the same problem, but for the space $\mathbb{R}^{m-1}$. Although the dimensionality m should be specified beforehand, the algorithm may stop when the distances d_H become practically zero. New points can be added to the existing map in the same recursive manner, based on the distances to the pivot objects. The algorithm requires then the computation of $2m$ distances, so the complexity is $\mathcal{O}(m)$.

Note that the cosine law captures the same relation as the one given by formula (3.2). The Euclidean embedding realized by classical scaling, as described in section 3.3.1, makes also use of the cosine law, however, the projection is optimized for all the triplets (defining Euclidean triangles) simultaneously, instead of in an incremental way as FastMap does. Note that for non-Euclidean dissimilarities the derived configuration approximates the original distances, since the cosine law (which is the foundation of FastMap) is valid for the Euclidean distances only. In the mapping process, at some point the distances d_H of the objects projected into the hyperplane H may become negative, which indicates that H exists in a pseudo-Euclidean space. Yet, the projection is always done to a Euclidean space. Formally, if the dissimilarity data D can be embedded into the $\mathbb{R}^{(p,q)}$ space, then the dimensionality m used in FastMap should be such that $m \leq p$, since FastMap preserves, in fact, Euclidean distances corresponding to the embedding into $\mathbb{R}^p$. In summary, FastMap is less optimal than classical scaling w.r.t. the preservation of distances, but it is an incremental mapping (hence a possibility to an early stopping), which makes it fast.

3.4.2 Multidimensional scaling

Multidimensional scaling (MDS) refers to a group of linear and nonlinear projection methods of the dissimilarities. Although the theory of MDS was developed in behavioral and social sciences [37, 72, 228], its applications were extended to pattern recognition and other related disciplines, since the MDS methods facilitate data visualization and exploration. These projection techniques aim to preserve all pairwise, symmetric dissimilarities between data objects, resulting in a low-dimensional representation of the geometrical relations between the points. Such a configuration is usually found in a Euclidean space, although any other l_p space, $p \geq 1$, can also be considered [37, 72]. Therefore, the output of MDS is a spatial representation of the data. Most of the concepts presented here as well as the discussion on the MDS algorithms can be found in the recent books of Borg and Groenen [37] and Cox and Cox [72]. The latter book provides a good, concise introduction into the subject, while the former book is meant as a thorough compendium. Our work is concerned with Sammon mapping and it is based on our research project for Shell [297–299, 302] and our experience gained in this area.

Metric MDS is a description of methods which assume that both the input data and the output configuration are metric, or rather that the dissimilarities are quantitative values. Suppose a set of n objects with the dissimilarities, measured between all pairs of objects, is given. Our aim is to find a possibly low-dimensional space such that the discrepancy between the original dissimilarities and the estimated distances is minimized. Intuitively, each pairwise distance corresponds to a 'spring' between two points in this lower-dimensional space. Then, the MDS technique tries to rearrange the points such that the overall 'stress' is minimized. The dissimilarities can describe the relations between objects represented originally in a high-dimensional space or just measured (e.g. matching costs of image patterns, road distances) or given (human judgments).

When the observed or measured dissimilarities convey qualitative instead of quantitative information, they give rise to *non-metric* MDS methods. In essence, they are solved in a similar way as metric MDS methods with the exception that the nature of dissimilarities is different, such as preferences or ranks [37, 72, 227]. Since such methods are not discussed, MDS will stand for metric MDS.

There are different ways of preserving the structure of the data, giving rise to somewhat different techniques of MDS. Traditional classical scaling (CS) is the most simple, a linear MDS algorithm. It has already been introduced in section 3.3, where the embeddings of pseudo-Euclidean distances have been discussed. Also that FastMap can be considered as a linear MDS example.

Nonlinear MDS. Nonlinear MDS projections are realized via an iterative optimization process. In such a process, a criterion is needed for deciding whether one configuration is better than another. For that purpose, a loss function, called *stress* (acronym for *st*andard *re*sidual *s*um of *s*quares), is considered, which measures the difference between the Euclidean (or l_p) distances of the present configuration of n points in $\mathbb{R}^m$ and the actual (given) dissimilarities. Here, for convenience, we will adopt the notation used in MDS. Let Δ be the given $n \times n$ dissimilarity matrix, expressing all the pairwise relations between n objects, and let D be the distance matrix for the projected configuration of estimated distances. For clarity, we will write $d_{ij}(X)$ to indicate that the distances are computed for a retrieved configuration X. The most elementary MDS loss function is the *raw stress* [37, 226]:

$$S^{raw}(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (f(\delta_{ij}) - d_{ij}(X))^2, \tag{3.22}$$

which yields, in fact, a square badness-of-fit measure for the entire representation. f is a continuous parametric monotonic function, a transformation applied to the given dissimilarities δ_{ij}. In many cases, f is the identity function, but it may be a polynomial or logarithmic function as well. Usually, the notation of $\hat{\delta}_{ij} = f(\delta_{ij})$, called *disparity*, is adopted. In our opinion, the raw stress as an absolute error is not an informative function to be minimized iteratively. (Yet, the raw stress is also used in practice e.g. [37].) The differences between actual and estimated dissimilarities should rather be expressed in relative terms to avoid that large absolute differences contribute significantly to the error function, while small differences do not. Note that the large differences do not necessarily indicate a bad approximation. Therefore, the stress should be normalized in a way that avoids a scale dependency. This leads to a *least squares scaling* (LSS) [72, 227, 228] loss function:

$$S^{LSS}(X) = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} d_{ij}^2(X)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} (\hat{\delta}_{ij} - d_{ij}(X))^2, \tag{3.23}$$

where w_{ij} are appropriately chosen weights. The purpose of the weights can be e.g. to shift the emphasis to small dissimilarities by choosing $w_{ij} = 1/\delta_{ij}$ for non-zero δ_{ij}. Concerning disparities, some straightforward choices are e.g. a linear or logarithmic function, i.e. $\hat{\delta}_{ij} = \alpha + \beta\delta_{ij}$ or $\hat{\delta}_{ij} =$

$\alpha + \log(\delta_{ij})$, where α and β are estimated in the least square sense by modeling the perfect relation $d_{ij}(X) = \hat{\delta}_{ij}$. The normalization by estimated distances makes the error measure invariant under rigid transformations, like shifts and rotations, and non-rigid transformations, like uniform stretching or shrinking, of the derived configuration.

The problem of finding the right spatial configuration resolves itself into an optimization problem, where a configuration yielding the minimum of the stress is sought. The stress is optimal when all the original disparities $\hat{\delta}_{ij}$ are equal to the estimated distances $d_{ij}(X)$. Since this is unlikely to happen, $d_{ij}(X)$ will be a distorted representation of the relations within the data. The larger the stress, the greater the distortion. The optimization procedure for the LSS is an iterative process of two alternating stages: fitting $\hat{\delta}_{ij}$ to d_{ij} for a present configuration X (hence d_{ij} are considered as fixed for that moment) and minimization of the stress function, i.e. updating X, given $\hat{\delta}_{ij}$.

Since from an application point of view, one is interested in the relative positions of objects in the spatial map, a general suggestion in the MDS area is to consider a *ratio* MDS [37], where $\hat{\delta}_{ij} = \beta\delta_{ij}$ for $\beta > 0$. It means that the ratio of two disparities should be equal to the corresponding dissimilarities: $\hat{\delta}_{ij}/\hat{\delta}_{kl} = \delta_{ij}/\delta_{kl}$. For the S^{LSS} stress, the optimal β^*_{LSS} can be derived analytically as the one minimizing S^{LSS}, provided that D is fixed. By setting up the derivative of S^{LSS} over β to zero, its optimal value is found as $\beta^*_{LSS} = \sum_{j<i} d^2_{ij}(X) / \sum_{j<i} \delta_{ij} d_{ij}(X)$. Alternating the computation of β^*_{LSS} with an iterative improvement to the stress provides an efficient procedure for finding the solution to the ratio MDS.

Most of the minimization algorithms are based on gradient methods [37, 227, 228], but also other techniques have been especially adopted for the MDS purposes, such as iterative majorization [37, 72]. In our experience [297], this algorithm has a slow convergence. An interesting modification for data originally supplied by points in a feature space is also studied by Webb [414, 415]. He looks for a nonlinear transformation to the reduced space $\mathbb{R}^m$ in which the approximated Minkowski distances are close to the actual Minkowski distances in terms of the weighted raw stress. The transformation is defined by radial basis functions, hence the iterative majorization technique determines its parameters. This results in a mapping that can be applied to new data.

Another way to normalize the raw stress is to use the original dissimilarities instead of the approximated ones. This leads to loss functions being variants of the Sammon mapping.

Sammon mapping. The original Sammon mapping [329] was proposed in pattern recognition by Sammon [329] as a tool for a nonlinear projection from a high-dimensional Euclidean space to a low-dimensional space. To our knowledge, it is not directly mentioned in books and articles devoted to the MDS research. However, it can be considered as a method in this area, if interpreted as a projection technique which tries to preserve the original dissimilarities. For the sake of simplicity, we will account variants of Sammon mappings as the MDS examples. Sammon mapping is a nonlinear projection realized by the minimization of the following loss function:

$$S(X) = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\delta_{ij}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \frac{(\delta_{ij} - d_{ij}(X))^2}{\delta_{ij}}. \tag{3.24}$$

In general, the stress function can be defined in a number of ways, e.g. as studied by us in [297–299]:

$$S_t(X) = \frac{1}{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\hat{\delta}_{ij}^{t+2}} \sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \hat{\delta}_{ij}^{t}\,(\hat{\delta}_{ij} - d_{ij}(X))^2, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots \tag{3.25}$$

which results in the following measures for the identity function f, i.e. $\hat{\delta}_{ij} = f(\delta_{ij}) = \delta_{ij}$:

$$S_{-2}(X) = \frac{2}{(n-1)(n-2)} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\frac{\delta_{ij} - d_{ij}(X)}{\delta_{ij}} \right)^2 \tag{3.26}$$

$$S_{-1}(X) = S(X)$$

$$S_0(X) = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}^2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (\delta_{ij} - d_{ij}(X))^2$$

$$S_1(X) = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}^3} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij} (\delta_{ij} - d_{ij}(X))^2$$

$$S_2(X) = \frac{1}{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}^4} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \delta_{ij}^2 (\delta_{ij} - d_{ij}(X))^2$$

We will refer to all of them as (variants of) Sammon mappings. Each of the loss functions mentioned above emphasizes a different aspect of the geometric relations between points, i.e. it emphasizes, to some extent, either smaller or larger distances, which directly influences either local or global aspect of the method. For instance, S_{-2} emphasizes very small distances, i.e. it penalizes the error in representing small dissimilarities more than the same error for large ones. Therefore, S_{-2} focuses on local details, hence it is very nonlinear. On the other hand, S_2 emphasizes larger distances, hence it tends to present more global map of relations. S_0 provides a balance between large and small distances, i.e. errors in representing small and large dissimilarities are penalized equally. Depending on the application requirements, the loss function can be chosen appropriately.

By applying the ratio approach to the Sammon stresses, i.e. the discrepancy $\hat{\delta}_{ij} = \beta \delta_{ij}$, one gets

$$S_t(X, \beta) = \frac{1}{\sum_{i<j} \beta^2 \delta_{ij}^{t+2}} \sum_{i<j} \delta_{ij}^t \, (\beta \delta_{ij} - d_{ij}(X))^2, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots \tag{3.27}$$

Note also that the scaling of δ_{ij} by β is equivalent to scaling of $d_{ij}(X)$ by $1/\beta$, which is further equivalent to scaling of X by $1/\beta$, i.e. $S_t(X, \beta) = S_t(1/\beta\, X, 1)$. The optimal β^* can be determined as the point yielding minimum of S_t for the present configuration X (hence also D). By setting the first derivative of $S_t(X, \beta)$ w.r.t. β to zero, after straightforward calculations, one obtains $\beta^* = (\sum_{j<i} \delta_{ij}^t \, d_{ij}^2(X)) / (\sum_{j<i} \delta_{ij}^{t+1} d_{ij}(X))$. After simplifications, inserting β^* into $S_t(X, \beta)$ yields

$$S_t(X, \beta^*) = 1 - \left(\frac{\sum_{j<i} \delta_{ij}^{t+1} d_{ij}(X)}{(\sum_{j<i} \delta_{ij}^{t+2})^{1/2} \, (\sum_{j<i} \delta_{ij}^t d_{ij}^2(X))^{1/2}} \right)^2, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots \tag{3.28}$$

Note that $0 \leq S_t(X, \beta^*) \leq 1$ by the nonnegativity of dissimilarities and the Schwartz inequality, Theorem 2.56, since $\sum_{j<i} \delta_{ij}^{t/2+1} (\delta_{ij}^{t/2} d_{ij}(X)) \leq (\sum_{j<i} \delta_{ij}^{t+2})^{1/2} (\sum_{j<i} \delta_{ij}^t d_{ij}^2(X))^{1/2}$.

In order to compare the Sammon stress functions to the LSS loss functions (3.23), let us introduce the variants of the S^{LSS} in the same way as for the Sammon mappings as

$$S_t^{LSS}(X) = \frac{1}{\sum_{i<j} d_{ij}^{t+2}(X)} \sum_{i<j} d_{ij}^t(X) \, (\hat{\delta}_{ij} - d_{ij}(X))^2, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots \tag{3.29}$$

Then, by considering the ratio MDS, $\hat{\delta}_{ij} = \beta \delta_{ij}$, one can express the optimal β (minimizing $S_t^{LSS}(X, \beta)$) as $\beta_{LSS}^* = (\sum_{j<i} \delta_{ij} \, d_{ij}^{t+1}(X)) (\sum_{j<i} \delta_{ij}^2 d_{ij}^t(X))$. The substitution of β_{LSS}^* into S_t^{LSS}

gives then

$$S_t^{LSS}(X,\beta^*_{LSS}) = 1 - \left(\frac{\sum_{j<i} \delta_{ij}\, d_{ij}^{t+1}(X)}{(\sum_{j<i} d_{ij}^{t+2}(X))^{1/2}\,(\sum_{j<i} \delta_{ij}^2 d_{ij}^{t}(X))^{1/2}} \right)^2, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots \tag{3.30}$$

If X^* is the optimal configuration (corresponding to a local minimum) of the Sammon error S_t, then the LSS stress $S_t^{LSS}(X^*, \beta^*_{LSS})$ is equal to the Sammon stress $S_t(X^*, \beta^*)$ for $t = 0$. This does not hold for other t, although for $t < 0$, the Sammon stress S_t at the local minimum of X^* would be smaller than the corresponding S_t^{LSS} and the other way around for $t > 0$. This can be directly deduced from the formulations of (3.30) and (3.28), taking into account that the MDS distances $d_{ij}(X)$ underestimate the actual dissimilarities, which leads to the following inequalities $\sum_{j<i} \delta_{ij} > \sum_{j<i} d_{ij}$ and $\sum_{j<i} \delta_{ij}/d_{ij} > 1$.

Note that except for the raw stress, both S_0^{LSS} and S_0 are the loss functions traditionally applied in MDS. Practically, they give the same (up to scaling and rotation) results. In general, due to the normalization by the actual dissimilarities, S_t will emphasize smaller dissimilarities than S_t^{LSS} for $t < 0$ and the other way around for $t > 0$. This is not a problem, since when needed, additional weights can be used. This means that the Sammon stress functions can be generalized in order to incorporate the nonnegative weights of individual pairs as

$$S_t^w = \frac{1}{\sum_{j<i} w_{ij} \hat{\delta}_{ij}^{t+2}} \sum_{j<i} w_{ij} \hat{\delta}_{ij}^{t} \,(\hat{\delta}_{ij} - d_{ij}(X))^2. \tag{3.31}$$

Usually, the weights are chosen to be either 0 or 1, where 0 is used to accommodate for missing dissimilarities. However, the weights can also be set to $1/\delta_{ij}$ or $1/\delta_{ij}^2$ for non-zero δ_{ij}. For instance, in the latter case, the weighted stress S_0^w becomes the unweighted stress S_{-2}.

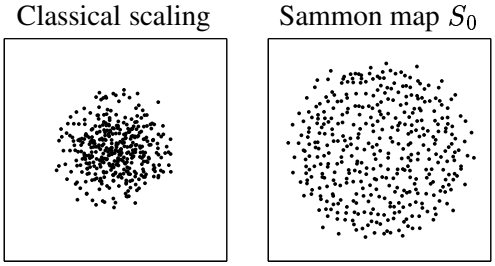


Fig. 3.9: 2D spatial maps of the Euclidean representation of 400 points uniformly distributed in a $10D$ space. The scale is preserved.

Since the optimization of the Sammon stress functions can be easier defined in the gradient terms, we give preference to Sammon mappings. To find a Sammon representation, one starts from the initial configuration of points X (e.g. randomly chosen or from the classical scaling result) in $\mathbb{R}^m$ for which all the pairwise distances are computed. Next, the points are adjusted so that the stress decreases. In an iterative process, the configuration is improved by shifting around all points to approximate better and better the model relation $\hat{\delta}_{ij} = d_{ij}$ for $i, j = 1, 2, \ldots, n$, until a (local) minimum of the stress is reached. In such a procedure, a steepest descent, Newton-Raphson algorithm [308], iterative majorization [37, 193], conjugate gradients [308] or scaled conjugate gradients (SCG) [272] can be used to search for the minimum. In our experiments with artificial and real data [297–299], we found out that concerning the convergence rate, the scaled conjugate gradients and Newton-Raphson techniques are preferable. The SCG algorithms allows for large improvements in the first iterations, but it approaches the minimum slowly later on. Therefore, a hybrid algorithm can be considered, which switches to the Newton-Raphson minimization after the first iterations.

The found minimum depends on the initialization. Usually, the output of classical scaling is a good suggestion, since it is the global minimizer of the raw stress (in a linear way). However, it is useful to compare its result to the Sammon output obtained from a random initialization, since the optimization algorithm may get stuck in a local minimum close to the initial configuration. Recently, a 'better' initial configuration, i.e. a scaled version of the CS result X_{CS} i.e. $t^* X_{CX}$ has been suggested in

[255, 400]. It is, however, anything new, since their t^* equals $\sum_{j<i} \delta_{ij} d_{ij}(X) / \sum_{j<i} d_{ij}^2(X)$, which is equivalent to $1/\beta^*$ in the ratio MDS with the stress S_0 (and also S_0^{LSS}). A good initialization is still an open problem. From our experience is follows that Sammon mappings are less sensitive to the starting configuration than the LSS mappings.

In summary, it is important to emphasize that the MDS techniques based on the minimization of the normalized square differences will produce maps, where projected points will tend to be enclosed in circular or ellipsoidal shapes. This can clearly be observed for a Euclidean distance matrix D computed for an artificial example of 400 points uniformly distributed in a $10D$ hypercube. The MDS result can be seen in Fig. 3.9. See [200] for the formal proofs referring to the raw stress based on square Euclidean distances. To avoid this artifact, other types of error measure can be considered, for instance in the form of $E(X) = \sum_{j<i} |\delta_{ij} - d_{ij}(X)| / \sum_{j<i} \delta_{ij}$ or $E(X) = \sum_{j<i} |\delta_{ij} - d_{ij}(X)| / \delta_{ij}$. The measures based on absolute values are, however, difficult to optimize (due to discontinuous derivatives). Another MDS technique, which is more robust against outliers can also be designed by considering the following fit $F(X) = \text{median}_{i,j \neq i} \frac{|\delta_{ij} - d_{ij}(X)|}{\delta_{ij}}$ [72].

Two different spatial configurations can be matched by the use of *Procrustes analysis*. This might be useful to compare two configurations derived from optimizations of different loss functions or to indicate how a configuration changes when the similarity between objects changes over time (as e.g. human preferences of some products). Basically, the configurations are matched be determining the optimal translations, rotations and scalings; see e.g. [37, 72] for details.

3.4.3 Reduction of complexity

For n objects a nonlinear MDS method requires the computation of $\mathcal{O}(n^2)$ distances in each iteration step and the same memory storage. However, for a low, m-dimensional representation, only mn values should be determined. This suggests that a number of constraints on distances is redundant, so some of them could be neglected. This leads to the idea that only distances to a subset of all objects could be preserved, for which a modified version of the MDS mapping will be considered.

Although X, derived from MDS, has the dimensionality m, it is determined by $n > m$ objects. In general, a linear space can be defined by $m+1$ linearly independent objects. If they were placed such that one lies in the origin and the others lie on the axes, they would determine the space exactly. Since this is unlikely, the space retrieved will be an approximation of the original one. When more objects are used, the space becomes more filled, hence better defined. Following [65], objects having relatively many close neighbors (lying in the areas of high density) can selected for the representation set $R \subseteq T$ of the size $r > m$, on which the (non-)linear mapping could be based. For a dissimilarity representation $\Delta(T, T)$, a natural way to proceed is the k-centers algorithm [426]. It looks for k center objects, i.e. examples that minimize the maximum of the distances over all objects to their nearest neighbors; see also section 7.1.2. It uses a forward search strategy, starting from a random initialization. Note that the k-means algorithm [97] cannot be used since no feature representation is assumed, only the dissimilarities Δ.

For a chosen set R, the linear mapping (based on $\Delta(R, R)$) into an m-dimensional space is defined by formulas (3.7)–(3.14). The remaining objects $\Delta(T \setminus R, R)$ can then be added to the map by the use of Corollary 3.46 and formula (3.20).

In case of the Sammon mapping, a modified version should be defined, which generalizes to new objects. Following [65], first the Sammon mapping of $\Delta(R, R)$ into the space $\mathbb{R}^m$ is performed, yielding the configuration X_R^*. The remaining objects can be mapped to this space, while preserving the dissimilarities to the set R, i.e. $\Delta' = \Delta(T \setminus R, R)$. This can be done via an iterative minimization

procedure of the modified stress M_t, using the found representation X_R^* as:

$$M_t = \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{r} f(\delta'_{ij})^{t+2}} \sum_{i=1}^{n}\sum_{j=1}^{r} \left(f(\delta'_{ij})^t \left(f(\delta'_{ij}) - d_{ij}(Y, X_R^*)\right)^2 \right),\ t = \ldots, -2, -1, 0, 1, 2, \ldots \tag{3.32}$$

Equivalently, the modified loss functions of the LSS and non-metric MDS can be defined as follows:

$$M_t^{LSS} = \frac{1}{\sum_{i=1}^{n}\sum_{j=1}^{r} d_{ij}^{t+2}(Y, X_R^*)} \sum_{i=1}^{n}\sum_{j=1}^{r} \left(d_{ij}^t(Y, X_R^*) \left(f(\delta'_{ij}) - d_{ij}(Y, X_R^*)\right)^2 \right). \tag{3.33}$$

Thanks to these procedures, new objects can be added to an existing map. Their complexity reduces from $\mathcal{O}(mn^2)$, computing $\mathcal{O}(n^2)$ distances in the $\mathbb{R}^m$ space, to $\mathcal{O}(nmr + nr^2)$ in each iteration step. Another possibility to define a Sammon mapping is by the use of neural networks, as studied in [260, 316].

3.5 Summary

This chapter presents some ways of characterizing dissimilarity measures, especially if they are represented as finite generalized metric spaces. The basic concern is whether a dissimilarity measure is metric or not, which can be easily checked for a finite representation. Also transformations preserving the metric properties are considered. Usually, such transformations can also make a non-Euclidean dissimilarity 'more' Euclidean. A more essential question, however, is whether a dissimilarity measure is Euclidean or city block. The importance of the Euclidean distance comes from the fact that a Euclidean space is both metric and an inner product space. Hence, there exists a natural connection between the traditional inner product and Euclidean distance, which allows one to embed any Euclidean distance matrix in a finite Euclidean space. Both isometric and approximate, linear and nonlinear embeddings are presented here, as well as their generalizations, which enable the projection of new examples. These are the multidimensional scaling techniques.

If a measure is non-Euclidean, no isometric projection into a Euclidean space exists. Some solutions are presented, where either the dissimilarity is corrected to become Euclidean or the projection is carried out into a pseudo-Euclidean space. Any premetric non-Euclidean measure (satisfying the definiteness and symmetry constraints) can be formalized in such an indefinite inner product space. This builds a general framework, where any symmetric dissimilarity representation can be explained.

On the other hand, the significance of the city block distance comes through its additivity property. Finite generalized metric spaces can also be represented by weighted, fully connected graphs, where the weights correspond to given dissimilarity values. A city block distance can be perfectly structured by an additive tree model, where the distance is understood in terms of the shortest path. Other dissimilarity measures can also be interpreted via such tree models, however, only approximately. See also chapter 7 for more discussion.

In brief, this chapter deals with the characterization of generalized metric spaces, especially finite spaces represented by $n \times n$ dissimilarity matrices. It introduces useful tools for checking metric or Euclidean properties and finding the dependencies in the family of Minkowski dissimilarities. In particular, this chapter discusses the issue of (approximate) embeddings into pseudo-Euclidean spaces which can be carried out for any symmetric dissimilarity measures. In this way, the foundation has been established for designing learning algorithms on spatial representations in Euclidean and pseudo-Euclidean spaces, as to be seen in chapter 4. Also the process of data exploration is supported either by visualization of 2-dimensional spatial maps or by the inference of the organization of objects, i.e. understanding of the underlying structure between them, given by a tree model.

4. Learning approaches

> *The learning and knowledge that we have, is, at the most, but little compared with that of which we are ignorant.*
>
> PLATO

In this thesis, although objects may have various intermediate representations, as given by relative graphs or numerical features, ultimately we will describe them by dissimilarities. Given such a numerical representation of classes of objects, for instance, learning paradigms are set up in some spaces, where the dissimilarity values can be interpreted. This leads to the use of statistical learning methods, which are briefly described in section 4.1.

By now we have established a ground for introducing the learning methodologies. First of all, various spaces and the relations between them have been characterized in chapter 2. They prepare mathematical frameworks, in which dissimilarities can be explained. Next, basic properties and transformations of dissimilarity matrices as representations of finite generalized metric spaces have been discussed in chapter 3, especially in the context of metric or Euclidean distances. Also general embedding issues have been described there. Finally, statistical learning aspects are recapitulated for the feature-based representations.

In section 4.2, we will formally introduce dissimilarity representations and explain their unifying role for the statistical and structural approaches to learning from examples. This section also mentions a possible extension of dissimilarity representations as the ones based on the 'true' inductive learning, as illuminated by Goldfarb [153, 156, 160, 161]. This is, however, left for further research. Next, three main dissimilarity-based learning approaches are presented. In fact, they refer to three interpretations of such representations in some spaces, for which particular statistical learning methodology can be adapted. In the first approach, the dissimilarity values are interpreted directly, hence they can be characterized in (pre)topological spaces. The second approach serves for the definition of dissimilarity spaces, where each dimension corresponds to a dissimilarity to a chosen object. The third approach finds a spatial representation, i.e. an embedded (pseudo-)Euclidean configuration such that the dissimilarities are preserved as well as possible. More details on these methodologies is given in the sections 4.3 - 4.5. This chapter ends up with some additional remarks on generalized kernels as well as some insights on the connections between dissimilarity spaces and the underlying pseudo-Euclidean spaces, as given in section 4.6.

Although some of this material is extracted from our publications [293, 301, 304, 305], there are many new insights and observations presented here. Also the perspective from which it is discussed, is new. So, a significant part of this chapter is our contribution to the pattern recognition field. Even a description of statistical learning, section 4.1.2, aims at establishing the context, where learning from dissimilarities will take place. The purpose of this chapter is not only educational, but presents the ideas behind the learning from dissimilarity representations and explains the basic methods.

4.1 Traditional learning

Learning from examples is a process where patterns present in the data are discovered, distinguished, detected or described. It relies on both extraction and representation of information from the collected measurements in order to understand the process (phenomenon) that created them. The result of learning is that the knowledge already captured in some mathematical terms is used to de-

scribe the present independences such that the relations between patterns are better understood or it is a formalization of some concept, e.g. of a class, such that it can be applied to unseen examples of the same domain. The latter means that data objects should obey the same deduction process, hence, a process of (simple) reasoning is imitated. Note that the word 'pattern' refers to both, a property of an individual object (i.e. its structural or mathematical representation) and a property of the entire set of objects given by their characteristics.

4.1.1 Data bias and model bias

In pattern recognition one is usually concerned with learning of a concept from a set of examples. Here, a concept is a general notion of an entity serving to designate a class of instances or other type of relations. More practically, an abstract or real set of all possible examples of the concept to be learned is a domain. For instance, if one wants to learn a concept of a dutch tulip (in fact of a tulip class), then a domain consists of all types of tulips ever grown in The Netherlands. So, a domain is a complete representation of the concept considered. In practical applications, due to the complexity of the domains, costs of the collection process, physical limitations of both measuring and storing devices and the measurement costs, the domains in their entirety cannot be studied. Consequently, domains are sampled. This means that only some examples are provided to represent a domain, and, as a result, only a limited amount of data is available for learning purposes. (Although there are problems, where the collection of data is either a relatively easy or cheap process, like gathering of web documents or scanning of handwritten digits, there are also problems where the collection of measurements is a mundane and costly task, for instance in medical diagnostics.) Consequently, *data represent information and knowledge that one has available for a particular domain*.

A (concept of a) class[1] is represented by a finite collection of instances, but it is not yet by this described. The description of a class has to be based on the description of each single instance in the measurement process, where each instance is characterized by a set measurements and additional knowledge that one has on the class. Measurements, in general, refer to the outputs of measuring tools, algorithms or procedures and they can be performed directly on the objects or inferred from raw measurements. Raw measurements refer to the raw outputs of sensors or devices which record signals, images, hyper-spectra images etc. All such outputs can serve for a definition of relational descriptions, features or a proximity measure. On the other hand, an abstract domain might be represented by some example structures, order in the data or inference rules, provided from outside. In such cases, the standard measurement process may not play a direct role, instead, the support is given by structural representations. It is often difficult, if possible, to define numerical features. Still, a proximity measure can be usually constructed.

Data introduce a *bias* ('a systematic error introduced in sampling or testing by selecting or encouraging one outcome over others' [416]) of the domain we wish to learn. We have a bias with respect to the chosen representation and to the chosen model such as a learning approach. The latter is caused by a dissonance between the learning procedure imposed on the data and the validity of the assumptions. Such a *model bias* is related to some error measuring the discrepancy between the assumed and learned values, so it is related to a bias of an estimator of an ideal model; see also section 4.1.2. *Data bias* refers to both domain and data description. The first one, *sampling bias*, comes by assuming that data examples are representative for the domain. Since it is often impossible to supply instances describing all the domain variety, a finite sample gives rise to the sampling bias. The *representation bias* results from a selection of characteristic features, proximity measure or a structural representation. Taking into account the efficiency concerning both data representation

[1] A class is either a natural category, i.e. present in reality, like a class of tomatoes or mugs, or an abstract category consisting of objects or instances sharing some common properties considered for the application's need, e.g. articles on sport, human silhouettes, people with a particular disease, etc.

and learning algorithms, as well as the resolution of measuring devices, it is impossible to consider an infinite set of features, infinitesimally precise proximity measure or complex and detailed structural information. The necessary simplification or redundancy of the data representation introduces the representation bias.

Data bias has important implications regarding the learning algorithms. It strongly contributes to the model bias; if the data examples are a poor representation of the domain, then the selected model, optimized by using the given examples, does not describe the reality well. Data are well described if similar objects are close in their representations (e.g. two similar objects are represented by two vectors, which lie close in a vector space), the so-called *compactness hypothesis* [4, 98, 102] and, two close descriptions correspond to the objects that resemble each other, the so-called *true representation* [323]. The basic principle is that the objects do not posses random descriptions, on the contrary, the neighbors of a particular object in the representation are similar to it in reality. Note that true representation implies that distinct objects lie far away (with respect to the chosen dissimilarity) in the representation. This means that the measurements contain sufficient information to both support the resembling objects and tell them apart from distinct objects. Moreover, data are well sampled if all instances in the domain are somehow described in data or, in other words, if adding new instances will not change this description significantly.

Given a lot of data relevant to the problem at hand (actually with respect to a model; see also footnote 5 on page 78), the learning task becomes relatively easy (in the methodological sense; the computational cost may increase), since the data bias becomes smaller. Consequently, if the data are representative and well sampled, there is enough support and information in the data to model their functional dependencies, hence the model bias becomes smaller, as well. Only such data will assure a good generalization of a learning algorithm. The problematic situations are these where the amount of data is small or when there are many unlabeled examples (sometimes the collection of the data can be automated, while the labeling process is slow and expensive since it should be done by humans).

Conventionally, data are described by features. For instance, the class (concept) of apples (domain) can be represented by features (obtained in the measurement process) such as weight, size and color. A feature-based representation of a concept relies on selecting n instances to represent the domain and on defining, say, m features for the description. So, we can think of vertical and horizontal samplings, where these samplings coincide with the choice of objects and features, respectively. Such data are often expressed as an $n \times m$ matrix A, where A is interpreted as a configuration of n points (feature vectors) in an m-dimensional feature space $\mathbb{R}^m$, usually Euclidean. This representation is mainly used in statistical pattern recognition [97, 138], where it is assumed that the distribution of pattern classes can be derived from a representative set of such points (a training set) with a sufficient accuracy. This often requires (strict) additional assumptions on the distribution characteristics.

4.1.2 Statistical learning

Statistical learning[2] is usually understood as a process of determining an unknown dependency between some inputs and outputs given a limited number of observations, i.e. training examples. A probabilistic framework is often considered for this task, as mathematically appealing for handling uncertainty. Input vectors $x \in \mathcal{X}$, usually $\mathcal{X} := \mathbb{R}^m$, are assumed to be drawn independently from a fixed, but *unknown* probability density function $p(x)$. The functional dependency between outputs $y \in Y$ and inputs x is given as a fixed conditional density $p(y|x)$, which is also unknown. De-

[2] For a more elaborate introduction to statistical learning, see the books of Fukunaga [138], Stork et al. [97], Hastie et al. [191], Vapnik [402, 403] and Schölkopf et al. [352].

pending on the domain of Y, different learning problems can be presented. If Y is discrete, then one deals with a classification problem, while for continuous Y, a regression problem is obtained. The training examples $T_n := \{(x_i, y_i) : i = 1, \ldots, n\}$ are considered to be iid (independent and identically distributed) according to the joint probability density[3] $p(x, y) = p(x)\, p(y|x)$. The difficulty relies on determining the relationship between $\mathcal{X}$ and Y, based on T_n only.

We are often interested in prediction and, therefore, in modeling the conditional probability of observing a particular y given a specific x. By Bayes rule, one can write $p(y|x) = \frac{p(x|y)p(y)}{p(x)}$. Assuming that the quantity $p(y|x)$ can be computed, the most appealing approach for assigning an output to a new x is the value of y which yields the maximum a posterior probability $p(y|x)$. This is known as a theoretical optimal Bayes rule. In practice, since the true distributions are unknown, the Bayes rule cannot be found. One, therefore, tries to estimate this ideal by a function $g(x)$ coming from a general hypothesis space of functions $\mathcal{G} := \{g : \mathcal{X} \to Y'\}$, where Y' is e.g. $\mathbb{R}^1$, $\{0, 1\}$ or $\{-1, 1\}$, depending on the task. The goal of learning is then formulated as a selection of $g^* \in \mathcal{G}$ which best approximates the outputs y. To measure the discrepancy (hence define the best fit) between the estimated outcome $g(x)$ and the original output y for a given x, a *loss* function $L : Y \times Y' \to [0, M]$ is needed. A single output $L(g(x), y)$, however, is not very informative over a particular function g. Rather the overall expected loss should be used to infer about g. This is the *true loss* of the hypothesis g, given by the *error* or *risk* functional as

$$\mathcal{E}(g) = \int_{\mathcal{X} \times Y} L(y, g(x))\, p(x, y)\, dx\, dy. \tag{4.1}$$

Ideally, the learning is a process of estimating $g_* \in \mathcal{G}$ which minimizes the error $\mathcal{E}(g)$. This requires the integration over the complete probability distribution of all possible inputs x and outputs y. Since $p(x, y)$ is unknown and the only information is a set of available training examples T_n that is available, the learning problem is ill-posed. To make the learning task feasible, one usually considers a specified class of functions $\{g_\alpha\}$ (e.g. polynomials), where α are parameters indexing the functions (e.g. polynomial degrees). Then $g_{\alpha *} \in \mathcal{G}$ minimizes the error $\mathcal{E}(g_\alpha)$. Note that the true, optimal Bayes solution g_* does not necessarily belong to $\{g_\alpha\}$. To tackle such a learning problem, an *empirical* error (or risk) is minimized. It is expressed as:

$$\mathcal{E}_{emp}(g_\alpha, T_n) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, g_\alpha(x_i)). \tag{4.2}$$

For a given finite training set T_n there might be infinitely many functions minimizing the empirical error, since they need to behave identically only for the training examples. Therefore, by the selection of a class of functions $\{g_\alpha\}$, i.e. narrowing the scope of interest, the learning task is better formulated. Note, however, that this is purely a choice made to be able to tackle the learning problem, unless some other prior knowledge exists.

Depending on the loss function, basic learning problems such as classification, regression, density estimation and clustering can be set up in this statistical framework. Since knowing $p(x, y)$ would allow one to solve any learning problem expressed by the minimization of the risk, the density estimation is the most general (hence most difficult) problem. Here, we will focus on predictive learning such as classification and regression.

Classification. A general multi-class classification problem can be decomposed into a number of two-class problems [138]. Hence, a two-class problem is considered as the basic one. Assume a set of training examples $\{(x_i, y_i)\}_{i=1}^n$, with the corresponding labels $y_i \in \{0, 1\}$ (sometimes also

[3] All these assumptions, although general, are in fact strong. They actually assume a fixed (stationary) distribution from which the examples are sampled. This is often violated in practical applications, e.g. when the data are collected in various conditions or even by differently calibrated sensors.

$y_i \in \{-1, 1\}$). The hypothesis space becomes then a set of indicator functions. The most common loss function is then $L(y, g_\alpha(x)) = \mathcal{I}(y \neq g_\alpha(x))$. The corresponding risk or the *true error* $\varepsilon_T := \mathcal{E}(g_\alpha)$ denotes the probability of misclassification (given equal costs). The empirical error, also called the *training* or *apparent* error, becomes then $\varepsilon_A := \mathcal{E}_{emp}(g_\alpha, T_n) = \frac{1}{n}\sum_{i=1}^{n} \mathcal{I}(y_i \neq g_\alpha(x_i))$. As a result, the learning can be simplified to finding a *classifier* $g_{\hat{\alpha}*}(x)$ that minimizes the empirical error.

Regression. Regression is based on estimating a functional dependence f between inputs x and outputs y in the form of $y = f(x) + \varepsilon$, where ε is such that $E(\varepsilon|x) := \int_Y \varepsilon\, p(y|x)\, dy = 0$, i.e. random noise with a zero mean. f is then seen as the expectation of the output conditional probability $f(x) = \int y\, p(y|x)\, dy$. The risk measures the dissonance between the actual outputs and the expected predictions with the common loss function being $L(y, g_\alpha(x)) = (y - g_\alpha(x))^2$. Under the assumption of a zero mean noise and based on the fact that $y - g_\alpha(x) = y - f(x) + f(x) - g_\alpha(x)$, the risk (4.1) can be decomposed[4] as a sum of two contributions, noise variance and approximation accuracy as:

$$\mathcal{E}(g_\alpha) = \int (y - f(x))^2 p(x, y)\, dx\, dy + \int (f(x) - g_\alpha(x))^2 p(x)\, dx = e_0 + \int (f(x) - g_\alpha(x))^2 p(x)\, dx, \quad (4.3)$$

where e_0 is a fixed value, since it does not depend on g_α. So, learning can be now stated as determining $g_{\alpha*} \in \mathcal{G}$ that best approximates the (unknown) f. The empirical risk with respect to the set of functions $\{g_\alpha\}$ is expressed as $\mathcal{E}_{emp}(g_\alpha, T_n) = \frac{1}{n}\sum_{i=1}^{n} (y_i - g_\alpha(x_i))^2$.

4.1.3 Inductive principles

Predictive learning such as regression or classification consists of two steps: the process of learning, i.e. the estimation of an (unknown) dependence between inputs and outputs, and the process of generalization, i.e. prediction of outcomes for newly coming examples based on the discovered concept. In practice, the first step is closely related to induction ('inference of a generalized conclusion from particular instances' [416]), while the second step refers to deduction ('derivation of a conclusion by reasoning' [416]). In a general form, however, the deduction is much simplified; it involves only the computation of outcomes based on the derived parameters in the learning stage. Probably, that is why such a process is called an *inductive* learning paradigm. The minimization of the expected risk relies on this principle. Hence, the entire problem is put the framework of a global function estimation.

Another approach is based on estimating the risk functional by using the training set at the moment, when a new example appears. This requires a reformulation of the learning problem such that additional unlabeled examples are treated in the context of the given training set. So, the dependence between the training data and test examples is estimated when required and may differ from instance to instance. This approach is called *transductive* inference [403]. Such an inference might be applied locally (the unknown examples are related to the objects in local neighborhoods), but not necessarily. If it is applied globally, the computational burden might become high (under the inductive paradigm, only *one* final functional dependence is estimated). Examples of this approach are the cases of learning from partially labeled sets or designing linear classifiers in local neighborhoods. This inference can also be reduced to the deduction step only, like in the k-NN rule for a fixed k. A schematic illustration of the inductive and transductive learning principles is shown in Fig. 4.1.

Statistical learning theory is mostly developed for inductive principles. This is somewhat surpris-

[4] One has: $\int_{\mathcal{X}\times Y} (y - g_\alpha(x))^2\, dx\, dy = \int_{\mathcal{X}\times Y} (y - f(x))^2 p(x, y)\, dx\, dy + \int_{\mathcal{X}\times Y} (f(x) - g_\alpha(x))^2 p(x, y)\, dx\, dy + 2\int_{\mathcal{X}\times Y} (y - f(x))\,(f(x) - g_\alpha(x)) p(x, y)\, dx\, dy$. Note that the latter term equals zero: $\int_{\mathcal{X}\times Y} \varepsilon\,(f(x) - g_\alpha(x))\, p(y|x) p(x)\, dx\, dy = \int_{\mathcal{X}} (f(x) - g_\alpha(x)) \big[\int_Y \varepsilon\, p(y|x) dy\big] p(x)\, dx = 0$, since ε is a random noise with a zero mean. Note also that $\int_{\mathcal{X}\times Y} (y - f(x))^2 p(x, y)\, dx\, dy = \int_{\mathcal{X}} (y - f(x))^2 p(x)\, dx$.

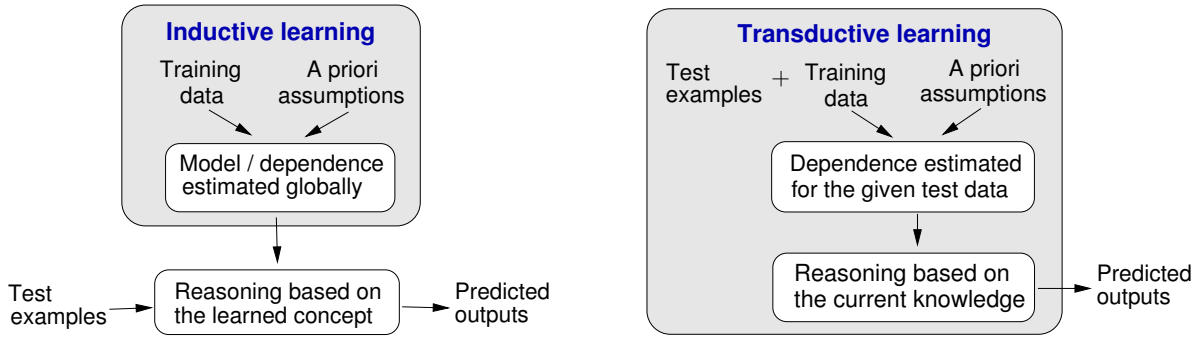


Fig. 4.1: Inductive (left) and transductive (right) learning paradigms. A priori assumptions are here understood in terms of the specified assumptions on a set of learning algorithms and related parameters.

ing, since in a general context of inference from small sample size training data[5], only restricted information is available. Vapnik [403] formulated the main learning principle as: *'If you posses a restricted amount of information for solving some problem, try to solve the problem directly and never solve a more general problem as an intermediate step.'* Following this rule, we conclude that not only a predictive learning problem should be approached directly (instead of e.g. estimating the probability density function first as usual parametric methods do), but, more importantly, that it should be solved only *for the points of interest*, instead of estimating a single function globally at the entire domain. This means that the learning problem is solved 'at the spot'. Consequently, the application of this principle naturally leads to a transductive learning. This type of learning has not yet evoked sufficient interest of researches, probably due to the expected computational cost in a testing stage. Yet, it becomes one of the open issues for further research.

In this dissertation, we will focus on inductive learning methods. These provide a general prescription for handling the data vectors and the assumptions on the approximating functions in the learning process. Here, the empirical risk minimization and a few paradigms based on the Occam's razor principle are considered within the framework of inductive principles.

4.1.3.1 Empirical risk minimization (ERM)

In this paradigm, a function $g_{\hat{\alpha}*}$ is sought such that the empirical error, i.e. the training error $\varepsilon_A := \mathcal{E}_{emp}(g_\alpha, T_n) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, g_\alpha(x_i))$ is minimized. The training error is a rough (biased) approximation of the (unknown) true error. We assume that an optimal function g_* (hence the Bayes rule) exists in $\mathcal{G}$, although the class of functions $\{g_\alpha\} \in \mathcal{G}$ might not contain it. In classification, the actual risk $\mathcal{E}(g_*)$ is the minimal risk ever possible, called also the *Bayes error*.

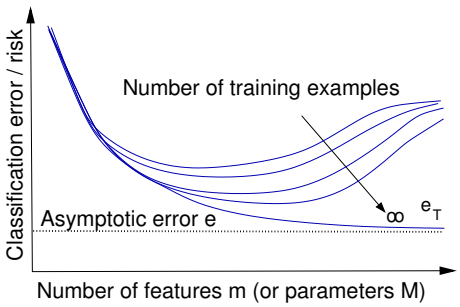
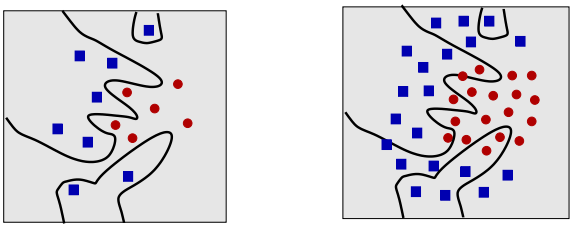


Fig. 4.2: Curse of dimensionality.

Depending on a loss function and a set of chosen g_α, the ERM can be employed in a number of ways, e.g. based on the maximum likelihood estimators or linear regression. Usually, this principle is used in the *parametric* methods, where a model is specified first (e.g. a normal density-based linear classifier assuming normal distributions) and then the parameters are estimated from the training data. This works well, provided that the number of training examples is large with respect to the model complexity (Vapnik's approach: related to the VC dimension, see section 4.1.3.2; classical approach: related to the

[5] In the classical sense, a small sample size problem is understood as an inference from n data vectors for an estimation of M free parameters of the approximating function used, where n/M is small, e.g. 2 or even ≤ 1. Vapnik [403] defines it with respect to a class of approximating functions of the VC dimensionality h_{vc} as a problem, where n/h_{vc} is small, such as 10. See section 4.1.3.2 for more details.



(a) Training set (b) Test set

Fig. 4.3: Overtraining. (a) A zero-error classifier. (b) This classifier is overtrained, since it yields a high error on an independent test set and its boundary is too complex for such a small training set.

number of free parameters, which agrees with the Vapnik's approach for polynomial classifiers). Such models do not have enough flexibility, hence they can result in a large bias (see below).

The difficulty of applying the ERM for limited training data is that it does not yet guarantee a small expected risk, i.e. the true error. In the classification case it means that a small error on the training set does not imply a small error on an independent test set. The phenomenon that $g_{\hat{\alpha}*}$ yields a small empirical risk, but still shows a large true error on an independent test set is called *overtraining* or *overfitting*. A sufficiently flexible function can perfectly fit the training data, completely adapting to all the information available there, reaching a zero empirical error. As a result, this function can describe structures (due to the noise) which in fact are not present in the data; see Fig. 4.3. Hence, to avoid overtraining for fixed and small sample sizes, simple models are preferred to the complex ones. The problem is much more pronounced when the number of features, hence the dimensionality m, is very large with respect to the number of data vectors. From the classical point of view, by adding new features while having a fixed number of objects, worse results can be obtained for an independent test set. This is caused by a poor estimation of the function parameters due to insufficient amount of the data vectors. This is called the *curse of dimensionality* [208, 210]. See also Fig. 4.2. There exist some solutions to the curse of dimensionality, e.g. feature selection [86] or feature extraction [97] techniques. The first ones find the best few features, while the latter construct new features functionally depending on the old ones, e.g. as their linear combination. Still, such procedures might not be sufficient to guarantee a good generalization for complex functions.

Bias-variance dilemma. The empirical risk depends on training examples, hence different training sets will yield different models $g_{\hat{\alpha}*}(T_n)$. Consequently, the loss function is also a function of a training set. This dependency can be removed by averaging over training sets of a fixed size. Then, the expected empirical risk with respect to all the training sets of cardinality n becomes $E_n[\mathcal{E}_{emp}(g_\alpha, T_n)]$, where $E_n[\cdot]$ denotes the expectation. In the case of regression, there is a clear decomposition of the latter quantity into a (squared) bias term[5], measuring 'the accuracy or a quality of the match' of the learning algorithm to the problem [97] and a variance term, measuring 'the precision or specificity of the match' [97]. Additionally, there is an irreducible term e_0, independent from the training sets as derived from formula (4.3) by using a summation instead of an integral. Hence, we have that $E_n[\mathcal{E}_{emp}(g_\alpha, T_n)] := e_0 + E_n\left[\frac{1}{n}\sum_i (g_\alpha(x_i) - f(x_i))^2\right] + E_n\left[\frac{1}{n}\sum_i (g_\alpha(x_i) - E_n[g_\alpha(x_i)])^2\right]$. This decomposition indicates that there exists a bias-variance trade-off, which is a fundamental problem while fitting a model to the data [145]. The practical implication of such a trade-off is that a flexible function g_α, i.e. a function which is able to model the irregularities well, will have a high variance since it will tend to fit the desired outputs well (yielding a smaller bias). Consequently, it will vary dramatically between various training sets. Conversely, an inflexible model will tend to behave similarly with respect to the training sets, yielding small variance, but its inflexibility might cause a high bias [186].

[5] Here, bias is understood as a bias of an estimator. If θ is a random variable, then the estimator $\hat{\theta}$ is biased if the bias $b := E[\hat{\theta}] - \theta$, where $E[\cdot]$ is the expectation, is non-zero.

A bias-variance decomposition becomes more complicated for the zero-one loss in the classification case. Although it is possible to extend the reasoning behind the square loss to a classification problem [97, 145], there is no clear interpretation of this phenomenon. A unified bias-variance decomposition was proposed by Domingos [91], of which the zero-one loss is a special case. According to him, the variance contribution is additive for unbiased examples (i.e. examples wrongly classified) and subtractive, otherwise. This means that the zero-one loss allows for a larger tolerance of a learning algorithm with respect to variance than in case of the square loss. This follows from the offset contribution to the averaged loss (i.e. empirical error) by the biased examples. This explanation is logical, since in the end, the classification problem directly focuses on a proper assignment of objects to classes and not on a proper estimation of probability functions. See [90, 91] for details.

Consistency. To assure that a small empirical error guarantees a small true error, a consistency between the true and empirical risks is needed. For a fixed function g_α, the empirical error (4.2) will converge to the true risk $\mathcal{E}(g_\alpha)$ (4.1) by the law of large numbers. But this is not enough, since it should hold for any g_α. Let $g_{\alpha*}$ minimize the true error, i.e. $g_{\alpha*} = \arg\min_{g_\alpha \in \mathcal{G}} \mathcal{E}(g_\alpha)$ and let $g_{\hat{\alpha}*}$ minimize the empirical risk (hence it depends on T_n). Then, the consistency of the ERM principle requires that $\lim_{n\to\infty} \mathcal{E}(g_{\hat{\alpha}*}) = \lim_{n\to\infty} \mathcal{E}_{emp}(g_{\hat{\alpha}*}, T_n) = \mathcal{E}(g_{\alpha*})$ holds in probability. This requires a one-sided uniform convergence of the empirical risk to the actual risk in probability [403], which is the necessary and sufficient condition for such a convergence:

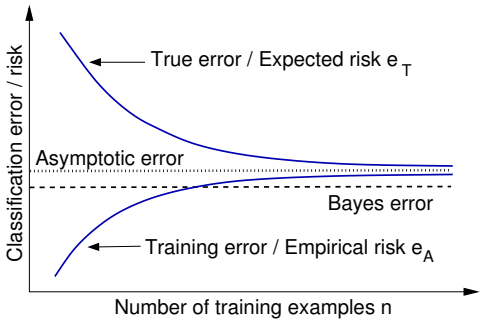


Fig. 4.4: Consistency of the ERM.

$$\forall_{\varepsilon>0} \quad \lim_{n\to\infty} P\Big\{ \sup_{g_\alpha \in \mathcal{G}} \left(\mathcal{E}(g_\alpha) - \mathcal{E}_{emp}(g_\alpha, T_n)\right) > \varepsilon \Big\} = 0. \tag{4.4}$$

This is illustrated in Fig. 4.4. Note that the asymptotic error might differ from the Bayes error.

4.1.3.2 Principles based on Occam's razor

Some statistical approaches have been developed to assure this convergence. These often rely on the Occam's razor principle. Assume a learning problem and a set of functions $\{g_\alpha\}$, depending on the parameters α, analyzed to find a solution. The learning problem is now complex since it relies on the estimation of both: the model structure or complexity (the degree of a polynomial), called the *model selection* of the parameters (coefficients) in some optimization procedure. Such methods are put in paradigms more general than the ERM. It is assumed that the best prediction is achieved for a model of the right complexity, found by applying the *Occam's razor* principle. This principle states that one should not presume more things than the required minimum; in the selection process, among otherwise equivalent models, it advocates to choose the simplest one. The Occam's razor principle can be implemented in a number of ways, taking into account that there is a trade-off between the model complexity (e.g. the number of free parameters) and the model fit to the training data. The most typical examples are: structural risk minimization, regularization principle, Bayesian inference and minimum description length. We will focus on the first two principles.

Structural Risk Minimization (SRM). The approximating functions are ordered according to their complexity (like ordering polynomials by the degree) such that a nested structure is formed. The complexity of functions linear in parameters is related to the number of parameters. In general, it is estimated by the so-called Vapnik-Chervonenkis (VC) dimension h_{vc} [403], which describes the capacity of a set of functions $\{g_\alpha\} \in \mathcal{G}$. In case of a binary classification, h_{vc} is equivalent to the maximal number of points N which can be separated into two classes in all 2^N ways by using functions from the considered set $\{g_\alpha\}$. It means that for each possible labeling of N points into two

classes, there exists a function from $\{g_\alpha\}$ which takes 1 for examples coming from one class and -1 (or 0) for examples from the other class. An analytic upper-bound based on the VC dimension is provided by Vapnik [403] to estimate the expected risk. Given n training points, with the probability at least $1-\eta$, the bound below remains true:

$$\mathcal{E}(g_\alpha) \leq \mathcal{E}_{emp}(g_\alpha, T_n) + \sqrt{\frac{h_{vc}\left(\log\frac{2n}{h_{vc}}+1\right) - \log\frac{\eta}{4}}{n}}. \tag{4.5}$$

The estimate above is used for the model selection of the optimal complexity in the following way. For n training examples, the expected risk is controlled by two quantities: the empirical risk, which depends on the chosen function for particular α and the VC dimension h_{vc} of the considered set of functions. Therefore, in order to control h_{vc}, the approximating functions are ordered according to their complexity such that if $\mathcal{G}_k = \{g_\alpha \colon \alpha \in A_k\}$, where A_k is a set of parameters, and $\mathcal{G}_1 \subset \mathcal{G}_2 \subset \mathcal{G}_3 \subset \ldots$, the corresponding VC dimensions fulfill $h_{vc}^{\mathcal{G}_1} \leq h_{vc}^{\mathcal{G}_2} \leq h_{vc}^{\mathcal{G}_3} \leq \ldots$. The SRM principle chooses the function from a subset $\mathcal{G}_k$ for which the bound yields minimum. Note that the bound derivation is based on the worst-case scenario, since the VC dimension considers *all* possible labellings of an arbitrary configuration of points. The importance of this bound, however, is that it guarantees the uniform convergence of $\mathcal{E}_{emp}$ to the actual risk, formula (4.4), for a finite h_{vc} (which is a necessary and sufficient condition) [121, 403] and for the indicator functions (hence a classification problem). It might not be true for other functions [121].

Regularization principle. This principle assumes a flexible set of approximating functions $\{g_\alpha\}$, but the restriction in the solution results from an additional term capturing the complexity of the function g_α. So, a penalized risk is minimized:

$$\mathcal{E}_{pen}(g_\alpha, T_n) = \mathcal{E}_{emp}(\alpha, T_n) + \lambda\,\phi(g_\alpha(x)), \tag{4.6}$$

where ϕ is a nonnegative functional and the nonnegative λ, independent of the training data, controls the strength of the regularization. For $\lambda = 0$, the penalized risk reduces to the empirical risk, while for a large λ, a simple solution is obtained, mostly ignoring the training examples. Hence, the model estimate is described as a trade-off between fitting the data and a priori knowledge on the function's complexity (regularization term). The ϕ functional can be selected in many ways. The simplest method counts the number of free parameters in the function, while a more sophisticated method uses the l_2 norm of the parameters α or the curvature estimator of g_α. See also [147] for the relation between this principle and the SRM.

In statistical learning, data are assumed to be represented by vectors in bounded regions in a feature space, so the learned function should change smoothly over the space, avoiding high oscillations. The smoother the function, the lower its complexity. So, functions of lower complexity are preferred for finite sample sizes (the regularization term is meant to penalize complex functions more). In the limit (when the number of training examples grows to infinity) complex functions offer better solutions (the bias is small). In the case of classification, this phenomenon is illustrated in Fig. 4.5.

Bayesian inference. This principle assumes that a model M, e.g. a particular g_α, has been selected adequately to describe the problem. The parameters α of the model M are assumed to be drawn from a theoretical parameter distribution. So, a prior distribution over these unknown parameters α is specified to capture our beliefs about the problem before seeing the data. The inference is then based on the Bayes formula for updating the priors given the evidence from the data as $P(M|data) = (P(data|M)\,P(M))/P(data)$, where $P(M)$ (or in fact $P(\alpha)$) is the prior probability, $P(data)$ is the probability of observing the data, $P(data|M)$ is the likelihood, i.e. the probability that the data are generated by the model M and $P(M|data)$ is the posterior probability of a model M given the data. Hence, one tries to find a complete density function for the parameters α, e.g. specifying a Gaussian density. As a result, all possible parameter values, although in different degree, play some role. A preference for more simple models is encoded by encouraging particular prior distributions.

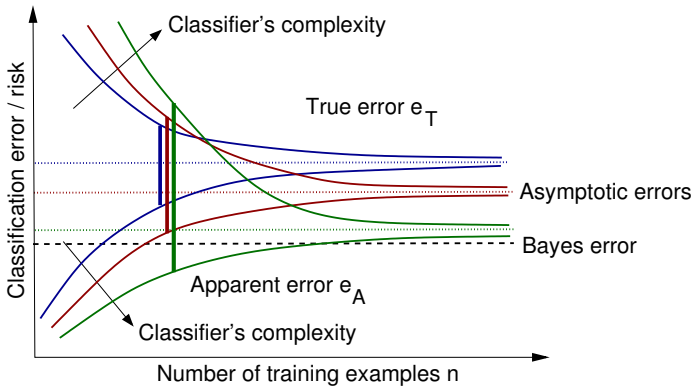


Fig. 4.5: Complexity of classifiers vs. cardinality of the training set.

Minimum description length (MDL). This principle is based on the information-theoretic basis and the concept of algorithmic complexity characterizing the randomness of the data. Briefly, it is related to the shortest binary code describing the output data. The output is split into two parts: model and noise contributions, which are encoded separately. The model is assumed to describe the regularities in the data and it should contain a few easily encoded parameters. For some relations between the MDL and the SRM, see [402].

4.1.4 Why is the statistical approach not good enough for learning from objects?

Even after such a brief review on statistical learning theory, the reader can be convinced that the methods developed in a proper framework guarantee good solutions. *The answer is affirmative, provided that the problems to be solved are originally generated as points in a vector space, such as Euclidean. There is a missing link between a collection of real or abstract objects to be learned from and their proper representations to be used as a basis in a learning paradigm.* In the statistical approach, the description of objects is dramatically reduced to points in a vector space. The analysis starts at this level, often neglecting the (one-way) correspondence between the objects and the points. Also such a simplification of an object to a numerical description only (i.e. without any structural information) precludes any inverse mapping, i.e. to retrieve the object itself (this is partly possible in the structural representation of objects, e.g. from the skeleton of an object in an image, its shape can be retrieved well enough). The objects are simply treated as if they have already been generated as points in the space. Note that these assumptions are very strong. Since the connection between the points and the objects is forgotten, any learning in such a framework is in fact learning with respect to the assumed distributions realized by a sample in the space $\mathbb{R}^m$. Hence, this learning is in a purely mathematical sense. Besides the guarantees on providing good learning solutions, one should be concerned with the guarantees that the representation of objects by points enables to achieve that. For all these reasons, Goldfarb and his colleagues [153, 155–157, 160, 161] strongly oppose the statistical approach to learning, separated from the 'real' objects themselves.

Real objects possess their internal structure, organization or 'interconnectivity', as e.g. observed in their shapes. This property can be reflected by the connectivity of neighboring samples in the sensory data, like in an image. However, in the traditional feature space representation, all the continuity of an object, all the structure is lost [59, 154, 156]. The structure information may partially be encoded in some feature values, e.g. when features are defined as the responses of several image (signal) filters, but in the representation itself it is not available anymore. This also holds for the pixel representation of images, in which each pixel defines a separate dimension in a feature space. The complete image resides there, but the fact that some pixels are neighboring and others are remote is not expressed in the representation[6]. In fact, the Euclidean feature space assumes in-

[6] The fact that consecutive features correspond to neighboring pixels in an image cannot be used in the feature-

dependent feature contributions and, therefore, it precludes the possibility of reflecting the structure of an image. The structure may be rediscovered to some extent by computing correlations between pixel-features from a set of images or by trying to find a low-dimensional manifold on which the set of images, represented as points, lies. Still, this is not the original structure. Moreover, the necessity of learning them in such a way is disputable if the primary structure of an image, reflected by the connectivity of neighboring pixels, is already given.

Structural representations, on the other hand, are specified in terms of instance's components and their interconnections. For a real object, it might be its structure. The structure, however, should be regular enough to be described by a relatively small number of primitives, i.e. fundamental structural elements, such as strokes, corners or other shape elements. For instance, shapes can be represented as their skeletons, contours by their string representations, where each character in the string corresponds to some kind of a stroke. Also more abstract instances, such as articles in a database can be represented structurally. An article might be organized in a hierarchical way, expressing the fact that it is composed of a title, an introduction, body, conclusions and references. The body might be e.g. an interview, a comment, a letter, a speech or a general writing. In turn, an interview can be made with a famous artist, actor, writer, scientist, etc. In this way, the detailed information on articles can be represented by trees. Other type of phenomena, e.g. a financial condition of a family, might be captured by graphs, expressing the relations between all the important factors, such as incomes of the family members, mortgage, loans, the number of children and their age, etc.

In general, structural PR [137] assumes that there exists sufficient and suitably formulated knowledge to build a structural description of objects and classes. This knowledge is defined and encoded either explicitly by an expert or implicitly by a set of (training) examples. In order to relate new objects to the described classes, a (dis)similarity measure between objects and the structural description of objects and/or classes is needed. Like in the statistical approach, the demands here are strong: suitable knowledge should be available to build the structural model and an informative dissimilarity measure should be defined between the model and real-world observations.

The research of pattern recognition is meant to establish a link between object representations, derived from their (sensor) measurements of objects or structural descriptions, and a learning algorithm; see also [110] for some perspectives. In the statistical learning theory, the bounds ensuring a good generalization (in classification: a small test error, given a small training error) are based on the notion of a classifier's complexity, which can be related to the VC dimension. For a binary classification problem, this notion is derived for the worst configuration of n (training) points and considering *all* 2^n labellings of them. This is a possible scenario to consider if the association to the (real or abstract) objects one started from is neglected. From the PR point of view, this is completely unrealistic to postulate a class of (similar) objects being described by arbitrarily labeled points. If this had been the case, one would have chosen another representation. Basically, the representation of objects is not accidental, it should be such that similar objects are close in their representations (of course, one should also define what the closeness mean). This is the compactness hypothesis [4, 98, 102]. (Ideally, also the true representation hypothesis [323] should hold, i.e. two close object representations correspond to objects that resemble each other). Note that in order to solve a real pattern recognition problem, such a subjective hypothesis is necessary to complement the available objective information. For that reason, the bounds derived by Vapnik are strongly over-pessimistic.

In summary, the answer to the question of this section is: *Suitable representations of objects should be considered first before using or adapting the developed learning methodology. Such representations should possibly embody a priori knowledge on the class of objects as well as their possible*

based representation. Features can be permuted and, in a Euclidean space, the configuration of points (representing the images) is the same up to the rotation, while the structure of an image is completely lost by shuffling the pixels.

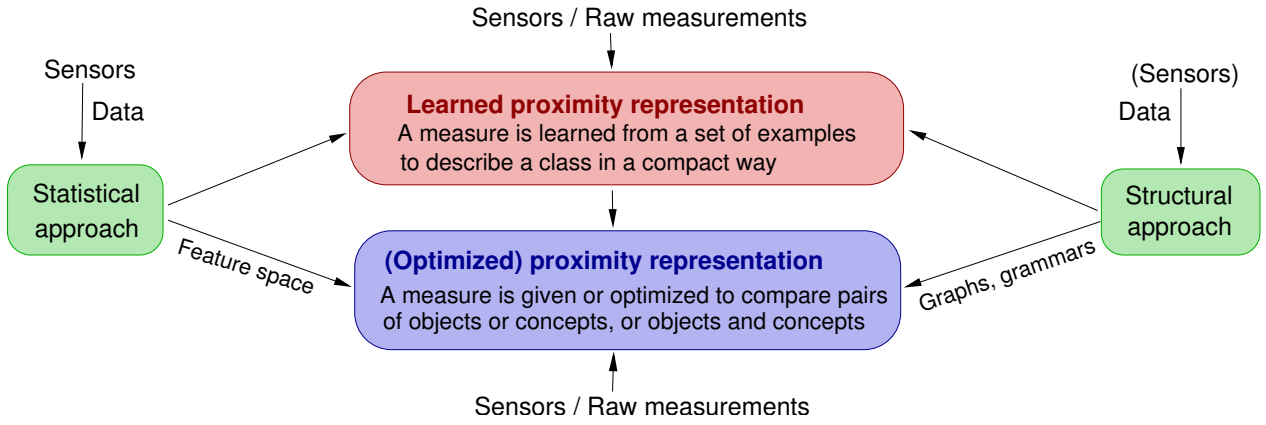


Fig. 4.6: Proximity as unifying statistical and structural approaches to learning [323]. This dissertation focuses on optimized dissimilarity representations.

structural information. There might be hybrid representations as well. Our work is concerned with an example representation based on the notion of (dis)similarity and by this relying on the compactness hypothesis. It is pioneering, not directly in the following methodology, but in the statement of the learning problem and resulting adaptations.

4.2 The role of dissimilarity representations

Pattern recognition relies on the description of regularities in observations of classes of objects. *A class is a set of similar objects* (e.g. sharing similar characteristics). This implies that the notion of 'similarity' is more fundamental than of a 'feature' or a 'class', since it is the similarity which serves for grouping objects together and, thereby, it should play a crucial role in class constitution [113, 115, 116]. Such a proximity should be possibly modeled such that a class has an efficient and compact description. In applications, however, features often come before proximity is taken into account. Using the notion of proximity (instead of features) as a primary concept renews the area of statistical learning in one of its foundations, i.e. the representation of objects [109, 180, 275, 276]. Conceptually, it is a novel approach, but some other researchers are conscious of the essential role that proximity plays for the class description [47, 113, 115, 151, 153, 160, 161, 206, 275, 276, 382]. Proximity measures can capture both the statistical and structural information of patterns and, thereby, they form a natural bridge between these approaches. Two main types of representations can be here considered: the ones which are learned and the ones which are fixed or optimized. This dissertation builds some foundations for the latter, called simply *proximity representations*. Learned representations remain an issue for further research.

Proximity representations can be divided into *relative* and *conceptual* representations. In the relative representation, pairs of objects are related and compared by measuring proximity between them. Consequently, each object is described by a set of proximities to other objects [106, 109, 293, 301]. They may be defined on a feature-based representation, see Fig.4.6, by using the distances between feature vectors, but also on the structural representation by distances between graphs or other structural models, or directly on the raw data, e.g. by similarities between shapes in images. So, proximity representations become very general as they combine all types of approaches. They describe the sampled domain in a relative way, based on the comparison of objects. Remember that an object is meant as a general notion of a real object, any entity process, phenomenon or any abstract instance. The basic principle is to be able to relate them to each other.

Proximity representations can be extended to depict a relation of one entity to a number of them or of a model to the whole concept. Such representations are called *conceptual* representations. Examples are a resemblance of a particular mug to a class of mugs, hence a similarity of an object

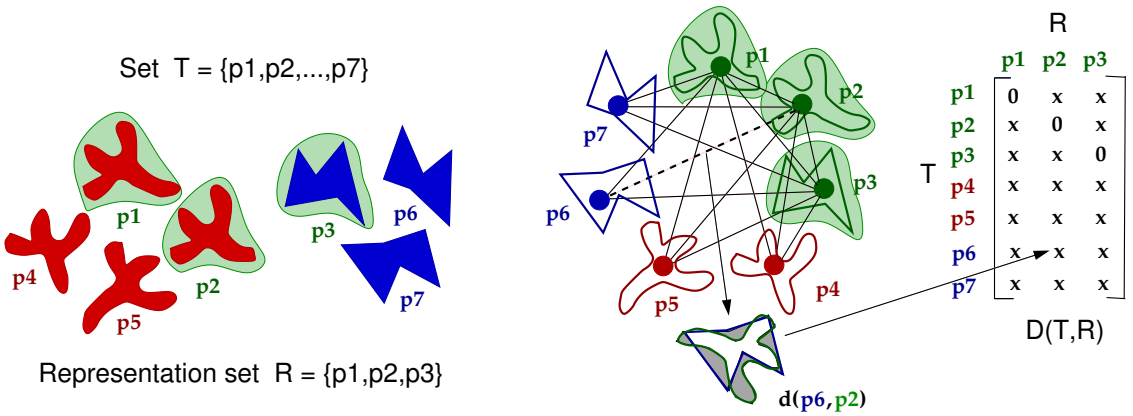


Fig. 4.7: A dissimilarity representation $D(T, R)$. Here, the representation objects come from the set T.

to a (sampled) domain, a similarity of a language to a group of European languages, a growth and development of a child to a model development or an image query serving the purpose of retrieving similar images, in a process of redefining the query. Also, in the statistical sense, the posterior probabilities of an object x (or in fact its feature-based representation) with respect to C classes, form a similarity conceptual representation $[P(x|\text{class } 1), \ldots, P(x|\text{class C})]$. Conceptual representations will appear in chapter 8, where one-class classifiers are built based on a proximity of an object to a class, and in chapter 10 in the context of classifier combining techniques. Now, we discuss relative representations, where our main focus is on *dissimilarity* representations.

Def. 4.1 (Dissimilarity representation) Assume a collection of objects $R := \{p_1, p_2, \ldots, p_n\}$, called a *representation set*, and a dissimilarity[7] measure d. The dissimilarity d is computed or derived from the objects directly, their sensor representations, or some other intermediate representations. To maintain generality, a notation of $d(p_i, p_j)$ is used, instead of $d(f(p_i), f(p_j))$, where $f(p_i)$ corresponds to some possible intermediate representation of p_i. A dissimilarity representation (DR) of an object x is a set of dissimilarities between x and the objects of R expressed as a vector $D(x, R) = [d(x, p_1),\ d(x, p_2), \ldots, d(x, p_n)]$. Consequently, for a collection of objects T, it extends to a dissimilarity matrix $D(T, R)$. The idea of a representation set is that R is a relatively small set of representative objects for the domain considered. The most simple DR is then $D(R, R)$, hence a square dissimilarity matrix with a zero diagonal. In general, R might be a subset of T ($R \subseteq T$) or they might be completely distinct sets. See also Fig. 4.7.

Although in a matrix notion, there exists some resemblance between dissimilarity and feature-based representations, yet, the meaning is completely different; see Fig. 4.8 for details.

Dissimilarity representations are meant to be used in (statistical) learning. An important question refers to the characteristics of informative dissimilarity measures. For instance, for a robust real-world object description, a measure should incorporate the necessary invariance, like translation, rotation, scale and illumination invariance. Essentially, the measure should be such that the compactness hypothesis [98, 102] holds, i.e. representations of similar objects are similar. This means that a small variation of an object should impose only a small change of a proximity value, hence the natural variation of objects of the same class should be captured there. Many dissimilarity measures are constructed by solving object matching problems, often defined in terms of the minimization of the mean square error or mean absolute error by the use of affine transformations. This often corresponds to the Euclidean or city block distance, which may not fully integrate the mentioned invariances. Such computed distances can not directly capture the structural information of the ob-

[7] If d is a similarity (or a proximity) measure, the corresponding representation is called appropriately. d is expected to capture the notion of closeness between two objects, however it might be non-metric. In general, we require that d is nonnegative and obeys the reflexivity condition; see Def. 2.30.

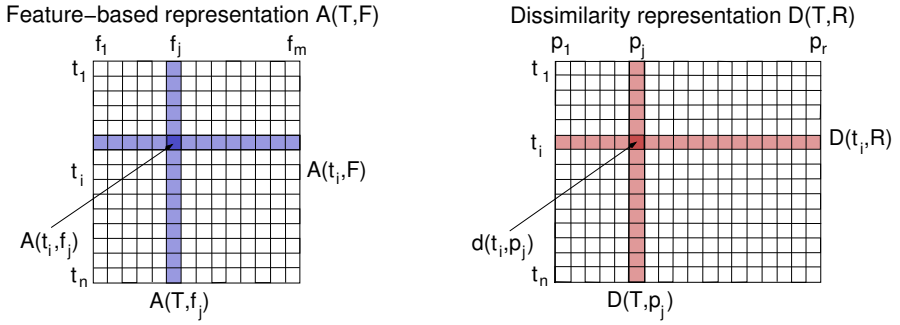


Fig. 4.8: A feature-based representation $A(T, \mathcal{F})$ (left) vs a dissimilarity representation $D(T, R)$ (right). Assume $T := \{t_1, \ldots, t_n\}$ is a set of training objects and $\mathcal{F} = \{f_1, \ldots, f_m\}$ are the features. An object t_i is then represented as a vector of its feature values $a(t_i, f_j)$ i.e. $A(t_i, \mathcal{F}) = [a(t_i, f_1), \ldots, a(t_i, f_m)]$. The feature f_j is represented as a vector $A(T, f_j)$. Hence, features correspond to dimensions in a (Euclidean) vector space, where objects become points. A dissimilarity representation describes the relations between objects, hence additionally, a collection of representatives $R := \{p_1, \ldots, p_r\}$ is needed. In the most simple case, $R{:=}T$ and for a quasimetric measure, the resulting $D(R, R)$ is a symmetric matrix with a zero diagonal. R might be a subset of T or a distinct set. An object t_i is represented by a vector of its dissimilarities $d(t_i, p_j)$ to the objects from R, i.e. $D(t_i, R) := [d(t_i, p_1), \ldots, d(t_i, p_r)]$. $D(T, p_j) := [d(t_1, p_j), \ldots, d(t_n, p_j)]^T$ refers to dissimilarities to a particular object p_j. Any entry in A is a feature value for a particular object, while any entry in D is a similarity value between two objects.

jects since they are based on sums of (weighted) independent contributions (referring only to some object properties). On the other hand, non-Euclidean or non-metric measures have become more popular, e.g. for measuring shape distances [93, 207] or others [115, 180, 206, 334].

For some reasons, such as measurement noise present in the sensory data there might be a necessity to improve the resulting dissimilarity measures. Noise can be reduced either in a pre-processing stage of the raw data or, if the measures are just given or directly result from an earlier analysis, by the use of (non-)linear transformations. Such transformations may be also applied to impose a more compact class description, e.g. by making large distances smaller, or (if required) by imposing particular characteristics of distances, e.g. a Euclidean behavior. Some of such transformations are described in section 3.1.

Since different proximity measures, as defined in feature spaces, between graphs and on the raw sensor data may reflect various aspects of data characteristics, as well as various kinds of expert knowledge, their combination might be beneficial. They can be considered either jointly or exclusively, or they might form a new proximity representation. The possibility of a combination makes a dissimilarity representation a more universal representation due to the increased flexibility. Now, a complex problem can be described by a number of DRs between their different aspects or characteristics; see also chapter 10. For instance, an article in a database can be represented (in intermediate stages) as a point in a feature space, where each feature corresponds to the frequency of the specified keyword, but also as a tree organization of a title, an introduction, body, conclusions and references, etc. Next, two different dissimilarity measures can be designed in the statistical and structural approaches, yielding two distinct DRs, which can be further combined. Combining proximity measures (or their transformed versions) [292] is closely related to the area of combining classifiers [217].

Another fundamental question refers to the learning paradigms, especially those which deal either with non-metric or non-Euclidean measures. Basically, they take place in spaces, already introduced in chapter 2. More precisely, they built on methods of linear algebra and functional analysis, as well as statistical learning [97, 191, 402], kernel methods [74, 352, 403] and approximate embeddings in pseudo-Euclidean spaces [103, 152, 295, 301], as presented in chapter 3. Further on, the usefulness of pretopological spaces, offering poorer axioms than Euclidean spaces, can also be studied. The

compactness hypothesis may serve as a basic demand for building pretopological spaces from more general neighborhood relations. The learning approaches are discussed in the next section.

Learned proximity representations. We realize that the developed framework for dissimilarity representations is only a first step in the direction of integrating both statistical and structural approaches, the problem of constructing an informative representation and proper learning methodologies. For dissimilarity representations, the measure itself is assumed to be given. To some extent, it can be optimized with respect to a set of objects, but rather in a limited way, like the specification of some parameters. The next step is to investigate how dissimilarity measures can be *learned* from a set of examples. For this purpose, a *learned* representation can be considered, primarily based on the structure present in real objects. Some proposals in this direction have been made by Goldfarb and his colleagues; see e.g. [154, 159–161].

Two possibilities can be now considered: to learn a relative representation or to learn a conceptual representation. The first focuses on defining a dissimilarity measure and a set of prototypes to which other objects will refer. Such a representation is used further for learning. The conceptual representation describes a dissimilarity of an object to a class. Such a dissimilarity is related to the costs (weights of transformations) of generating an object from a set of primitives (basic descriptors) in the context of other objects within a class, as well as objects outside this class. This is an attempt of a truly inductive way of learning [161], where not only the essential transformations and the weights are learned, but primitives as well. Such a formulation is close to the one-class classification [386, 390]. How to learn such measures is open for future research.

Another simpler approach is to combine the strengths of the structural and statistical frameworks on the level of a relative representation. Assume that one deals with objects that possess such a structure, such as spectra, time-signal, images or text documents. The first step is to define a small collection of fundamental structural detectors, yet general enough to be applicable in many problems, independent of a specific expert knowledge of the application. This means that such detectors are defined for the given measurement domain, e.g. spectra or images. The useful subpatterns should be then identified by the detectors when applied to the consecutive measurement values. The inter-relationships between the subpatterns should be captured in some relational intermediate representation (e.g. by a graph or by a string). These would be the basis for the matching process and the derivation of the final dissimilarity. The learning relies then on the learning of proper weights (contributions) assigned to the identified subpatterns such that the specified dissimilarity is optimal for the discrimination between the classes. The most simple example is the edit-distance between string descriptions of objects, however, more general approaches are needed to be developed. Note that one may also consider statistical feature extractors (such as wavelets or Gabor filters), which work on the consecutive measurements, to be the building blocks of the learned dissimilarity. How to learn such measures is open for research.

4.2.1 Dissimilarity representations: learning

Statistical learning approaches are adapted here for dissimilarity representations. The added value of a dissimilarity-based framework lies not directly in the following methodology, but in the representation itself. As we discussed in the previous section, a dissimilarity representation can include both the statistical and structural properties of data. Hence, instead of a single representation of a problem, one may also consider either a complex representation, as one built from many dissimilarity representations or a hybrid representation, where different aspects of the data are described in various ways such as by features, dissimilarities, and inference rules. Then, one needs to face the task of combining the expertise [266, 267]; see also section 10.

We will concentrate here on the classification task. Given a training set of K classes, a classifier

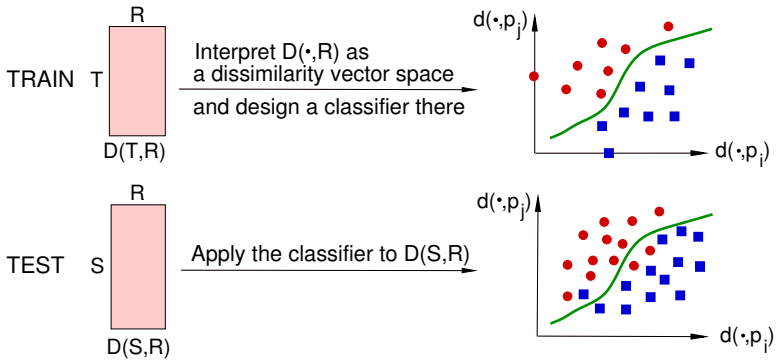


Fig. 4.9: An illustration of a classification process in dissimilarity spaces.

tries to model a functional dependence between the data representation and the class indicators (labels) such that a new object can be assigned to a specific class. The goal is the minimization of (the cost of) misclassification such that novel examples are possibly correctly labeled. The problem to be faced in establishing classification methodologies for dissimilarities is that the measures used in practice are often non-Euclidean or even non-metric. Nevertheless, they may perform well and it remains of practical interest to study their properties fundamentally. Since DRs encode the information on objects dissimilarities in a numerical way, the nature of learning is unavoidably numerical, which leads to the use of spaces. In general, we can distinguish three main approaches to dissimilarity representations, where all of them are interpreted in the context of some spaces.

Assume a dissimilarity representation $D(T, R)$, where R is a representation set and T is a training set. The measure d is general, our basic requirements are only the nonnegativity and reflexivity, Def.2.30. In the first *pretopological approach*, making use (directly or not) of pretopological spaces, the dissimilarities between the objects are interpreted directly. This means that a dissimilarity representation describes an abstract space[6], where the neighborhoods or closure operators play a key role; see also section 2.1. These are defined based on the dissimilarities to the objects from R. An example classifier is the k-nearest neighbor rule (k-NN).

The second *dissimilarity space approach* addresses a dissimilarity representation as a data-dependent mapping specified by the representation set R. A mapping $D(z, R) : \mathcal{X} \to \mathbb{R}^n$ is defined as $D(z, R) = [d(z, p_1)\, d(z, p_2)\, \ldots\, d(z, p_n)]$. Note that $\mathcal{X}$ expresses either objects themselves, or an original or intermediate feature space of objects, which might not be given explicitly. The dimensionality of such a space is controlled by the cardinality of R. Using this formulation, classifiers can be constructed directly on the DRs, as in a dissimilarity space, where each dimension corresponds to a dissimilarity to a representative, say $d(\cdot, p_i)$. Since dissimilarities are nonnegative, all the data are mapped as points to a nonnegative part of the complete space. Many traditional classifiers can be applied there [290, 293, 301]. See also Fig. 4.9 and section 4.4.

The third *embedding approach* relies on a DR, where $R \subseteq T$. First, a spatial representation of the symmetric $D(R, R)$ is found (a space V where the objects are mapped as points such that their distances reflect the actual dissimilarities; see section 3.4) and then, the remaining objects $T \backslash R$, if exist, are projected there. Note that $D(R, R)$ should be a symmetric matrix. Next, a classifier, e.g. a linear classifier built in feature vector spaces, is trained in the determined space. New data S, described as $D(S, R)$, are first projected to the space V and then the classifier is applied. See also Fig. 4.10. Further details can be found in section 4.5.

In summary, to construct a classifier for dissimilarity representations, the training set T of N objects

[6] This is an abstract space in the sense that it is not explicitly given. It is defined by a set of available objects, performed measurements and the number of factors, such as camera positions or lighting conditions, playing role in the measurement process. So, this abstract space might be seen as a measurement space.

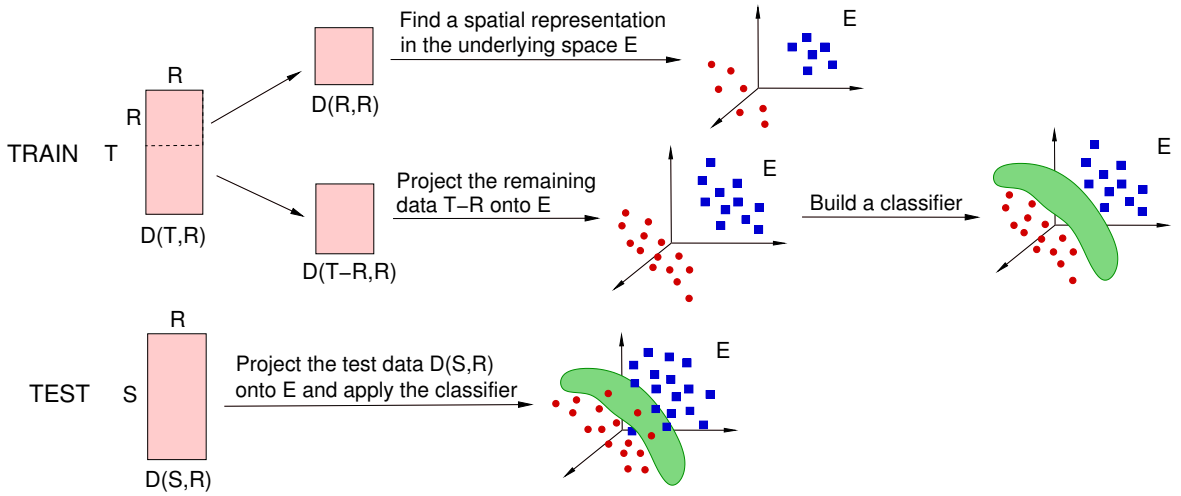


Fig. 4.10: An illustration of a classification process in the embedding approach.

and the representation set R [99] of n objects are used. R is a set of prototypes, possibly covering all the present classes. R is usually considered to be a subset of T ($R \subseteq T$), although R and T might be disjunct for the first two approaches. In the former case, R might be chosen from T randomly or in a systematic way, starting from $R := T$ and the complete representation $D(T,T)$. For instance, n objects can be chosen such that the minimum distance between any of them is maximized. Another possibility is based on a greedy approach. Starting from a randomly chosen object, in an iterative procedure, an object is added which is the most dissimilar to all objects already chosen. It might be done globally or for each class separately. In the case when the most dissimilar objects are chosen, they are likely to be outliers or positioned between the classes. Some of the selection methods will be discussed in chapter 9. In the learning process, a classifier is constructed by making use of the $N \times n$ $D(T,R)$, relating all training objects to all the prototypes. The information on a set T_s of s new objects is provided by their dissimilarities to R, i.e. an $s \times n$ matrix $D(T_s, R)$.

4.3 Classification in generalized topological spaces

Let X be either a finite set or a vector space. Consider a generalized metric space (X, ρ) with a dissimilarity measure $\rho\colon X \times X \to \mathbb{R}^0_+$ such that $\rho(x,x) = 0$. Let $B_\delta(x) := \{y \in X : \rho(x,y) < \delta\}$ be a δ-ball for $\delta > 0$. For each $x \in X$ define its minimal neighborhood as $B_{\delta_{nn}}(x) := \{y \in X : \rho(x,y) < \delta_{nn}\}$, where $\delta_{nn} = 1.00001 \cdot \rho(x, nn(x))$ and $\rho(x, nn(x))$ is the dissimilarity of x to its nearest neighbor $nn(x)$. By the reflexivity property of ρ, $x \in B_{\delta_{nn}}(x)$. A growth function gr is now defined on the power set $\mathcal{P}(X)$ as a generalized closure operator such that for every $A \subseteq X$

(1) $\mathrm{gr}(\emptyset) = \emptyset$.

(2) $\mathrm{gr}(x) = x \cup B_{\delta_{nn}}(x) = B_{\delta_{nn}}(x)$.

(3) $\mathrm{gr}(A) = \bigcup_{x \in A} \mathrm{gr}(x)$, where $A \in X$.

It is straightforward to check that this growth function fulfills the axioms (1) – (4) of Def. 2.9, hence (X, gr) is a pretopological space. Such a closure operator describes the δ_{nn}-neighbors pretopology. Imagine now that δ_{nn} does not depend on x, hence $\delta_{nn} := \varepsilon > 0$. If ε is chosen as $\min_{x \in X} \rho(x, nn(x))$, then the growth of x becomes $\mathrm{gr}(x) = B_\varepsilon(x)$.

If X is a Hilbert vector space, then due to the property of convex neighborhoods for a metric distance ρ, $\mathrm{gr}^2(x) := \mathrm{gr}(\mathrm{gr}(x)) = B_{2\varepsilon}(x)$ and, more generally, $\mathrm{gr}^m(x) = B_{m\varepsilon}(x)$ for $m \in \mathbb{N}$. For a non-metric measure ρ, this is not true, in general. Still it may happen that $B_{m\varepsilon}(x) \subseteq \mathrm{gr}^m(x)$ from below[7]. See

[7] We will show that $B_{2\varepsilon}(x) = \mathrm{gr}^2(x)$ for a metric dissimilarity ρ. Note that $B_{2\varepsilon}(x) = \{z : \rho(z,x) < 2\varepsilon\}$ and $\mathrm{gr}^2(x) = \bigcup_{y \in B_\varepsilon(x)} B_\varepsilon(y) = \{(z,y) : \rho(z,y) < \varepsilon \wedge \rho(y,x) < \varepsilon\}$.

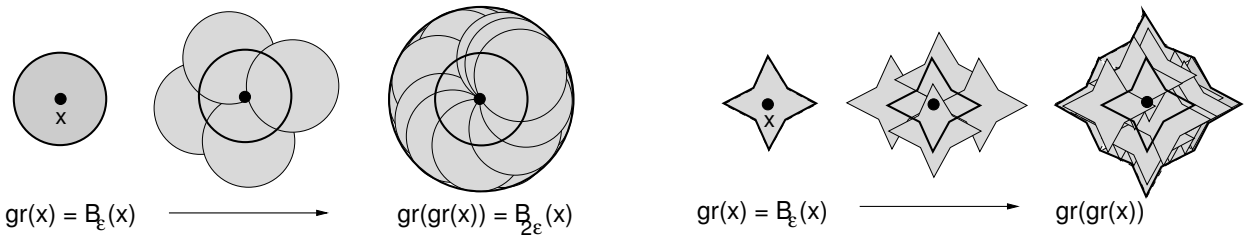


Fig. 4.11: Assume a vector space X with the additional dissimilarity ρ (it might be for instance a measurement space). Example growth operators $\mathrm{gr}(x) = B_\varepsilon(x)$ with a fixed $\varepsilon > 0$ are shown, when ρ is a metric distance (left) or when ρ is a non-metric dissimilarity (right). In the non-metric case, $\mathrm{gr}(\mathrm{gr}(x))$ is not necessarily identical to $B_{2\varepsilon}(x)$, as in this metric case.

also Fig. 4.11 for an illustration.

Alternatively, one can define the k-nearest neighbor pretopology, where the growth of x is given as $\mathrm{gr}(x) = \{y : y \text{ belongs to a set of } k\text{-th nearest neighbors of } x\}$, satisfying $\mathrm{gr}(\emptyset) = \emptyset$ and $\mathrm{gr}(A) = \bigcup_{x \in A} \mathrm{gr}(x)$. It is then straightforward to check that gr fulfills the axioms of pretopology, Def. 2.9.

Another possibility is to use neighborhoods to define the neighborhood basis at x. Let $B_\varepsilon(x) := \{y \in X : \rho(x,y) < \varepsilon\}$ be an ε-ball for a positive ε. Then, the neighborhood basis is defined as $\mathcal{N}_B(x) = \{B_\varepsilon(x) : \varepsilon = 1.00001 \cdot \rho(x, nn_k(x)),\ k = 1, 2, \ldots\}$, where $nn_k(x)$ is the k-th neighbor of x. Consequently, $(X, \mathcal{N}_B)$ describes a pretopological space.

Consider now a training set T and a dissimilarity representation $D(T,R)$. A generalized closure (growth) operator or neighborhood basis can be defined for every class c_i based on $D(T^i, R^i)$, where $T_i \subset T$ and $R_i \subset R$ correspond to the objects from c_i. So, B_{ε_i} is the neighborhood basis for the class c_i. An unknown object is assigned to the class c_k if it belongs to a generalized closure or a neighborhood of one or more objects from the class c_k only. If no single class exists, then the sets $B_{m\varepsilon_i}$ for $m \in \mathbb{N}$ can be used instead as an approximation of the successive growth by a repetitive use of the generalized closure (in Hilbert vector space with a growth function defined by a metric distance, $B_{m\varepsilon}(x) \subseteq \mathrm{gr}^m(x)$ holds). If an object belongs to the intersection of neighborhoods (or closures) of two (or more) classes, then the final decision should be made by looking at the majority of objects from a particular class within the neighborhoods.

It means that the decision rules built on the dissimilarities directly can be interpreted as classifiers in pretopological spaces. Examples are variants of the nearest neighbor rules. A classifier based on the repetitive closure operators in pretopological spaces (hence based on growing neighborhoods) was also discussed in [134, 240], which we discovered just in the moment of writing this thesis.

Nearest neighbor (NN) rule. A straightforward approach to dissimilarities leads to the nearest neighbor rule [71, 138] or, more generally, to the instance-based learning [2]. In its simplest form, the 1-NN rule assigns a new object to the class of its nearest neighbor from the representation set R by finding minimal in the rows of $D(T_s, R)$, where T_s is a test set. (Originally, $R := T$ is assumed.) The k-NN rule is based on majority voting, i.e. an unknown object becomes a member of the class the most frequently occurring among the k nearest neighbors. Usually, k is assumed to be odd to avoid ties (for two-class problems). Note that when k is fixed, *no* training is involved. Traditionally, the k-NN rule is applied for the data represented as vectors in a feature space, often based on the Euclidean or city block metrics (which means that indirectly the corresponding dissimilarity repre-

$\Rightarrow$ Let $z \in B_\varepsilon(x) \Leftrightarrow \rho(z,x) < 2\varepsilon$. In a Hilbert vector space with the metric ρ, there exist a unique middle point such that $\rho(z,x) = \rho(z,y) + \rho(y,x)$. Hence, $\rho(z,y) < \varepsilon$ and $\rho(y,x) < \varepsilon$. It follows that $z \in \mathrm{gr}^2(x)$ and consequently, $B_{2\varepsilon}(x) \subseteq \mathrm{gr}^2(x)$.

$\Leftarrow$ Let $z \in \mathrm{gr}^2(x) \Leftrightarrow \rho(z,y) < \varepsilon \ \wedge \ \rho(y,x) < \varepsilon$. Since ρ is nonnegative, then $\rho(z,y) + \rho(y,x) < 2\varepsilon$. By the triangle inequality, $\rho(x,z) < \rho(z,y) + \rho(y,x)$. Hence, $\rho(x,z) \leq \varepsilon$. It follows that $z \in B_{2\varepsilon}(x)$ and, consequently, $\mathrm{gr}^2(x) \subseteq B_{2\varepsilon}(x)$. .

sentation is used). Yet, its principles can be extended to other (non-metric) dissimilarity measures, obeying the compactness hypothesis; see also section 4.1.

The k-NN classifier is attractive, since it is simple, intuitively appealing and no prior knowledge of the data distributions is required. It can estimate complex boundaries locally and differently for each new instance (hence its adaptations can be seen as an example of transductive learning). Moreover, it is known [87, 97] that for the k-NN rule f_{kNN}, the empirical risk $\mathcal{E}_{emp}(f_{kNN})$ converges uniformly to the actual risk $\mathcal{E}_{kNN}$ with increasing n, see also equation (4.4), such that $\mathcal{E}(f_*) \leq \ldots \leq \mathcal{E}_{2l+1NN} \leq \mathcal{E}_{2l-1NN} \leq \ldots \mathcal{E}_{3NN} \leq \mathcal{E}_{1NN} \leq \mathcal{E}(f_*)(2 - \frac{K}{K-1}\mathcal{E}(f_*))$, where $\mathcal{E}(f_*)$ is the Bayes error and K is the number of classes. This means that the k-NN rule is asymptotically at most twice as bad as the Bayes rule; see also the book of Devroye et al. [87] for other bounds. In practice, when one deals with finite sample sizes, the asymptotic inequalities will not hold. The k-NN rule is expected to perform well, provided that the domain of a problem, hence the data are well sampled. In cases, where at least one of the classes is undersampled or badly sampled, the k-NN rule deteriorates.

The k-NN rule can also be interpreted as the one which locally tries to estimate the posterior probabilities. These estimates rely in fact on a neighborhood determined by the k-furthest neighbor. For small k, the nearest neighbors might often lie further away due to the data sparseness or the estimates might be poor due to noisy examples. Increasing k allows one to reduce the noise influence, however, the nearest neighbors with large dissimilarities in the voting scheme may lead to an unnecessary error. Therefore, a weighted voting [87] might be an option, where the neighbor contributions are weighted accordingly to their dissimilarities to a particular object.

Edited and condensed nearest neighbor rules. Despite the simplicity and good performance of the k-NN rule, the criticism points at both the space requirement to store the entire training set and the computational expense of computing dissimilarity to all training examples. Consequently, there has been an interest in *condensing* the training set in order to reduce its size; see e.g. [77, 187, 422]. In our terminology this is equivalent to a selection of a proper representation set R out of the training set T, hence it might be called a prototype selection, as well. Also *editing* [86] is considered, which goal is to increase the accuracy of the k-NN predictions, given noise in the training data. A basic editing algorithm removes noisy instances as well as close border cases, leaving smoother decision boundaries. It also retains all 'internal' points; i.e. it does not reduce the number of objects as much as most other reduction algorithms. More on condensing can be found in in section 9.2.

Many variants of the NN-rule, taking into account the local structure of the data or weighting the neighbor contributions appropriately, have been invented or adopted for feature based representations, see section 5.5 for a brief information. The question on how such measures should be constructed is beyond the scope of this dissertation.

4.4 Classification in dissimilarity spaces

The novelty of our approach relies on interpreting $D(T, R)$ as a representation of a vector space, called a *dissimilarity space*, where each dimension corresponds to the dissimilarity to an object from the set R. $D(z, R)$ defines then a vector[8] consisting of n dissimilarities found between the object z and all the objects from R. see also Def. 4.1 and Fig. 4.9. Therefore, $D(\cdot, R)$ is seen as a data-dependent mapping onto an n-dimensional dissimilarity space. The advantage of such a representation is that any traditional classifier operating in vector spaces can be used.

[8] Here, $D(z, R)$ is a row vector, however, to avoid extra complications, $D(z, R)$ is silently assumed to be a column vector, if necessary. Hence $D(z, R) := D(z, R)^T$.

If the dissimilarity measure d is a metric, then all vectors $D(z,R)$ lie in an n-dimensional prism, bounded from below by a hyperplane on which the objects from R are and which is bounded from above in case of bounded dissimilarities. Consider a 2D representation $D(z,R)$, where $R=[p_i,p_j]$. For brevity, denote that $d_{ij}:=d(p_i,p_j)$ and $x:=d(z,p_i)$, and $y:=d(z,p_j)$ for an object z. Then, for a metric d, the following triangle inequalities should hold: $x+y\geq d_{ij}$, $d_{ij}+x\geq y$ and $d_{ij}+y\geq x$. Depending on x and y, a prism is formed as denoted in Fig. 4.12. Note that in higher-dimensional spaces, the prism is asymmetric and the vertices of its base do not lie on the axes (e.g. in a 3D space the vertices lie in the xy, yz and xz planes). In principle, z may be placed anywhere in a dissimilarity space $D(\cdot,R)$ only if the triangle inequality is completely violated. This is, however, not possible from the practical point of view, because then the compactness hypothesis will not be fulfilled. Consequently, this would mean that d has lost its discriminating properties of being (relatively) small for similar objects. Therefore, the measure d, if not metric, it has to be sufficiently close to a metric and, thereby, $D(z,R)$ will still lie either in the prism or in its relatively close neighborhood[9]. See Fig. 4.13 to get some intuition.

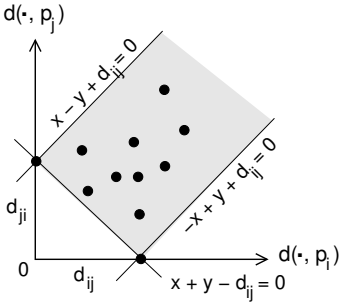


Fig. 4.12: Metric 2D dissimilarity space.

A justification for the construction of classifiers in dissimilarity spaces is as follows. The property that dissimilarities should be small for similar objects, i.e. belonging to the same class, and large for distinct objects, gives a possibility for a discrimination. Thereby, $D(\cdot,p_i)$ defined by the dissimilarities to the representative p_i can be interpreted as a 'feature'. If p_i is a characteristic object of a particular class, then the discrimination power of $D(\cdot,p_i)$ can be large. If p_i is a drastically atypical object of its class, then $D(\cdot,p_i)$ may not be informative. However, the strength lies in using all dissimilarity values $D(\cdot,R)$. Another reasoning relies on the fact that if the objects x and y are similar in reality and the dissimilarity $d(x,y)$ is small, then for some other objects z, the dissimilarities $d(x,z)$ and $d(y,z)$ might not express similar values if the measure is non-metric. However, is the dissimilarities of x and y to a given set of prototypes R are inspected, one can expect that, although the individual values may differ, in their entirety, the vectors $D(x,R)$ and $D(z,R)$ are correlated. If so, then the representations $D(x,R)$ and $D(z,R)$ are close in a dissimilarity space. Consequently, the dissimilarity space approach should be useful for non-metric measures.

An important point can be made for metric distances. A max-norm dissimilarity space is in fact the result of an embedding of a metric distance representation D, as presented in Lemma 3.7. In other words, this means that the max-norm distance in a dissimilarity space $D(\cdot,R)$ $d_\infty(d(p_i,R),d(p_j,R))$ *is equal to the original distance* $d(p_i,p_j)$. *This justifies the construction of a dissimilarity space for a metric distance.*

One may wonder what the added value of such a representation over a feature-based representation is, if the traditional classifiers designed for vectors spaces may be applied in the end. First of all, the strength of a dissimilarity representation relies on its flexibility to encode both statistical and structural characteristics of the data, so it is a representation, where object properties can be captured more adequately. Hence, more emphasis and knowledge is put to a class of similar objects. Since a DR is a numerical description, its interpretation will take place in some space. The choice of a vector space $D(\cdot,R)$ is in agreement with its mathematical concept in the following way. The dimensions of such a space are now dissimilarities to the prototypes which are derived according to a specified measure. Hence, they convey homogeneous type of information. In that sense, the dimensions are equally important. This is not valid for a general feature-based representation, where

[9] This is not always true. If one considers a power transformation of e.g. a metric, which as a monotonic transformation preserves the order of dissimilarities, then a large deviation from the triangle inequality can be expected. For instance, this happens for $d:=d_E^1 0$ for d_E taking values in $[0,5]$.

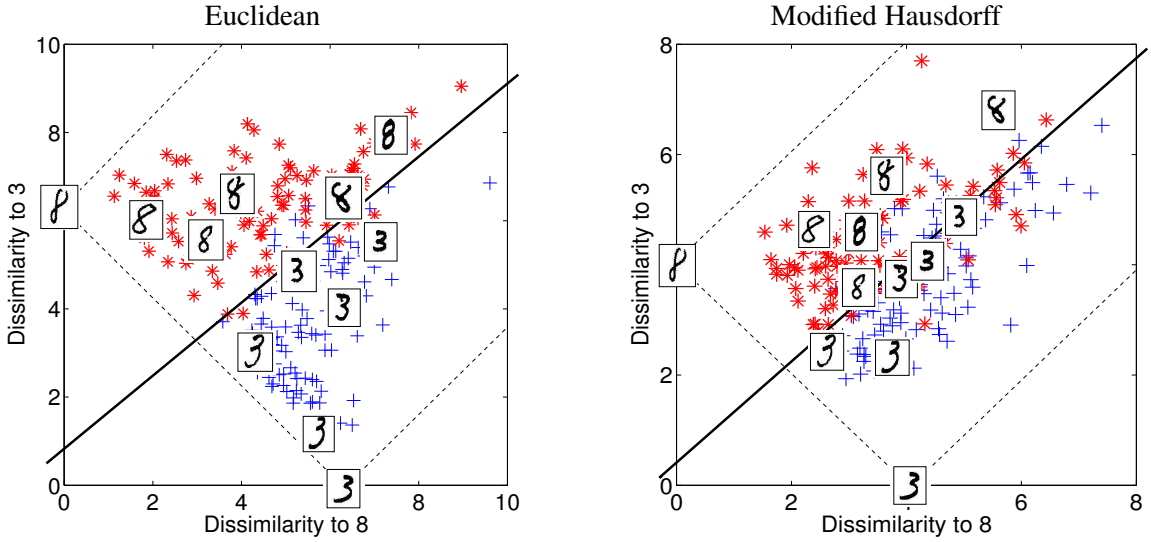


Fig. 4.13: Examples of 2D dissimilarity spaces and linear classifiers for a subset of handwritten digits 3 and 8. Two dissimilarity representations $D(T,R)$ are shown based on the Euclidean distance (metric) between blurred images and the modified Hausdorff distance (non-metric) between image contours; see also section 5.4. R is randomly chosen and consist of two examples, one for each digit.

features have different character and range, as for instance weight or length. Another advantage of a DR is that since a dissimilarity measure already possibly encodes the object structure and/or other characteristics, the designed classifiers might be chosen as to be simple, e.g. linear models.

Defining a well-discriminating dissimilarity measure for a non-trivial learning problem is difficult. Designing such a measure is equivalent to defining good features in a traditional feature-based classification problem. If a good measure can be found and a training set is representative, then the k-NN rule is expected to perform well. The decision of the k-NN is based on local neighborhoods and it is, in general, sensitive to noise. It means that k nearest neighbors found might not be the best representatives for making a decision to which class an object should be assigned. In cases of small or non-representative training sets, a better generalization can be achieved by a classifier built in a dissimilarity space.

For instance, a linear classifier in a dissimilarity space is a weighted linear combination of dissimilarities between an object and the representation examples. The weights are optimized on the training set and large weights (in magnitude) emphasize objects which essentially influence the final decision. By doing this, a more global classifier can be built, by which its sensitivity to noisy representative examples is reduced. Our experience confirms that a linear or quadratic classifier can often generalize better than the k-NN rule, especially for a small representation set R; see also [291].

4.4.1 Classifiers

$D(x,R)$ is now considered as an $N \times 1$ vector. $D(T,R)$ describes an $N \times n$ dissimilarity matrix. A linear classifier built in a dissimilarity space $D(\cdot,R)$ is, in general, expressed as

$$f(D(x,R)) = \sum_{j=1}^{n} w_j\, d(x,p_j) + w_0 = \mathbf{w}^T D(x,R) + w_0. \tag{4.7}$$

In fact, f written above is just a linear function. In the training process, it is designed to be the boundary between the classes. The classifier is a function returning the class assignments. Usually, one assumes that the equation $f(D(x,R)) = 0$ defines a classifier, hence for a two-class problem, the sign of $f(D(z,R))$ determines which class z will belong to. For simplicity, we will only discuss the form of f in our further considerations.

Classifiers originally defined in feature vector spaces can be applied to dissimilarity representations. Some of them are briefly described here; see [97, 138, 191, 317] for a more elaborate introduction. In all the descriptions below, we will distinguish K classes, usually two, since each multi-class classification problem can be decomposed into a number of two-class problems. The k-th class with cardinality n_k is denoted by c_k, $k = 1, \ldots, K$, and its prior probability by $p(c_k)$. The class mean vectors found in the space $D(\cdot, R)$ are denoted by $\mathbf{m}_i$ and the overall mean is given by $\mathbf{m}$. When classifiers are described in features spaces, for simplicity, the training pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i$ is a vector and $y_i = \{-1, 1\}$ is a label, are considered.

Normal density based linear/quadratic classifiers (NLC/NQC). Most of the commonly-used dissimilarity measures, like Euclidean, city block or Hamming distance, are based on sums of differences between measurements. The central limit theorem states that the sum of iid random variables tends to be normally distributed in the limit, provided that none of the variances of the sum's components dominates (otherwise, the distribution is χ^2). An approximation can already be good for a relatively small number of variables, e.g. such as 10. Practice shows that summation-based distances which are built from many components of similar variances are often approximately normally distributed (in fact, this is a clipped distribution due to the nonnegativity of dissimilarities). This suggests that the (regularized) linear/quadratic normal density based classifiers [317], which assume normal class distributions, should be of use in dissimilarity spaces. For a two-class problem, the NLC based on the set R is given by:

$$f\left(D(x,R)\right) = [D(x,R) - \frac{1}{2}(\mathbf{m}_1 + \mathbf{m}_2)]^T \, C^{-1} \, (\mathbf{m}_1 - \mathbf{m}_2) + \log \frac{p(c_1)}{p(c_2)}, \tag{4.8}$$

and the NQC is given by

$$f(D(x,R)) = \sum_{i=1}^{2} (-1)^i \, (D(x,R) - \mathbf{m}_i)^T \, C_i^{-1} (D(x,R) - \mathbf{m}_i) + 2 \log \frac{p(c_1)}{p(c_2)} + \log \frac{|C_1|}{|C_2|}, \tag{4.9}$$

where C_1 and C_2 are the estimated class covariance matrices and $C := (C_1 + C_2)/2$ is the sample covariance matrix, determined in a dissimilarity space. The value of $(D(x,R) - \mathbf{m}_i)^T \, C_i^{-1} (D(x,R) - \mathbf{m}_i)$ is the square Mahalanobis distance between $D(x,R)$ and the class mean $\mathbf{m}_i$; see a paragraph on quantitative data in section 5.1.

When C (C_1 or C_2) becomes singular, its inverse cannot be computed. A solution is offered by using a regularized version instead [317] as $C_{\text{reg}} = (1 - \lambda) \, C + \lambda \, I$, where I is the identity matrix. Since it is hard to choose a proper λ, in our implementations we make use of the following regularization: $C_{reg}^{\lambda} = (1 - 2\lambda) \, C + \lambda \, \text{diag}(C) + \frac{\lambda}{n} \text{trace}(C) \, I$ ($n = |R|$). The regularization term is now expressed relatively to the variances, so it can be determined more easily. In practice, λ equals 0.05, 0.01 or less. The resulting regularized classifiers are called appropriately and denoted by the RNLC and RNQC[8].

Strongly regularized quadratic classifiers (SRQC). This classifier is similar to the NQC defined above, but with a difference in the used regularization which is expressed by diminishing the influence of covariances with respect to variances. This means that each class covariance matrix is estimated as $C_i^{\kappa} := (1 - \kappa) \, C_i + \kappa \, p(c_i) \, \text{diag}\,(C_i)$, where $\kappa \in [0, 1]$. If $\kappa = 0$, then the classifier reduces to the NQC, while if $\kappa = 1$, then the classifier becomes the scaled nearest mean linear classifier [138, 367]. So, the variation of κ corresponds to a change between these two extremes. We often use $\kappa = 0.2$ or 0.8, where we become closer to one of the extreme cases.

[8] Although the NLC and NQC classifiers rely on the ERM principle, their regularized equivalents, the RNLC and RQNC, as well as the SRQC are based on the regularization of the covariance matrices. This, in turn, allows for finding their bounded inverses, which implies that the classifier's weights are bounded as well. So, this is an indirect attempt for the use of the regularization principle; see section 4.1.3.2.

Fisher and pseudo-Fisher linear discriminants (FLD and PFLD). The Fisher linear discriminant is a linear decision rule obtained by maximizing the *Fisher criterion*, i.e. $\max_{\mathbf{w}} \frac{\mathbf{w}^T C_B \mathbf{w}}{\mathbf{w}^T C_W \mathbf{w}}$, where $C_B = \sum_{k=1}^{K} n_k (\mathbf{m}_k - \mathbf{m})(\mathbf{m}_k - \mathbf{m})^T$ is the between-class scatter and C_W is the within-class scatter (sum of the class covariance matrices) [97, 138]. It is known that for a two-class problem with equally probable classes, the FLD is equivalent to the NLC. For a dissimilarity representation $D(T, R)$, the FLD [138] is constructed as:

$$f(D(x,R)) = (\mathbf{m}_1 - \mathbf{m}_2)^T C_W^{-1} \, D(x,R) - \frac{1}{2} (\mathbf{m}_1 + \mathbf{m}_2)^T \, C_W^{-1} (\mathbf{m}_1 - \mathbf{m}_2) + \log \frac{p(c_1)}{p(c_2)}, \qquad (4.10)$$

If the estimated covariance matrix C_W becomes singular, a pseudo-inverse operation is proposed instead, yielding the pseudo-Fisher linear discriminant [314]. The pseudo-inverse relies on the singular value decomposition of the matrix C_W. It is computed as the inverse of C_W, but in the subspace spanned by the eigenvectors corresponding to m largest non-zero eigenvalues. The classifier is found in this subspace, being orthogonal to it in the remaining $(n-m)$ directions. The PFLD is reached in the limit of the RNLC if the regularization λ goes to zero [314].

Support vector classifier (SVM). Let n training pairs $\{\mathbf{x}_i, y_i\}_{i=1}^n$ be given in a Euclidean (Hilbert) space. Each point $\mathbf{x}_i$ belongs to one of two classes as described by the corresponding label $y_i \in \{-1, 1\}$. The support vector machine is the hyperplane $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + w_0$ maximizing the margin $2/||\mathbf{w}||^2$ [52, 403] between two separable classes (or, alternatively, minimizing the norm $||\mathbf{w}||^2$). In case of an overlap, a soft-margin hyperplane is introduced, which handles the misclassified objects. The linear SVM is expressed as $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \, y_i \, \langle \mathbf{x}, \mathbf{x}_i \rangle + \alpha_0$, where $\langle \mathbf{x}, \mathbf{x}_i \rangle := \mathbf{x}^T \mathbf{x}_i$ is the dot product operation and α_i are nonnegative values determined by maximizing the (soft) margin. Note also that $\mathbf{w} = \sum_{i=1}^n \alpha_i \, y_i \, \mathbf{x}_i$. Since many α_i appear to be zero, only the objects corresponding to non-zero weights, the *support vectors* (SV), contribute to the classifier.

The SVM is an elegant implementation of the SRM principle in practice (hence its importance), section 4.1.3.2, by combining the theory of the largest margin with the control over the VC dimension of a class of linear functions. We will briefly recapitulate this fact here, see e.g. [345, 402, 403] for details. Assume separable classes. Let $\mathcal{G}^h$ be a subclass of all hyperplanes in a space $\mathbb{R}^r$, i.e. $\mathcal{G}^h := \{g : \, g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + w_0\}$. The VC dimension of $\mathcal{G}^h$ is $h_{vc} = r + 1$, which means that the maximal number of arbitrary labeled points in $\mathbb{R}^r$ separated by hyperplanes into two classes is $n+1$. Since h_{vc} is finite, the Vapnik's bound (4.5) holds. Still, we need to introduce the nested structure of function classes for the SRM principle to be true. This turns out to be possible by bounding the linear functions in $\mathcal{G}^h$. Denote $\mathcal{G}^h_\lambda := \{g : \, g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + w_0, \; ||\mathbf{w}||_2 \leq \lambda\}$. Clearly, if $\lambda_1 < \lambda_2$, then $\mathcal{G}^h_{\lambda_1} \subseteq \mathcal{G}^h_{\lambda_2}$. To ensure that the same inequality follows for the corresponding VC dimensions, one should require that the hyperplanes are selected as the largest margin hyperplanes for a given labeled set of data points. It was then shown [345] that the VC dimension for $\mathcal{G}^h_\lambda$ can be then effectively bounded as $h_{vc} \leq \min\{\lambda^2 R^2 + 1, \; r+1\}$ for $||\mathbf{w}||_2 \leq \lambda$, where R is the radius of the smallest sphere enclosing the data points.

An extension to a nonlinear decision function is obtained by a mapping Φ of the input data to a high-dimensional Hilbert space and finding a linear classifier there. This is expressed as [403]: $f(\mathbf{x}) = \sum_{i=1}^n \alpha_i \, y_i \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle + \alpha_0$. The dot product can then be replaced by its generalized version $K(\mathbf{x}, \mathbf{x}_i) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i) \rangle$, a reproducing kernel K; see also Def. 2.63 and section 2.3.1. Since in a high-dimensional space, the SVM is based on inner products of vectors and support vectors only, the kernel operator can be defined explicitly instead of the map Φ. The kernel can be any symmetric and positive definite (pd) function[9] [403], see also Theorem 2.66. Hence, in a general formulation,

[9] Actually, any conditionally positive definite (cpd) kernel $\tilde{K}$ can be used. (Note also that any pd kernel is also

where the (non-)linearity of K determines the nonlinearity of f, the SVM is defined as

$$f(\mathbf{x}) = \sum_{\alpha_i > 0} \alpha_i \, y_i \, K(\mathbf{x}, \mathbf{x}_i) + \alpha_0. \tag{4.11}$$

According to definitions from section 2.3.1, the kernel K is a reproducing kernel, hence it defines a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ on the functions $h(\mathbf{x}) = \sum_i \alpha_i \, y_i \, K(\mathbf{x}, \mathbf{x}_i)$.

For linearly non-separable classes, nonnegative slack variables ξ_i are introduced, accounting for classification errors. The soft margin SVM [52, 403] is found as the primal solution of the quadratic programming (QP) procedure:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \gamma \sum_{i=1}^{n} \xi_i \\ \text{s.t.} & y_i \, (\mathbf{w}^T \Phi(\mathbf{x}_i) + w_0) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \end{array} \tag{4.12}$$

The term $\sum_{i=1}^{n} \xi_i$ is an upper bound on the misclassification of the training samples and γ can be regarded as a regularization parameter, a trade-off between the number of errors and the width of the margin. For an $n \times n$ kernel matrix K, $K_{ij} = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$, the dual formulation becomes:

$$\begin{array}{ll} \text{Maximize} & -\frac{1}{2} \, \boldsymbol{\alpha}^T \text{diag}(\mathbf{y}) \, K \, \text{diag}(\mathbf{y}) \, \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1} \\ \text{s.t.} & \boldsymbol{\alpha}^T \mathbf{y} = 0 \\ & 0 \leq \alpha_i \leq \gamma, \qquad i = 1, 2, \ldots, n \end{array} \tag{4.13}$$

Note that the norm of h in $\mathcal{H}_K$ is computed as $||h||_{\mathcal{H}_K} = \langle \sum_i \alpha_i \, y_i \, K(\mathbf{x}, \mathbf{x}_i), \sum_j \alpha_j \, y_j \, K(\mathbf{x}, \mathbf{x}_j) \rangle_{\mathcal{H}_K} = \boldsymbol{\alpha}^T \text{diag}(\mathbf{y}) \, K \, \text{diag}(\mathbf{y}) \, \boldsymbol{\alpha}$ due to the reproducing property of the kernel; see also section 2.3.1. Minimizing the latter quantity, which is equivalent to maximizing $-||h||_{\mathcal{H}_K}$ in the dual formulation above, corresponds then to bounding a class of hyperplanes h in the regularization principle, mentioned in section 4.1.3.2.

To introduce an SVM in a dissimilarity space, one needs to build it on $D(T, R)$. It is then a straightforward implementation. In the most simple, linear case, it leads to the kernel K consisting of the elements $K_{ij} = \langle D(x_i, R), D(x_j, R) \rangle$. Therefore, in the formulation of a linear SVM, in the training stage, K becomes $K = D \, D^T$, which is pd by construction. Also other positive definite kernels can be used as well. In such a case, however, a sparse solution, provided by the method, is obtained in the complete dissimilarity space $D(\cdot, R)$. It means that for the evaluation of new objects, still the dissimilarities to *all* representative objects from R should be computed, since our SVM is in the form of (4.7).

Linear programming machines (LP). Given a properly defined objective function and constraints, a separating hyperplane can be obtained by solving a linear programming (LP) task, making the optimization problem easier. Assume N training pairs $(D(x_i, R), y_i)$, $i = 1, \ldots, N$, $y_i = \{1, -1\}$, and a two-class problem, with the classes c_1 and c_2 of cardinalities n_1 and n_2, respectively $N = n_1 + n_2$. Let f be a separating hyperplane built for the representation set R, i.e. $f(D(x, R)) = \mathbf{w}^T D(x, R) + w_0$. Then, a simple optimization problem minimizing the number of misclassification errors ξ_j can be

a cpd kernel. This is trivial, since for a pd real kernel $\mathbf{z}^T K \mathbf{z} > 0$ for any $\mathbf{z}$, hence also for $\mathbf{z}$ such that $\mathbf{z}^T \mathbf{1} = 0$, which defines a cpd kernel. The vice versa formulation does not hold.) Let $K = \frac{1}{2} \, (I - \mathbf{1}\mathbf{s}^T) \, \tilde{K} \, (I - \mathbf{s}\mathbf{1}^T)$, where $\mathbf{s}^T \mathbf{1} = 1$. It is known [346] that K is pd iff $\tilde{K}$ is cpd; see also Theorems 3.31, 3.32 and 3.38. It follows from the above theorems that an $n \times n$ matrix $\tilde{K}$ is a matrix of negative square Euclidean distances, i.e. $\tilde{K} := -D^{*2}$. Then $\tilde{K} = 2 \, K - \text{diag}\,(K) \, \mathbf{1}^T - \mathbf{1} \, \text{diag}\,(K)^T$. By the use of equivalent algebraic transformations one can check that the function $-\frac{1}{2} \, \boldsymbol{\alpha}^T \text{diag}\,(\mathbf{y}) \, \tilde{K} \, \text{diag}\,(\mathbf{y}) \, \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1}$, which has to be maximized in the formulation (4.13) reduces to $-\frac{1}{2} \, \boldsymbol{\alpha}^T \text{diag}\,(\mathbf{y}) \, 2 \, K \, \text{diag}\,(\mathbf{y}) \, \boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{1}$ thanks to the condition that $\boldsymbol{\alpha}^T \mathbf{y} = 0$. This is a proper SVM optimization.

defined as:

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{N} \theta_i \, \xi_i \\ \text{s.t.} & y_i \, f(D(x_i, R)) \geq 1 - \xi_i, \;\; i = 1, \ldots, N \\ & \xi_i \geq 0, \end{array} \tag{4.14}$$

where either $\theta_i = 1$ for $i = 1, \ldots, N$ or $\theta_i = \frac{1}{n_1}$ if $y_i = 1$ and $\theta_i = \frac{1}{n_2}$, otherwise. It is argued in [19] that the latter formulation guarantees a nontrivial solution. This LP task can be solved by standard optimization methods, such as the simplex algorithm or interior-point methods [19]. Since no other constraints are included, the hyperplane is constructed in an n-dimensional dissimilarity space $D(\cdot, R)$. A sparse solution can be, however, imposed by minimizing the l_1-norm of the weight vector $\mathbf{w}$, $||\mathbf{w}||_1 = \sum_{j=1}^{n} |w_j|$, of the hyperplane (4.7). To formulate such a minimization task in terms of an LP problem (i.e. to eliminate the absolute value $|w_j|$ from the objective function), w_j is expressed by nonnegative variables α_j and β_j as $w_j = \alpha_j - \beta_j$. (When the pairs (α_j, β_j) are determined, then at least one of them is zero.) Similarly to the SVM formulation, nonnegative slack variables ξ_i, accounting for classification errors, and a regularization parameter γ are introduced. The minimization problem becomes then:

$$\begin{array}{ll} \text{Minimize} & \sum_{i=1}^{N} (\alpha_i + \beta_i) + \gamma \, \sum_{i=1}^{N} \xi_i \\ \text{s.t.} & y_i \, f(D(\mathbf{x}_i, R)) \geq 1 - \xi_i, \qquad i = 1, \ldots, N \\ & \alpha_i, \, \beta_i, \, \xi_i \geq 0. \end{array} \tag{4.15}$$

A more flexible formulation of a classification problem has been proposed by Graepel et al. [173]. Now, the problem is to minimize $||\mathbf{w}||_1 - \mu \, \rho$, which basically means that the margin ρ becomes a variable of the optimization problem. Note that $\rho = 1$ for the formulation (4.15). By imposing $||\mathbf{w}||_1$ to be constant, the modified version of (4.15) can be introduced as:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{n} \sum_{i=1}^{N} \xi_i - \mu \, \rho \\ \text{s.t.} & \sum_{i=1}^{N} (\alpha_i + \beta_i) = 1 \\ & y_i \, f(D(\mathbf{x}_i, R)) \geq 1 - \xi_i, \;\; i = 1, \ldots, N \\ & \xi_i, \, \alpha_i, \, \beta_i, \, \rho \geq 0. \end{array} \tag{4.16}$$

In this approach, a sparse solution $\mathbf{w}$ is obtained, which means that important objects are selected (by non-zero weights) from the original representation set R, resulting in a reduced set R_{so}. Therefore, this classifier can also be used for the selection of the representative objects, starting from $R := T$. This solution is similar to an adaptation of the SVM for feature representations defined with the LP machines [355, 369]. From the computational point of view, such an LP classifier is advantageous for two-class problems, since for new objects, only the dissimilarities to the objects from R_{so} have to be determined. Since multi-class problems are tackled by a number of two-class problems, in such cases, the reduction of R to R_{so} might be insignificant for the combined results.

Nearest neighbor classifier. The k-NN method constructed in a dissimilarity space relies on computing new dissimilarities (e.g. Minkowski distances) between object representations $D(x, R)$ in such a space. This means that indirectly an another dissimilarity representation is built over the given one.

Parzen classifier. The Parzen classifier models the class conditional probabilities $P(D(\cdot, R)|c_i)$ by density kernel estimation, here, the normal density function. Let σ_i be the smoothing parameter in the i-th dimension. The posterior probability of the class c_j is estimated as:

$$P(D(x, R)|c_j) = \frac{1}{n_j} \sum_{i=1}^{n_j} \frac{1}{\sqrt{2\,\pi} \, \prod_k \sigma_k} \, e^{-\frac{1}{2\,\sigma_i} (D(x,R) - D(x_i,R)) \, (D(x,R) - D(x_i,R))^T}. \tag{4.17}$$

4.5 Classification in pseudo-Euclidean spaces

Assume that R and T are identical. A symmetric dissimilarity $D(R,R)$ can be seen as a description of an underlying, lower-dimensional vector configuration X, which can be determined by a linear embedding of D as described in section 3.3. Remember that this procedure relies on the embedding of the Gram matrix $K := G$, derived from D^{*2} by formula (3.7), which can be seen as a Hermitian (symmetric) kernel in a Euclidean or pseudo-Euclidean space. So, one can, in fact, directly start from a similarity representation given by K. If $D(R,R)$ is asymmetric, then two symmetric DRs can be constructed $D_1 := \frac{1}{2}(D+D^T)$ and $D_2 := \frac{1}{2}(D-D^T)$ yielding two pseudo-Euclidean configurations X_1 and X_2. Then, two classifiers can be built and later combined; see also section 10. So, without loss of generality, we focus on symmetric representations. If $R \subset T$ (note that the embedding cannot be performed when $|R \cap T| \leq 1$), then our reasoning relies on the embedding of $D(R,R)$ and projecting the remaining objects $T \backslash R$ to an embedded space as described in section 3.3.5; see also Fig. 4.10. For simplicity of our presentation, we assume that $R{:=}T$.

If X is determined in a Euclidean space, then any traditional classifier, e.g. the ones described in the previous section, can be used. If X happens to be a pseudo-Euclidean representation, then the conventional classifiers should be adapted. Here, we limit ourselves to simple linear and quadratic decision rules, since they naturally rely on the pseudo-Euclidean inner products. See also section 2.4 for details on pseudo-Euclidean spaces.

Let $\mathcal{E} := \mathbb{R}^m = \mathbb{R}^{(p,q)}$, $m = p+q$, be a pseudo-Euclidean space. A linear function in $\mathcal{E}$ becomes then

$$f(\mathbf{x}) = \langle \mathbf{v}, \mathbf{x} \rangle_{\mathcal{E}} + v_0 = \mathbf{v}^T \mathcal{J}_{pq}\, \mathbf{x} + v_0, \quad \mathcal{J}_{pq} = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q}. \end{bmatrix}. \tag{4.18}$$

This decision rule can also be interpreted as $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + v_0$, where $\mathbf{w} = \mathbf{v}\mathcal{J}_{pq}$ in the associated Euclidean space $|\mathcal{E}| := \mathbb{R}^{p+q}$. (Remember that $\mathcal{E}$ is constructed by replacing the negative definite inner product in $\mathbb{R}^q$ by a positive definite one.) Analogous to the Euclidean case (i.e. from the Euclidean perspective), one can require that a signed distance of a point $\mathbf{x}$ to the hyperplane. That means that $\frac{\langle \mathbf{w},\mathbf{x}\rangle_{\mathcal{E}}+v_0}{||\mathbf{v}||^2_{\mathcal{E}}} = \frac{\mathbf{v}^T\mathcal{J}_{pq}\mathbf{x}+v_0}{\mathbf{v}^T\mathcal{J}_{pq}\mathbf{v}}$ is positive for objects $\mathbf{x}$ which lie on the same side of the hyperplane, where $\mathbf{v}$ is pointing to. Note that in order to satisfy this condition one should require that $||\mathbf{v}||^2_{\mathcal{E}} > 0$, otherwise the ambiguity arises if $||\mathbf{v}||^2_{\mathcal{E}}$ can have any sign[12]. In practice that means that the Euclidean norm of $\mathbf{v}$ in $\mathbb{R}^p$ should be larger than its Euclidean norm in $\mathbb{R}^q$. Since we start from positive dissimilarities, in practical cases the data are embedded such that the negative contribution is much smaller than the positive one. Hence, we will always have the case that the pseudo-norm of $\mathbf{v}$ is positive. Moreover, $||\mathbf{x}||^2_{\mathcal{E}} > 0$ for any $\mathbf{x} \in \mathbb{R}^{(p,q)}$ coming from the embedding of D. Note that this is not any longer guaranteed when one starts from an arbitrary symmetric (in a pseudo-Euclidean sense) kernel. Then, the negative contribution might be dominant, as e.g. for the kernel $K(\mathbf{x},\mathbf{y}) = x_1y_1 - \sum_{i=1}^m x_iy_i$. An illustration of possible and impossible linear classifiers is presented in Fig. 4.14. Note that such a situation can never result from an embedding of nonnegative dissimilarities. By the embedding of dissimilarities, spaces of higher dimensionality are obtained, hence it is actually hard to construct an example, which yields $R^{(1,1)}$ as the solution; see also Fig. 4.15.

Generalized nearest mean classifier (GNMC). The nearest mean classifier (NMC) is the simplest linear classifier which assigns an unknown object to the class of its nearest mean. In a pseudo-

[12] Imagine a simple case in $\mathbb{R}^{(1,1)}$, where $f(\mathbf{x}) = \mathbf{v}^T\mathcal{J}_{11}\mathbf{x} = 0$ and $\mathbf{v} = [0.5 \;\; -1]^T$ separates two classes of objects, as presented in Fig. 4.14, right plot. Since $||\mathbf{v}||^2_{\mathbb{R}^{(1,1)}} = -0.75$, then $\mathbf{v}$ and $\mathcal{J}_{11}\mathbf{v} = [0.5\;\; 1]$ point into different directions, similarly as presented in Fig. 4.14, on the right. Assume that the class labeled by $y_i = 1$ lies above the hyperplane (this is where $\mathbf{v}$ is pointing), while the class labeled by $y_j = -1$ is below the hyperplane (this is where $\mathcal{J}_{11}\mathbf{v}$ is pointing). Then, $\mathbf{x}_i = [1\;2]^T$ has a label $y_i = 1$ and $\mathbf{x}_j = [1\;\; -2]^T$ has a label $y_j = -1$. The signed distances to the hyperplane f are $dist(\mathbf{x}_i, f) = -3.33$ and $dist(\mathbf{x}_j, f) = 2$. So, the ambiguity arises.

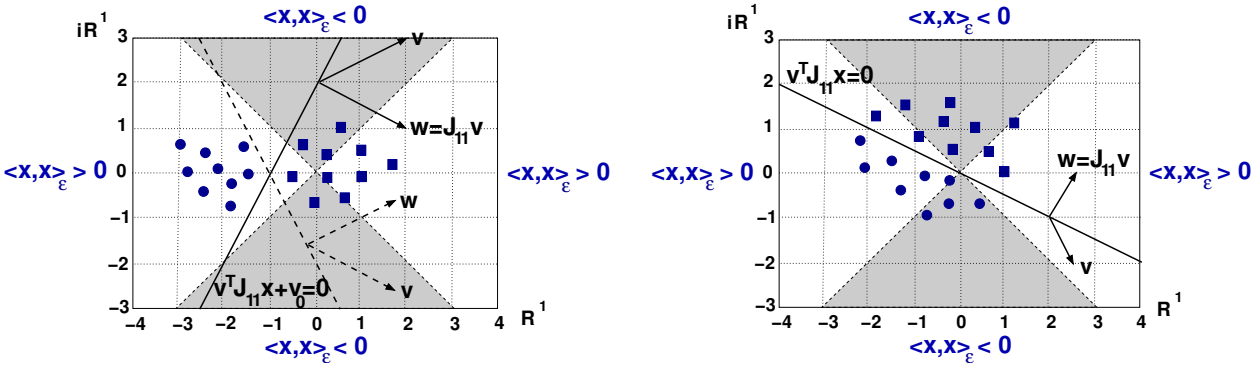


Fig. 4.14: A hypothetical example of a possible (left) and impossible (right) classification scheme by a linear classifier (solid line) for the data points in $\mathbb{R}^{(1,1)}$. Both cases will never result from the embedding of nonnegative dissimilarities, sine there exists a number of pairs of points which yield negative square pseudo-Euclidean distances. In the situation on the right, this also holds for pairs of objects coming from different classes. Note also that for the situation on the left, the two hyperplanes $\langle \mathbf{v}_1, \mathbf{x}\rangle_{\mathcal{E}} + v_0 = 0$ $\langle \mathbf{v}_2, \mathbf{x}\rangle_{\mathcal{E}} - v_0 = 0$, (marked by solid and dotted lines, respectively,) are related such that $\mathbf{v}_2 = \mathcal{J}_{11}\mathbf{v}_1$. Hence $||\mathbf{v}_1||_{\mathcal{E}} = ||\mathbf{v}_2||_{\mathcal{E}}$.

Euclidean space $\mathcal{E}$ such a decision is based on the pseudo-Euclidean distance. Given D, assume a two-class problem with the classes c_1 and c_2 and the embedded vector representation $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. Let $\overline{\mathbf{x}}_{(i)}$ be the mean vector of the class c_i in $\mathcal{E}$. For a new object z represented in this space as $\mathbf{z}_x$, the NMC classification rule becomes:

$$\text{Assign } z \text{ to } c_1, \text{ iff } d^2_{\mathcal{E}}(\mathbf{z}_x, \overline{\mathbf{x}}_{(1)}) < d^2_{\mathcal{E}}(\mathbf{z}_x, \overline{\mathbf{x}}_{(2)}), \text{ and assign } z \text{ to } c_2, \text{ otherwise.} \tag{4.19}$$

$d^2_{\mathcal{E}}(\mathbf{x}, \mathbf{y}) = ||\mathbf{x}-\mathbf{y}||^2_{\mathcal{E}} = (\mathbf{x}-\mathbf{y})^T \mathcal{J}_{pq} (\mathbf{x}-\mathbf{y})$ and $\mathcal{J}_{pq}$ is the fundamental symmetry in $\mathbb{R}^{(p,q)}$. Here, only the pseudo-Euclidean distances to the class mean vectors have to be computed. Such a classification process can be carried out in a somewhat modified way without performing the exact embedding of D (as needed in the case of (4.19)). As a result, the generalized nearest mean classifier is obtained.

Assume that the class c_i is represented by a dissimilarity matrix $D(R^i, R^i)$ based on the set $R^i = \{p^i_1, \ldots, p^i_{n_i}\}$. Let a new object z be represented by the dissimilarities to the set R^i. Then, the proximity of z to the class c_i is measured by the function f_i, defined as:

$$f_i(z) = \frac{1}{n_i} \sum_{j=1}^{n_i} d^2(z, p^i_j) - V_d(R^i), \text{ where } \quad V_d(R^i) = \frac{1}{2n_i^2} \sum_{j=1}^{n_i} \sum_{k=1}^{n_i} d^2(p^i_j, p^i_k). \tag{4.20}$$

$V_d(R^i)$ is a generalized average variance for the class c_i; see section 3.3.4 for details. Assume that X^i is a pseudo-Euclidean configuration obtained from the embedding of $D(R^i, R^i)$. So, X^i is represented in $\mathcal{E}_i := \mathbb{R}^{(p_i, q_i)}$. (Note that $\mathcal{E}_i$ usually differs from the space $\mathcal{E}$ referring to the complete matrix $D(R, R)$). It follows from section 3.3.4 that $f_i(z)$ can be equivalently formulated as $f_i(z) = ||\mathbf{z}^i_x - \overline{\mathbf{x}}^i||^2_{\mathcal{E}_i} = d^2_{\mathcal{E}_i}(\mathbf{z}^i_x, \overline{\mathbf{x}}^i)$, where $\mathbf{z}^i_x$ and $\overline{\mathbf{x}}^i$ are the representations of the object z and a mean vector of the entire R^i, respectively, expressed in $\mathcal{E}_i$. Hence, $f_i(z)$ measures the pseudo-Euclidean square distance of $\mathbf{z}^i_x$ to the mean of the i-th class. The interesting point is that such a distance can be computed *without* performing the embedding explicitly, since it operates only on the given dissimilarities D, formula (4.20). As a result, a K-class GNMC is defined as

$$\text{Assign } z \text{ to } c_j, \text{ iff } f_j(z) = \min_{i=1,\ldots,K} \{f_i(z)\}. \tag{4.21}$$

In summary, z is assigned to the class of the nearest mean, where each mean is described in an underlying space defined by the within-class dissimilarities. Additionally, we will derive what the average of the between-class square dissimilarities stands for.

Corollary⋆ 4.2 Assume an $n_i \times n_j$ submatrix $D_{ij} = D(R^i, R^j)$ describing the between-class dissimilarities for the classes c_i and c_j. Let $\mathcal{E}_{ij}$ denote a pseudo-Euclidean space resulting from the embedding of $D([R^i\ R^j],[R^i\ R^j])$. Let $\mathcal{J}_{ij}$ be the fundamental symmetry. Then, the average between-class dissimilarity $d_b^2(c_i,c_j)$ equals to

$$d_b^2(c_i,c_j) := \frac{1}{n_i n_j}\sum_{k=1}^{n_i}\sum_{l=1}^{n_j} d^2(p_k^i,p_l^j) = \frac{1}{n_i}\sum_{k=1}^{n_i}||\mathbf{x}_k^i||_{\mathcal{E}_{ij}}^2 + \frac{1}{n_j}\sum_{l=1}^{n_j}||\mathbf{x}_l^j||_{\mathcal{E}_{ij}}^2 - 2\,\langle\overline{\mathbf{x}}^i,\overline{\mathbf{x}}^j\rangle_{\mathcal{E}_{ij}}, \qquad (4.22)$$

where $\mathbf{x}_k^i$ and $\mathbf{x}_l^j$, as well as, $\overline{\mathbf{x}}^i$ and $\overline{\mathbf{x}}^j$ are represented now in the space $\mathcal{E}_{ij}$.

Proof. In general, $D^{*2} = \mathbf{g}\mathbf{1}^T + \mathbf{1}\mathbf{g}^T - 2\,G$, where $G = X\,\mathcal{J}_{pq}X^T$ is the Gram matrix of X and $\mathbf{g} = diag(G)$. Let $G_{ij} = X^i\,\mathcal{J}_{ij}(X^j)^T$, $\mathbf{g}_i := \text{diag}\,(G_{ii})$ and $\mathbf{g}_j := \text{diag}\,(G_{jj})$. Assume that $\mathbf{1}_{n_i}$ stands for a vector of the length n_i. Then, one has ${D^{*2}}_{ij} = \mathbf{g}_i\mathbf{1}_{n_j}^T + \mathbf{1}_{n_i}\mathbf{g}_j^T - 2G_{ij}$ and also, $\mathbf{1}_{n_i}^T\mathbf{g}_i = \text{tr}(G_{ii}) = \sum_{k=1}^{n_i}||\mathbf{x}_k^i||_{\mathcal{E}_{ij}}^2$. Now, one gets $d_b^2(c_i,c_j) = \frac{1}{n_i n_j}\,\mathbf{1}_{n_i}^T{D^{*2}}_{ij}\mathbf{1}_{n_j} = \frac{1}{n_i n_j}\,[\mathbf{1}_{n_i}^T\mathbf{g}_i\mathbf{1}_{n_j}^T\mathbf{1}_{n_j} + \mathbf{1}_{n_i}^T\mathbf{1}_{n_i}\mathbf{g}_j^T\mathbf{1}_{n_j} - 2\,\mathbf{1}_{n_i}^T G_{ij}\mathbf{1}_{n_j}] = \frac{1}{n_i n_j}\,[n_j\mathbf{1}_{n_i}^T\mathbf{g}_i + n_i\mathbf{g}_j^T\mathbf{1}_{n_j} - 2\,\mathbf{1}_{n_i}^T X^i\mathcal{J}_{ij}(X^j)^T\mathbf{1}_{n_j}] = \frac{1}{n_i}\sum_{k=1}^{n_i}||\mathbf{x}_k^i||_{\mathcal{E}_{ij}}^2 + \frac{1}{n_j}\sum_{l=1}^{n_j}||\mathbf{x}_l^j||_{\mathcal{E}_{ij}}^2 - 2\,\langle\overline{\mathbf{x}}^i,\overline{\mathbf{x}}^j\rangle_{\mathcal{E}_{ij}}$. ■

If we assume that the spaces $\mathcal{E}_i$, $\mathcal{E}_j$ and $\mathcal{E}_{ij}$ yield the same signatures (although this is not likely), then based on the relations (3.16) and (3.17), one can write $d_b^2(c_i,c_j) = V_d(R^i) + V_d(R^j) + ||\overline{\mathbf{x}}^i - \overline{\mathbf{x}}^j||_{\mathcal{E}_{ij}}^2$. By this, the square pseudo-Euclidean distance between class means in $\mathcal{E}_{ij}$ can be expressed by the use of the distances as:

$$||\overline{\mathbf{x}}^i - \overline{\mathbf{x}}^j||_{\mathcal{E}_{ij}}^2 = d_b^2(c_i,c_j) - V_d(R^i) - V_d(R^j). \qquad (4.23)$$

For two classes, the above equation is the difference between the average between-class square dissimilarities and the average within-class square dissimilarities. So, the value of $d_b^2(c_i,c_j) - V_d(R^i) - V_d(R^j)$ computed in a general case, approximates the square pseudo-distance between the class means in the embedded space.

In general, the NMC and the GNMC in a pseudo-Euclidean space are *not* identical classifiers. The NMC is trained in a pseudo-Euclidean space $\mathcal{E}$ found from a linear embedding of the complete D. Therefore, the dimensionality of $\mathcal{E}$ is determined by both the within-class and between-class dissimilarities. The GNMC operates only on the within-class dissimilarities. Although the embedding is not performed directly, the GNMC works in underlying feature spaces $\mathcal{E}_i$, defined for each class separately. It may happen that the signatures of feature spaces $\mathcal{E}_i$ are not the same. In such a case, the performances of the the NMC and the GNMC differ, because the NMC unifies the pseudo-Euclidean spaces and the signatures for all the classes, while the GNMC treats them separately. Since the GNMC makes use of distinct signatures, its accuracy is expected to be higher for problems in which the classes are described in different ways.

Fisher linear discriminant (FLD). To construct the Fisher linear discriminant, the notion of a pseudo-Euclidean covariance matrix is needed. For the representation of n vectors, it is defined as [152]:

$$C_{\mathcal{E}} = \frac{1}{n-1}\left[\sum_{i=1}^{n}(\mathbf{x}_i - \overline{\mathbf{x}})\,(\mathbf{x}_i - \overline{\mathbf{x}})^T\right]\mathcal{J}_{pq} = C\,\mathcal{J}_{pq}, \quad \overline{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i,$$

where C is the covariance matrix in the associated Euclidean space $|\mathcal{E}|$. Note that $C_{\mathcal{E}}$ is k-pd[13] (positive definite in a pseudo-Euclidean sense).

[13] First, $C_{\mathcal{E}}$ is self-adjoint (symmetric), Def2.74 and Def2.85, since $C_{\mathcal{E}}^* = \mathcal{J}_{pq}\,C_{\mathcal{E}}^T\,\mathcal{J}_{pq} = \mathcal{J}_{pq}\mathcal{J}_{pq}C^T\,\mathcal{J}_{pq} = C\,\mathcal{J}_{pq} = C_{\mathcal{E}}$ (here we made use of the fact that $\mathcal{J}_{pq}\mathcal{J}_{pq} = I$ and that $C = C^T$). Now, $C_{\mathcal{E}}$ is k-pd, since $\mathcal{J}_{pq}C_{\mathcal{E}}$ by Def. 2.92 is pd in an Euclidean sense. This is true, since C, as a pd matrix, can be expressed as $C := A^TA$. One, therefore, has: $\mathcal{J}_{pq}C_{\mathcal{E}} = \mathcal{J}_{pq}C\,\mathcal{J}_{qq} = \mathcal{J}_{pq}A^TA\,\mathcal{J}_{qq} = (A\,\mathcal{J}_{pq})^T(A\,\mathcal{J}_{pq})$, where the latter matrix is pd in an Euclidean sense by construction.

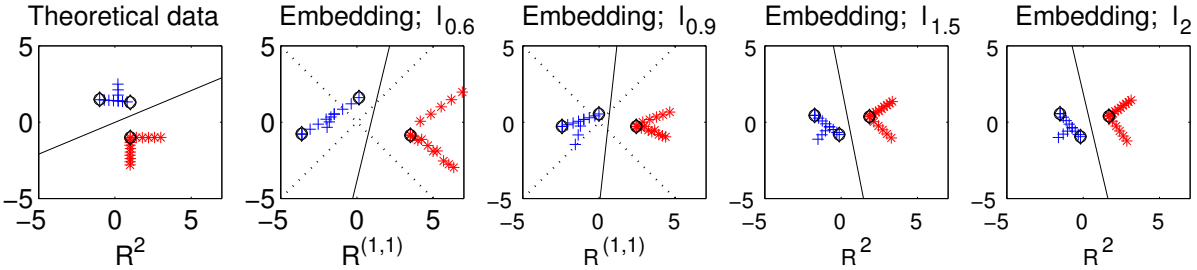


Fig. 4.15: A simple illustration of the FLD decision boundary in embedded spaces. The leftmost plot presents a 2D theoretical data. Only three points (marked by circles) are used for training, since then the data can be perfectly embedded in $\mathbb{R}^2$. The remaining points, marked by '+' and '*', belong to the test examples, projected on the retrieved (pseudo-)Euclidean spaces. The following plots show the embedding results of the l_p distance representations $D=(d_{ij})$, where $d_{ij}=(\sum_{k=1}^{2}|x_{ik}-x_{jk}|^p)^{1/p}$ and $p=\{0.6,0.9,1.5,2\}$. For positive $p<1$, the l_p distance is non-metric. In all the plots, the FLD determined by the three training points in the original or embedded spaces is drawn. For $p=2$ (the rightmost plot), the theoretical data are retrieved up to rotation.

Following [152], the FLD, $f(\mathbf{x})=\mathbf{v}^T\mathcal{J}_{pq}\,\mathbf{x}+v_0$, is obtained by maximizing (in a pseudo-Euclidean sense) the Fisher criterion $\max_{\mathbf{w}}\frac{\langle\mathbf{w},(C_B\mathcal{J}_{pq})\mathbf{w}\rangle_{\mathcal{E}}}{\langle\mathbf{w},(C_W\mathcal{J}_{pq})\mathbf{w}\rangle_{\mathcal{E}}}$. $C_B\,\mathcal{J}_{pq}$ and $C_W\,\mathcal{J}_{pq}$ are the pseudo-Euclidean between-class and pooled within-class covariance matrices, respectively. For a two-class problem, the FLD is determined by $\mathbf{v}=\mathcal{J}_{pq}\,C_W^{-1}\,(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)$ and $v_0=-\frac{1}{2}\,(\overline{\mathbf{x}}_1+\overline{\mathbf{x}}_2)^T\,\mathcal{J}_{pq}\mathbf{v}+\log\frac{p(c_1)}{p(c_2)}$, which can be simplified to:

$$f(\mathbf{x})=(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)^T C_W^{-1}\,\mathbf{x}-\frac{1}{2}\,(\overline{\mathbf{x}}_1+\overline{\mathbf{x}}_2)^T\,C_W^{-1}(\overline{\mathbf{x}}_1-\overline{\mathbf{x}}_2)+\log\frac{p(c_1)}{p(c_2)}, \tag{4.24}$$

where C_W is the pooled within-class covariance matrix in the associated Euclidean space $\mathbb{R}^{p+q}$, $\overline{\mathbf{x}}_1$ and $\overline{\mathbf{x}}_2$ stand for the class means and $p(c_1)$ and $p(c_2)$ are the prior probabilities. This means that the FLD in a pseudo-Euclidean space coincides with the FLD built in $\mathbb{R}^{p+q}$. See also Fig. 4.15.

Quadratic classifier (QC). Consider a pseudo-Euclidean space $\mathcal{E}:=\mathbb{R}^{(p,q)}$. Then, $C_{\mathcal{E}}=C\mathcal{J}_{pq}$ is a covariance matrix of the configuration X in $\mathbb{R}^{(p,q)}$ and C is the covariance matrix in the associated space $\mathbb{R}^{p+q}$. Analogous to the Euclidean case, the Mahalanobis distance between a vector $\mathbf{x}$ and the mean $\overline{\mathbf{x}}$ of X in $\mathcal{E}$ is given as $\langle(\mathbf{x}-\overline{\mathbf{x}}),C_{\mathcal{E}}^{-1}(\mathbf{x}-\overline{\mathbf{x}})\rangle_{\mathcal{E}}=(\mathbf{x}-\overline{\mathbf{x}})^T\mathcal{J}_{pq}C_{\mathcal{E}}^{-1}(\mathbf{x}-\overline{\mathbf{x}})=(\mathbf{x}-\overline{\mathbf{x}})^T C^{-1}(\mathbf{x}-\overline{\mathbf{x}})$. The latter follows from $C_{\mathcal{E}}^{-1}=\mathcal{J}_{pq}C^{-1}$ and $\mathcal{J}_{pq}\mathcal{J}_{pq}=I$. So, the quadratic classifier for a two-class problem can be constructed similarly to the Euclidean case (see also section 4.4.1) as:

$$f(\mathbf{x})=\sum_{i=1}^{2}(-1)^i\,(\mathbf{x}-\overline{\mathbf{x}}_i)^T\,C_i^{-1}\,(\mathbf{x}-\overline{\mathbf{x}}_i)+2\log\frac{p(c_1)}{p(c_2)}+\log\frac{|C_1\mathcal{J}_{pq}|}{|C_2\mathcal{J}_{pq}|}, \tag{4.25}$$

where C_1 and C_2 are the estimated class covariance matrices in $\mathbb{R}^{p+q}$, and $p(c_1)$ and $p(c_2)$ are the class prior probabilities. So, the QC in $\mathbb{R}^{(p,q)}$ coincides with the NQC in the associated $\mathbb{R}^{p+q}$.

Support vector machine (SVM). The principles behind the SVM in Euclidean (Hilbert) spaces are described in section 4.4.1. The SVM is presented there as $f(\mathbf{x})=\sum_{\alpha_i>0}\alpha_i\,y_i\,K(\mathbf{x},\mathbf{x}_i)+\alpha_0$, where K is a (conditionally) positive definite kernel. A linear kernel can be expressed as $K=X\,X^T$, which is equivalent to G, the Gram matrix defined by formula (3.7). Since the linear SVM is based on the inner products only and by the linear relation (3.7) between D^{*2} and K $(=G)$, the SVM can easily be constructed in the underlying space without performing the embedding explicitly, provided that the dissimilarities $D:=D(R,R)$ are Euclidean. For new objects, represented by $D(T_s,R)$, the SVM can immediately be tested by using K_n, the cross-Gram matrix between new objects and objects originally embedded, as explained in Corollary 3.46. Even simpler, if you consider $K:=-D^{*2}$,

then by footnote 9 on page 96, K is cpd, hence it can directly be used in the SVM optimization as given by formula (4.13). Moreover, since D is Euclidean, then D^{*2} is cnd by Theorem 3.31. Then, $K_1 := exp(-\sigma D^{*2})$ and $K_2 := (\sigma + D^{*2})^{*(-1)}$ are positive semidefinite, see Corollary 4.5, hence they can also directly be used as Mercer kernels in the SVM.

For a non-Euclidean D, the corresponding Gram matrix K is not pd, hence it refers to a pseudo-Euclidean space. The configuration X is found by (3.14), i.e. $X = Q\,|\Lambda|^{1/2}$, for which a linear classifier is defined by (4.18). If we now assign $\mathbf{w}^T = \mathbf{v}^T \mathcal{J}_{pq}$, then the classifier $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + v_0$ can be treated in the associated Euclidean space. The operation $\mathbf{v}^T \mathcal{J}_{pq}$ is seen as flipping the values of the vector $\mathbf{v}$ in all 'negative' directions of the pseudo-Euclidean space. This is equivalent to flipping the negative eigenvalues to positive ones and considering the inner product $K' = Q\,|\Lambda|\,Q^T$ in the associated Euclidean space as positive definite [172]. This procedure relies on the complete embedding of D.

Actually, one can try to define a SVM directly in the pseudo-Euclidean space. Consider a Hermitian (self-adjoint) kernel K in a pseudo-Euclidean space, Def. 2.100 (K could be considered in a Kreĭn space, but since K is finite, this reduces to the pseudo-Euclidean case). Consider a linear classifier $f(\mathbf{x}) = \mathbf{v}^T \mathcal{J}_{pq}\ \mathbf{x} + v_0$ in $\mathcal{E}$. Analogous to the Euclidean space, the margin between two separable classes equals $2/||\mathbf{v}||^2_{\mathcal{E}}$. The traditional SVM relies on finding a linear classifier maximizing the margin, hence minimizing the norm of the weight vector. This would translate to the minimization of the pseudo-Euclidean norm $\frac{1}{2}||\mathbf{v}||^2_{\mathcal{E}}$, which is a proper formulation in a pseudo-Euclidean space. Remember that in our case $||\mathbf{v}||^2_{\mathcal{E}} > 0$ is required, by our discussion in the first paragraph of this section, so the pseudo-Euclidean norm of $\mathbf{v}$ is bounded by zero from below. To require that the stationary point $\mathbf{v}_s$ (of a constrained problem) is taken as the minimum, one should require, by analogy to the Euclidean case, that in the neighborhood of $\mathbf{v}_s$ the Hessian H of $F(\mathbf{v}) = \frac{1}{2}||\mathbf{v}||^2_{\mathcal{E}}$, interpreted as the Hessian $H = \mathcal{J}_{pq}$ of $\frac{1}{2}\mathbf{v}^T \mathcal{J}_{pq}\mathbf{v}$ is k-pd in $|\mathcal{E}| := \mathbb{R}^{p+q}$. This is equivalent to stating that $\mathbf{v}^T \mathcal{J}_{pq}\mathbf{v}$ should be positive, which is true thanks to our requirement $||\mathbf{v}||^2_{\mathcal{E}} > 0$. So, the primal formulation of a 'soft margin' indefinite SVM can be solved by a non-convex QP as[10]:

$$\begin{array}{ll} \text{Minimize} & \frac{1}{2}\mathbf{v}^T \mathcal{J}_{pq}\mathbf{v} + \gamma \sum_{i=1}^n \xi_i \\ \text{s.t.} & y_i\,(\mathbf{v}^T \mathcal{J}_{pq}\mathbf{x}_i + w_0) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, n \\ & \xi_i \geq 0. \end{array} \tag{4.26}$$

The term $\sum_{i=1}^n \xi_i$ is an upper bound on the misclassification of the training samples and γ is a regularization parameter. Note that the indefinite SVM is not unique. For instance if $\mathbf{v}$ is the found solution such that $||\mathbf{v}||^2_{\mathcal{E}} = a > 0$, then for $\mathbf{v}' = \mathcal{J}_{pq}\mathbf{v}$, we have $||\mathbf{v}'||^2_{\mathcal{E}} = \mathbf{v}^T \mathcal{J}_{pq}\mathcal{J}_{pq}\mathcal{J}_{pq}\mathbf{v} = a$, which means that $\mathbf{v}'$ provides also a solution; see Fig. 4.14 for a simple illustration.

Let K be the Gram matrix determined in the embedding process of D. Then, the dual formulation becomes the same as (4.13), but for non cpd K:

$$\begin{array}{ll} \text{Maximize} & -\frac{1}{2}\,\boldsymbol{\alpha}^T \text{diag}(\mathbf{y})\,K\,\text{diag}(\mathbf{y})\,\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{1} \\ \text{s.t.} & \boldsymbol{\alpha}^T\mathbf{y} = 0, \\ & 0 \leq \alpha_i \leq \gamma, \qquad i = 1, 2, \ldots, n. \end{array} \tag{4.27}$$

The SVM becomes then $f(\mathbf{x}) = \sum_{\alpha_i > 0} \alpha_i\, y_i\, K(\mathbf{x}, \mathbf{x}_i) + \alpha_0$.

Analogous to a Hilbertian kernel, based on the definitions from section 2.4.1, K is a reproducing kernel, hence it defines a reproducing kernel pseudo-Euclidean space $\mathcal{K}_K$ on linear functions $h(\mathbf{x}) = \sum_i \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i)$. Note that the pseudo-norm of h in $\mathcal{K}_K$ is computed as $||h||_{\mathcal{K}_K} = \langle \sum_i \alpha_i\, y_i\, K(\mathbf{x}, \mathbf{x}_i), \sum_j \alpha_j\, y_j\, K(\mathbf{x}, \mathbf{x}_j)\rangle_{\mathcal{K}_K} = \boldsymbol{\alpha}^T \text{diag}(\mathbf{y})\,K\,\text{diag}(\mathbf{y})\,\boldsymbol{\alpha}$ due to the reproducing property

[10] Although $\mathcal{J}_{pq}$, $q > 0$, is not pd in a Euclidean sense, the optimized criterion is pd at the sought minimum.

of the kernel. Minimizing the latter quantity (or maximizing $-||h||_{\mathcal{K}_K}$) in the dual formulation above, corresponds then to bounding a class of hyperplanes h in the regularization principle, section 4.1.3.2. Hence, an indefinite SVM with a positive pseudo-norm $||h||_{\mathcal{K}_K}$ is a proper statistical learning algorithm. Note also that instead of a kernel K, directly $-D^{*2}$ can be used in the optimization (4.27). For proofs and the connection between the SVM and separation of convex hulls in pseudo-Euclidean spaces, see [182].

In summary, given a dissimilarity representation $D(R,R)$, the SVM classifier can be built in the underlying feature space as follows. First, the matrix $K := G$ is computed according to (3.7). If K is not pd, then either the problem is treated in a Euclidean space by considering the pd kernel $K' = Q\,|\Lambda|\,Q^T$ and using it to construct the SVM according to (4.11), or an indefinite SVM is built directly on K by solving (4.27). The latter case can only be accepted if the found solution is such that $\boldsymbol{\alpha}^T \mathrm{diag}(\mathbf{y})\, K\, \mathrm{diag}\,(\mathbf{y})\, \boldsymbol{\alpha}$ is positive.

4.6 Generalized kernels and classifiers in dissimilarity spaces

Kernels are usually understood as symmetric (Hermitian) operators in some Hilbert space being pd or cpd, see also Def. 2.63. Here, we will focus on real kernels. Any Mercer kernel, such as a finite symmetric pd matrix, can be seen as a Gram operator in some Hilbert space, hence as a (nonlinear) generalization of the similarity measure based on inner products. This holds due to the Mercer's condition, Theorem 2.66, which guarantees the existence of a mapping $\phi\colon \mathcal{X} \to \mathcal{H}$ from some input space $\mathcal{X}$ (which might not be explicitly given) to a Hilbert space $\mathcal{H}$ such that $K(x,y) = \langle \phi(x), \phi(y) \rangle$, where $\phi(x)$ is the image of $x \in \mathcal{X}$ in $\mathcal{H}$. Assume that K is real. Then, based on the notion of the norm, the squared distance in $\mathcal{H}$ is defined as $d^2_{\mathcal{H}}(x,y) := ||\phi(x) - \phi(y)||^2$. Thanks to the relation of $K(x,y) = \langle \phi(x), \phi(y) \rangle$, one has:

$$d^2_{\mathcal{H}}(x,y) := ||\phi(x) - \phi(y)||^2 = K(x,x) - 2\,K(x,y) + K(y,y). \tag{4.28}$$

Note that we write $d^2_{\mathcal{H}}(x,y)$ instead of $d^2_{\mathcal{H}}(\phi(x),\phi(y))$, since $d^2_{\mathcal{H}}$ can be determined by the kernel values only (without knowing ϕ).

Corollary* 4.3 $d^2_{\mathcal{H}}(x,y)$ is a cnd kernel.

Proof. Making use of Def. 2.63, it is sufficient to show that $\sum_{i,j}^n c_i\, c_j\, d^2_{\mathcal{H}}(x_i,x_j)$ is positive for all $n \in \mathbb{N}$ and all sets $\{x_1,\ldots,x_n\} \subseteq \mathcal{X}$ and $\{c_1,\ldots,c_n\} \subseteq \mathbb{C}$ such that $\sum_{i=1}^n c_i = 0$. One has: $\sum_{i,j}^n c_i\, c_j d^2_{\mathcal{H}}(x_i,x_j) = (\sum_{i=1}^n c_i)(\sum_{j=1}^n c_j K(x_j,x_j)) + (\sum_{j=1}^n c_j)(\sum_{i=1}^n c_i K(x_i,x_i)) - 2\sum_{i,j}^n c_i\, c_j K(x_i,x_j) = -2\sum_{i,j}^n c_i\, c_j K(x_i,x_j) < 0$, since $\sum_{i=1}^n c_i = 0$ and K is pd. ■

Note that $d^2_{\mathcal{H}}(x,y)$ is cnd only because K is pd. This is in agreement with our previous results discussed for finite Euclidean matrices and the corresponding Gram matrices, as presented in Theorem 3.31 and Theorem 3.38. By fixing an origin in $\mathcal{H}$ such that $K(x,x) := d^2_{\mathcal{H}}(x,0)$, formula (4.28) becomes

$$K(x,y) = -\frac{1}{2}\,[d^2_{\mathcal{H}}(x,y) - d^2_{\mathcal{H}}(x,0) - d^2_{\mathcal{H}}(y,0)]. \tag{4.29}$$

Note, however, that in a practical case, the distances refer to the mapped vectors $\phi(x_1),\ldots,\phi(x_n)$ in $\mathcal{H}$, hence the origin can only be chosen in their convex hull. So, the zero vector $\mathbf{0}$ in $\mathcal{H}$ can be set to a weighted mean of these vectors, i.e. $\mathbf{0} = \frac{1}{n}\sum_{i=1}^n s_i\,\phi(x_i) = \phi(\overline{x}_s) := \overline{\boldsymbol{\phi}}_s$, where $\mathbf{s}^T\mathbf{1} = 1$ and $\overline{\boldsymbol{\phi}}_s$ stands for a weighted mean in $\mathcal{H}$. By straightforward algebraic operations, similar to the ones in (3.4), one can find that $d^2_{\mathcal{H}}(x_i,\overline{x}_s) = \sum_{k=1}^n s_k\, d^2_{\mathcal{H}}(x_i,x_k) - \frac{1}{2}\sum_{k=1}^n\sum_{l=1}^n s_k\, s_l\, d^2_{\mathcal{H}}(x_k,x_l)$, where ϕ is omitted. Assuming that this expresses a square distance to the origin in $\mathcal{H}$, for $K \in \mathbb{R}^{n\times n}$, formula (4.29) translates to

$$K = -\frac{1}{2}(I - \mathbf{1}\mathbf{s}^T)\, D^{*2}_{\mathcal{H}}\, (I - \mathbf{s}\mathbf{1}^T), \quad \text{where } \mathbf{s}^T\mathbf{1} = 1. \tag{4.30}$$

K is pd iff $D^{*2}_{\mathcal{H}}$ is cnd (or equivalently, iff $-D^{*2}_{\mathcal{H}}$ is conditionally positive definite). This follows from our considerations in section 3.2.

Some of the kernel properties can be expressed in the continuous domain and by this an additional understanding can be gained. Now we briefly present some characteristics of the positive and conditionally negative definite kernels. Then, we explain how to interpret dissimilarities as distances from a possibly higher-dimensional space, where the mapping from an underlying abstract space is known only by the (generalized) inner product. This part is essential for understanding of some of our classification methods, introduced in chapter 9.

The class of pd kernels is closed under addition, multiplication by a positive constant and pointwise limits [22]. Moreover, it is also closed under the tensor product and a direct sum. Formally, we have:

Corollary 4.4 (Closure under the (tensor) product and the direct sum) [22, 74]

1. Let K_1, K_2 be Hermitian pd (psd) kernels. Then $K(x,y) = K_1(x,y)\, K_2(x,y)$ is also pd (psd).
   Proof: Proof follows from the Schur theorem [199] that the Hadamard product of positive definite matrices is also positive definite [22].
2. Let $K_1 \colon \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ and $K_2 \colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{C}$ be Hermitian pd kernels.
   Then $K_1 \odot K2((x_1,z_1),(x_2,z_2)) = K_1(x_1,x_2)\, K_2(z_1,z_2)$ is a pd kernel on $(\mathcal{X} \times \mathcal{Z}) \times (\mathcal{X} \times \mathcal{Z})$.
3. Let $K_1 \colon \mathcal{X} \times \mathcal{X} \to \mathbb{C}$ and $K_2 \colon \mathcal{Z} \times \mathcal{Z} \to \mathbb{C}$ be Hermitian pd kernels.
   Then $K_1 \oplus K_2((x_1,z_1),(x_2,z_2)) = K_1(x_1,x_2) + K_2(z_1,z_2)$ is a pd kernel on $(\mathcal{X} \times \mathcal{Z}) \times (\mathcal{X} \times \mathcal{Z})$.

Corollary 4.5 (Relations between pd and cnd kernels) Let K and D be real kernels and let $\sigma > 0$. One has [22, 74]:

1. If K is psd, then $\tilde{K} := e^{*\sigma K} = (e^{\sigma K_{ij}})$ is psd.
   Proof: By the Taylor expansion, one gets $e^{*\sigma K} = \sigma(\mathbf{1}\mathbf{1}^T + K + \frac{1}{2!}K^{*2} + \frac{1}{3!}K^{*3} + ...)$. By Corollary 4.4, K^{*r} is psd for a positive integer r, hence their sum is psd as well.
2. D^{*2} is cnd iff $\tilde{K} := e^{-*\sigma\, D^{*2}}$ is psd.
   Proof: We know that $D^{*2} = \text{diag}\,(K)\mathbf{1}^T + \mathbf{1}\text{diag}\,(K)^T - 2\,K$. D^{*2} is cnd is equivalent to K being pd. Then, one has $\tilde{K} := e^{-*\sigma\, D^{*2}} = e^{-\sigma\,(\text{diag}\,(K)\mathbf{1}^T + \mathbf{1}\text{diag}\,(K)^T - 2\,K)} = e^{-\sigma\,\text{diag}\,(K)\mathbf{1}^T} e^{2\,\sigma\,K} e^{-\sigma\,\mathbf{1}\text{diag}\,(K)^T} = U e^{2\,\sigma\,K} U$, where U is a diagonal matrix of positive numbers $U := \text{diag}\,(e^{-\sigma K})$, hence U is pd. Since $e^{2\,\sigma\,K}$ is psd by the statement above, then $\tilde{K}$ is psd by the Schur theorem.
3. D^{*2} is cnd iff $\tilde{K} := (\sigma + D^{*2})^{*(-1)} = (1/(\sigma + d^2_{ij}))$ is psd.
   Sketch of proof: First, it is trivial to show that if D^{*2} is cnd, then $D^{*2} + \sigma\mathbf{1}\mathbf{1}^T$ is cnd as well. Next, note that $\int_0^\infty e^{-(\sigma+z)x}\, dx = \frac{1}{\sigma+z}$. Then, $\mathbf{c}^T \tilde{K}\, \mathbf{c} = \mathbf{c}^T (\sigma + D^{*2})^{*(-1)}\, \mathbf{c} = \int_0^\infty \mathbf{c}\, (e^{-(\sigma + D^{*2})} * (x\, \mathbf{1}\mathbf{1}^T))\, \mathbf{c}\, dx$. By the point above, for positive x, the matrix $(e^{-D^{*2} + \sigma \mathbf{1}\mathbf{1}^T}) * (x\, \mathbf{1}\mathbf{1}^T)$ is psd. So, $\tilde{K}$ is psd as well.
4. If D^{*2} is cnd and for all x, one has $D^{*2}(x,x) \geq 0$, then $\tilde{K}_1 := D^{*2r}$, with $r \in (0,1)$ and $\tilde{K}_2 := \log(1 + D^{*2})$ are cnd.

From section 2.3.1, we know that any symmetric pd kernel defined on a compact set or an index set T is a reproducing kernel for a Hilbert space $\mathcal{H}_K$ consisting of bounded linear maps defined by the evaluation map $\phi \colon x \to K(x,\cdot)$. Hence, $\mathcal{H}_K$ contains all finite linear combinations of the form $h(x) \sum_k a_k\, K(x_k, x)$. As a result, $K(x,y) = \langle K(x,\cdot), K(y,\cdot) \rangle_{\mathcal{H}_K}$. If T is a set of finite cardinality, say n, then the functions are evaluated only at a finite number of points. Consequently, the RKHS becomes an n-dimensional space, where the functions simplify to n-dimensional vectors.

Now, we propose to consider *generalized kernels* as arbitrary symmetric countable (finite or infinite) matrices. Such similarity matrices can be seen as kernels of the pseudo-Euclidean space $\mathcal{E}$ (or, more general, of a Kreĭn space). Remember that any symmetric matrix K is self-adjoint in the pseudo-

Euclidean sense, which is guaranteed by the fact that $\mathcal{J}_{pq}K := K^T(I - \frac{1}{n}\mathbf{1}\mathbf{1}^T)_{pq}$; see Def. 2.74. Therefore, one has $K(x,y) = \langle \psi(x), \psi(y) \rangle$, where $\psi(x)$ is the image of an object x in $\mathcal{E}$. Based on a logically appealing extension from the positive definite inner product to the indefinite inner product, the squared distance in $\mathcal{E}$ is defined as $d^2_{\mathcal{E}}(x,y) := ||\psi(x) - \psi(y)||^2_{\mathcal{E}}$, which reduces to $d^2_{\mathcal{E}}(x,y) = K(x,x) - 2\,K(x,y) + K(y,y)$. By similar considerations as for the pd and cpd kernels, the equivalent formulations for an indefinite K are obtained as:

$$K(x,y) = -\frac{1}{2}\,[d^2_{\mathcal{E}}(x,y) - d^2_{\mathcal{E}}(x,0) - d^2_{\mathcal{E}}(y,0)] \tag{4.31}$$

and also

$$K = -\frac{1}{2}(I - \mathbf{1}\mathbf{s}^T)\, D^{*2}_{\mathcal{E}}\, (I - \mathbf{s}\mathbf{1}^T), \quad \text{where } \mathbf{s}^T\mathbf{1} = 1, \tag{4.32}$$

where $D^{*2}_{\mathcal{E}}$ is a matrix of square pseudo-Euclidean (Kreĭn) distances in $\mathcal{E}$. Hence, K and D^{*2} described above are related by linear operations. Any of them can determine the corresponding space $\mathcal{E}$. Remember also that K is a reproducing kernel in $\mathcal{E}$ as follows from section 2.4.1.

An asymmetric matrix K can uniquely be described by two symmetric matrices, i.e. $K_1 := \frac{1}{2}(K + K^T)$ and $K_2 := \frac{1}{2}(K - K^T)$, where each of them can be treated as a generalized kernel. If K is nearly symmetric, then K_2 contains little information, hence negligible. In such cases, $K \approx K_1$. Such transformations are needed for the interpretation of K in pseudo-Euclidean spaces. For an asymmetric K, the corresponding D^{*2} is defined as $D^{*2} = \mathrm{diag}\,(K)\mathbf{1}^T + \mathbf{1}\mathrm{diag}\,(K)^T - K - K^T$, which is symmetric. An asymmetric square dissimilarity matrix can then be considered as $D^{*2} = \mathrm{diag}\,(K)\mathbf{1}^T + \mathbf{1}\mathrm{diag}\,(K)^T - 2\,K$. Anyway, asymmetric D^{*2} or K can directly be treated for building classifiers in a dissimilarity space.

4.6.1 Connection between dissimilarity spaces and pseudo-Euclidean spaces

The idea of building classifiers in dissimilarity spaces is general, since such classifiers can be interpreted as some classifiers in the underlying pseudo-Euclidean spaces. Hence, there is a connection between these two concepts. Assume a dissimilarity representation $D(R,R)$ and the corresponding matrix K of inner products derived as $K := -\frac{1}{2}(I - \frac{1}{r}\mathbf{1}\mathbf{1}^T)D^{*2}(I - \frac{1}{r}\mathbf{1}\mathbf{1}^T)$. Let X be a pseudo-Euclidean configuration in $\mathbb{R}^{(p,q)}$ obtained from the embedding of D. (This actually means that there exists a mapping $\phi : p_j \to \mathbf{x}_j$, where $\mathbf{x}_j := \phi(p_j)$.) Then, $K := X\mathcal{J}_{pq}X^T$. Consider now a general linear classifier built in a dissimilarity space $D^{*2}(\cdot, R)$. Then, we have:

Proposition⋆ 4.6 A linear classifier $f(D(z,R)) = \sum_{j=1}^{r} w_j\, d^2(z,p_j) + w_0$ constructed in a dissimilarity space $D^{*2}(\cdot,R)$ becomes a quadratic classifier in the underlying pseudo-Euclidean space $\mathbb{R}^{(p,q)}$.

Proof. Let X result from the embedding of $D(R,R)$ into $\mathbb{R}^{(p,q)}$. Let the object z be represented in $\mathbb{R}^{(p,q)}$ as a vector $\mathbf{z}$. Based on the relations between the square distances D^{*2} and inner products K, one can write: $f(D^{*2}(z,R)) = \sum_{j=1}^{n} w_j\, d^2(z,p_j) + w_0 = \sum_{j=1}^{n} w_j\,[K(z,z) - 2\,K(z,p_j) + K(p_j,p_j)] + w_0 = \sum_{j=1}^{n} w_j\,[\mathbf{z}^T\mathcal{J}_{pq}\mathbf{z} - 2\mathbf{z}^T\mathcal{J}_{pq}\mathbf{x}_j + \mathbf{x}_j^T\mathcal{J}_{pq}\mathbf{x}_j] + w_0 = \mathbf{w}^T\mathbf{1}\,\mathbf{z}^T\mathcal{J}_{pq}\mathbf{z} - 2\,\mathbf{w}^T X\mathcal{J}_{pq}\mathbf{z} + \mathbf{w}^T\mathrm{diag}\,(X\mathcal{J}_{pq}X^T) + w_0$. The latter formulation describes a quadratic classifier in $\mathbb{R}^{(p,q)}$. ■

If D is Euclidean, then $\mathbb{R}^{(p,q)}$ simplifies to a Euclidean space $\mathbb{R}^p$. Without loss of generality, a similar relation holds for a dissimilarity space $D(\cdot,R)$, which can be seen as $\tilde{D}^{*2}(\cdot,R)$, where $\tilde{D} := D^{*1/2}$. So, the linear classifier $f(D(z,R))$ is in fact a quadratic classifier in the underlying pseudo-Euclidean space $\mathbb{R}^{(p',q')}$ as determined by the embedding of $D^{*1/2}$. If D is Euclidean, then $D^{*1/2}$ is Euclidean as well, as guaranteed by Theorem 3.41. Another important observation is that the quadratic classifier in $\mathbb{R}^{(p',q')}$, becomes an even more nonlinear decision rule, when projected to the $\mathbb{R}^{(p,q)}$ space. In fact, any monotonically increasing nonlinear transformation of $g(D)$, such as D^{*r}, where $r \in (0,1)$ or $\mathrm{sigm}(D)$ will influence the nonlinearity of $f(g(D(z,R)))$ as observed in the $\mathbb{R}^{(p,q)}$

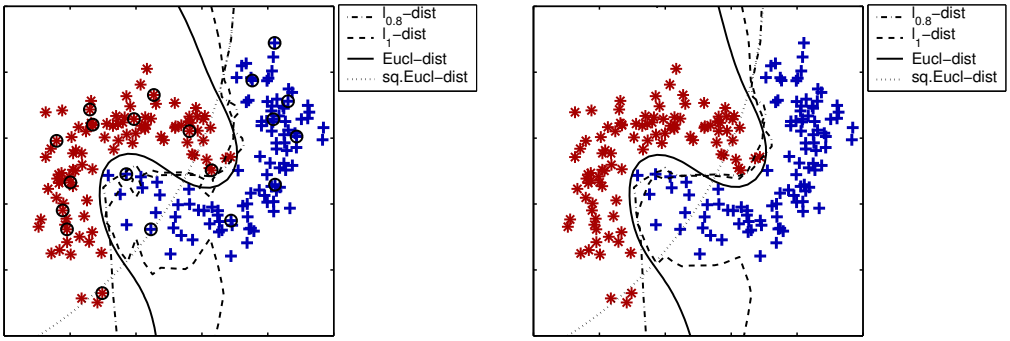


Fig. 4.16: Assume 2-dimensional theoretical banana data. Four dissimilarity representations are considered: $D_k(T,R)$, $k=1,2,3,4$, based on the $l_{0.7}$-distance (non-metric), l_1-distance (metric, non-Euclidean), Euclidean and square Euclidean distance, correspondingly. A linear classifier $f(D_k(x,R))=\sum_j w_j D_k(x,p_j)$ is trained on each $D_k(T,R)$, where T is a training set of 200 points and R is either a subset of T consisting of 20 points chosen by the k-centres procedure (such points minimize the maximum of the dissimilarities over all objects to their nearest neighbors; see also section 7.1.2) or $R=T$. Formally, a (R)NLC classifier $f(D_k(x,R))$ is built in a dissimilarity space of the dimensionality $|R|$. Since the theoretical data are 2D, a discrimination boundary can be drawn in the original 2D space. The subplots show the data points and the projected discrimination boundaries found originally in four dissimilarity spaces $D_k(\cdot,R)$. The left subplot presents the results when $R\subset T$, where points of R are marked by circles. The right subplot shows the results when $R:=T$, hence a regularized classifier had to be used. Note that a linear classifier in a square Euclidean dissimilarity space $D_4(T,R)$ is quadratic in the original space, which is in agreement with our observations made in section 4.6.1. Other classifiers are nonlinear with respect to ${D^*}^2_k$. This example shows that the decision boundaries in both plots look similar, so an adequate and small representation set R may serve for a good discrimination.

space. Analogous to the linear case, a quadratic classifier in a dissimilarity space would translate to a 4-th order polynomial in the corresponding pseudo-Euclidean space. Note also that a linear classifier built in a *similarity space* $K(\cdot,R)$, i.e. $f(K(z,R)):=\sum_j w_j\,K(z,p_j)+w_0=\mathbf{w}^T X\mathcal{J}_{pq}\mathbf{z}+w_0$ is a linear classifier in $\mathbb{R}^{(p,q)}$. This can also be used for any similarity kernel derived from the dissimilarities by a monotonically decreasing transformation, e.g. $\tilde{K}:=(e^{-d_{ij}^2/\sigma^2})$ or $\tilde{K}:=((d_{ij}^2+\sigma^2)^{-1})$. If D is Euclidean, then based on Corollary 4.5, such transformed kernels describe relations in some Hilbert spaces.

For a dissimilarity representation $D(T,R)$, where $R\subset T$, a linear classifier in dissimilarity spaces can be approximated by a quadratic classifier in the underlying pseudo-Euclidean space corresponding to the embedding of $D(R,R)$ and projecting the remaining $T\backslash R$ objects there. The reason of such an approximation is caused by the orthogonal projections of the $T\backslash R$ objects which are likely to yield some errors. This means that the dissimilarities $D(T\backslash R,R)$ are not ideally preserved. Such an approximation can still be very good. Fig. 4.16 should help in getting some intuition.

Proposition* 4.7 Assume a two-class classification problem described by $D(T,R)$ and the labels $y_j\in\{1,-1\}$. A linear classifier $f(D^{*2}(z,R))=\sum_{j=1}^n w_j y_j\,d^2(z,p_j)+w_0$, constructed such that $\mathbf{w}^T\mathbf{y}=0$ in a dissimilarity space $D^{*2}(\cdot,R)$ is a linear classifier in the underlying pseudo-Euclidean space $\mathbb{R}^{(p,q)}$.

Proof. Let the object z be represented in $\mathbb{R}^{(p,q)}$ as a vector $\mathbf{z}$. Following the same reasoning as above, one has: $f(D^{*2}(z,R))=\sum_{j=1}^n w_j y_j\,d^2(z,p_j)+w_0=\mathbf{w}^T\mathbf{y}\,\mathbf{z}^T\mathcal{J}_{pq}\mathbf{z}-2\,\mathbf{w}^T\mathrm{diag}\,(\mathbf{y})X\mathcal{J}_{pq}\mathbf{z}+\mathbf{w}^T\mathrm{diag}\left(\mathrm{diag}\,(\mathbf{y})X\mathcal{J}_{pq}X^T\right)+w_0=-2\,\mathbf{w}^T\mathrm{diag}\,(\mathbf{y})X\mathcal{J}_{pq}\mathbf{z}+\mathbf{w}^T\mathrm{diag}\left(\mathrm{diag}\,(\mathbf{y})X\mathcal{J}_{pq}X^T\right)+w_0$. The quadratic term vanishes due to the requirement $\mathbf{w}^T\mathbf{y}=0$. So, the latter formulation describes a linear classifier in $\mathbb{R}^{(p,q)}$. From a computational point of view, it might be useful to consider a classifier on $-D^{*2}(\cdot,R)$, which becomes $f(-D^{*2}(z,R))=2\,\mathbf{w}^T\mathrm{diag}\,(\mathbf{y})X\mathcal{J}_{pq}\mathbf{z}-\mathbf{w}^T\mathrm{diag}\left(\mathrm{diag}\,(\mathbf{y})X\mathcal{J}_{pq}X^T\right)+w_0$. ■

An example of such a classifier is the SVM. Note that the same reasoning as above holds for the classifier $f(g(D(z,R)))$, where g is a monotonically increasing nonlinear transformation.

4.7 Discussion

Since the notion of proximity underpins the description of a class as a group of similar objects, we propose to move the emphasis from features to a proper proximity measure. This leads to representations based on this concept. Here, mainly dissimilarity representations $D(T,R)$ are considered. These are relative representations describing the pairwise dissimilarities between the objects from a (training) set T and the representation set R. The strength of such representations comes from their general applicability as they can be derived from any measurements or structural descriptions, such as strings or graphs, or some other intermediate representations. Since a learning problem can be characterized by various kinds of expert knowledge, as a result a number of dissimilarity representations can be created and combined to better describe the underlying concept. This is studied in chapter 10.

This chapter is devoted to learning approaches, mostly classification, on dissimilarity representations. Three main strategies are distinguished, which rely on various interpretations of the dissimilarities:

1. The first approach focuses on the relations in local neighborhoods, defined for each object by the dissimilarities to its neighboring objects. This is always applicable, although, in general, a large representation set R is needed for a good performance.
2. The second strategy defines classifiers in a dissimilarity space, a vector space where each dimension corresponds to a dissimilarity to a representation object. This paradigm can be used for any dissimilarity measure. Classifiers built there rely on the dissimilarities to all objects from R. Hence, this is a global approach.
3. The third methodology is applicable for symmetric measures and when $R \subseteq T$. However, since any square asymmetric representation D can be expressed as a sum of two symmetric representations $D_1 := \frac{1}{2}(D + D^T)$ and $D_2 := \frac{1}{2}(D - D^T)$, each of them can be considered separately and the results can be combined. The learning algorithm relies first on determining a pseudo-Euclidean vector configuration such that the dissimilarities $D(T,R)$ are preserved as well as possible. Then, many traditional classifiers can be modified and applied in such a space.

All dissimilarity-based learning approaches are designed for numerical representations, hence they reside in some spaces. Ineluctably, they make use of the statistical methodologies already developed in vector spaces and adapt them appropriately. The innovation of our methods lies in the acceptance of any nonnegative dissimilarity measure satisfying the reflexivity condition (hence also non-Euclidean and non-metric). These two requirements are not only logical, but enable a clear interpretation of the compactness hypothesis, where a small dissimilarity depicts a good resemblance of the compared objects. Our algorithms can handle negative dissimilarities as well. The problem lies, however, in an adequately found meaning of such dissimilarities.

Although our focus is put to dissimilarities, there is an algebraic relation between dissimilarity and similarity representations. One can be derived from the other by proper linear operations. This holds both for their interpretations in inner product and indefinite inner product spaces, namely Euclidean (Hilbert) and pseudo-Euclidean (Kreĭn) spaces. Therefore, any symmetric $n \times n$ similarity matrix can be seen as a generalized inner product (Gram) matrix in the corresponding (pseudo-)Euclidean space. Based on such an inner product, a symmetric square distance can be defined. So, any symmetric $n \times n$ square dissimilarity matrix can be understood as a matrix of square pseudo-

Euclidean distances. Because of such relations, linear decision rules built in dissimilarity spaces can be presented as quadratic (or linear) classifiers in the underlying pseudo-Euclidean spaces.

All these considerations refer to dissimilarity representations, comparing pairs of objects. A natural extension is to depict a relation of one entity to a number of them or of a partial concept to the whole concept, e.g. a resemblance of an object to a (sampled) domain, or a particular process to a model process. This would require that a measure itself is learned from a collection of objects belonging to a class, as well as, other non-class representatives. Such representations are still an open issue.

5. Dissimilarity measures

> *In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it. I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meager and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be.*
>
> "POPULAR LECTURES AND ADDRESSES", SIR WILLIAM THOMPSON, LORD KELVIN

Relative similarity can be defined as a relationship between two entities which are of the same nature or possess the same characteristics, but in different measure or degree[1]. The larger the similarity value, the better the resemblance between the objects. Relative dissimilarity, on the other hand, focuses on the differences; the smaller the dissimilarity, the more alike the objects. Both similarity and dissimilarity values express the notion of likeness between objects, but their emphasis is different. Which is more suitable to define depends on the type of data and a problem at hand. In general, such a proximity is a function of the observed variables or collected measurements. We will refer to it as to a measure, although it might not be such in the classical sense of the probability or set theory.

In this chapter, we will present a brief overview of (dis)similarity measures for various types of data, together with their brief characteristics. Some of them are well known, while others are relatively new. The measures defined on the features are described in section 5.1. Section 5.2 elaborates further on probabilistic measures, i.e. dissimilarity measures between distributions. Such measures are important when we deal with images, sets of points or representations of the data by clouds of vectors in a vector space, since such data can be described by probability functions. In sections 5.3 and 5.4, we will move to measures, more specifically used in the pattern learning area; these are measures created in the process of matching of two sequences, shapes or digitally represented objects. Some more important dissimilarity measures are described more thoroughly to emphasize their properties and a potential use. Section 5.5 finishes this chapter with a brief survey on measures developed for particular applications, while section 5.6 presents a general summary.

5.1 Measures depending on feature types

In the statistical approach, the data objects are described by features. Although such representations are not our main concern here, the learning methods designed for them constitute an important basis; see section 4.4. Therefore, some attention will be devoted to features. Moreover, the use of dissimilarities is an option for data consisting of mixed features. We distinguish the following feature types: binary, categorical, ordinal, symbolic and quantitative, introduced in Def. 5.1. These types might not be sufficient for the complete description, since the real-world data may suffer from (selective) lack of information which leads to imprecise, vague, probabilistic or even missing data.

Def. 5.1 (Feature types) Let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be a set of features, called also variables or attributes, and $\mathcal{D}_{f_i}$ be a set of valid values for f_i. The following features $f \in \mathcal{F}$ can be considered:

[1] The word 'relative' emphasizes the pairwise comparisons of objects. Conceptual measures are not discussed here.

- *binary* if $\mathcal{D}_f$ is a set of two symbols or two numbers, e.g. $0/1$ to encode the gender.
- *categorical* if $\mathcal{D}_f$ is a finite, discrete set of numbers, e.g. from 1 to 4 to encode hair color. Here, we also include the case of a *discrete* feature, i.e. a feature with distinct and separate values, which can be counted, such as the number of children.
- *symbolic* or *nominal* if $\mathcal{D}_f$ is a finite, discrete set of symbols; e.g. nationality. Symbolic features represent a set of possible values, symbols or modalities. Their values can be counted, but not ordered.
- *quantitative* if f is measured on an interval and $\mathcal{D}_f$ is a convex subset of $\mathbb{R}$; e.g. height, temperature or the time required to reach a chosen place by car.
- *ordinal* if $\mathcal{D}_f$ is a finite, discrete set of ordered symbols, e.g. a scale from 1 to 5 representing the answers of 'strongly dislike', 'dislike', 'neutral', 'like' and 'strongly like', after tasting a particular food product. The distinction between consecutive points on the scale is not necessarily always the same; the difference in pleasure expressed by giving a rating of 2 rather than 1 might be much less than by giving a rating of 4 instead of 3.

Measures for dichotomous data. Dichotomous (or binary) features have only two values possible. They represent either the presence (1) or absence (0) of a particular characteristics or some opposite qualities, e.g. such as large (1) and small (0). The i-th object is represented by a binary vector $\mathbf{x}_i \in \mathcal{B}^m$, where $\mathcal{B} = \{0, 1\}$. For $\mathbf{x}_i,\ \mathbf{x}_j \in \mathcal{B}^m$, $\mathbf{x}_i^T \mathbf{x}_j = \sum_{k=1}^{m} x_{ik}\, x_{jk}$ is the binary scalar product and $\mathbf{1} - \mathbf{x}$ is the complementary vector of $\mathbf{x}$. This allows us to define the following counters:

- $a_{ij} := \mathbf{x}_i^T \mathbf{x}_j$ - the number of properties common to both objects
- $b_{ij} := \mathbf{x}_i^T (\mathbf{1} - \mathbf{x}_j)$ - the number of properties which i has and j lacks
- $c_{ij} := (\mathbf{1} - \mathbf{x}_i)^T \mathbf{x}_j$ - the number of properties which j has and i lacks
- $d_{ij} := (\mathbf{1} - \mathbf{x}_i)^T (\mathbf{1} - \mathbf{x}_j)$ - the number of properties that both objects lack

		object j	
		1	0
object i	1	a_{ij}	b_{ij}
	0	c_{ij}	d_{ij}

where $a_{ij} + b_{ij} + c_{ij} + d_{ij} = m$[1]. For various definitions of similarity measures, a 2×2 contingency table is considered for each pair of objects i and j as given above on the right.

A number of measures has been proposed based on these values; see e.g. [12, 13, 72, 171]. Some of them are presented in Table 5.1, where the suffices i and j are omitted for simplicity. Such measures are often binary equivalents of other well-known formulations. For instance, in Table 5.1, the first measure is the binary dot product, the Jaccard measure is the similarity ratio, the Ochiai measure refers to the cross-product ratio, while the Pearson2 measure corresponds to the binary correlation coefficient. Gower [171] introduced also two families of binary similarity coefficients depending on a parameter θ and defined as (the suffices i and j are dropped):

$$S_\theta = \frac{a + d}{a + d + \theta\,(b + c)} \quad \text{and} \quad T_\theta = \frac{a}{a + \theta\,(b + c)}, \tag{5.1}$$

For particular values of θ the above measures reduce to some of the forms presented in Table 5.1. For instance, S_1 corresponds to the simple matching similarity and $T_{1/2}$ refers to the Dice similarity. The metric and Euclidean properties of the dissimilarities $1 - S_\theta$, $1 - T_\theta$ and their square roots depend on θ. They are summarized below:

Theorem 5.2 (Gower) (see [171] for proofs)

1. $1 - S_\theta$ and $1 - T_\theta$ are metric for $\theta \geq 1$. $(1 - S_\theta)^{1/2}$ and $(1 - T_\theta)^{1/2}$ are metric for $\theta \geq 1/3$.
2. If $(1 - S_\theta)^{1/2}$ is Euclidean, then so is $(1 - S_\phi)^{1/2}$ for $\phi \geq \theta$. The same relation holds for T_θ.
3. $(1 - S_\theta)^{1/2}$ is Euclidean for $\theta \geq 1$ and $(1 - T_\theta)^{1/2}$ is Euclidean for $\theta \geq 1/2$.

[1] Although d is used to denote both the counter and the dissimilarity, its use is apparent from the context.

Table 5.1: Similarity and dissimilarity measures for dichotomous features [72, 171].

Reference	Similarity S	Range	S psd	Dissimilarity D			
				$D=(1-S)^{\frac{1}{2}}$		$D=1-S$	
				Metric	Euclidean	Metric	Euclidean
Russel & Rao	$\frac{a}{a+b+c+d}$	$[0,1]$	Yes	Yes	Yes	Yes	No
Simple matching	$\frac{a+d}{a+b+c+d}$	$[0,1]$	Yes	Yes	Yes	Yes	No
Jaccard	$\frac{a}{a+b+c}$	$[0,1]$	Yes	Yes	Yes	Yes	No
Dice	$\frac{a}{a+(b+c)/2}$	$[0,1]$	Yes	Yes	Yes	No	No
Sokal & Sneath	$\frac{a+d}{a+(b+c)/2+d}$	$[0,1]$	No	Yes	No	No	No
Anderberg	$\frac{a}{a+2(b+c)}$	$[0,1]$	Yes	Yes	Yes	Yes	No
Rogers & Tanimoto	$\frac{a+d}{a+2(b+c)+d}$	$[0,1]$	Yes	Yes	Yes	Yes	No
Kulczynski	$\frac{1}{2}\left(\frac{a}{a+b}+\frac{a}{a+c}\right)$	$[0,1]$	No	No	No	No	No
Anderberg2	$\frac{1}{4}\left(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{c+d}+\frac{d}{b+d}\right)$	$[0,1]$	No	No	No	No	No
Hamman	$\frac{(a+d)-(b+c)}{a+b+c+d}$	$[-1,1]$	Yes	Yes	Yes	Yes	No
Yule	$\frac{ad-bc}{ad+bc}$	$[-1,1]$	No	No	No	No	No
Pearson	$\frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	$[0,1]$	Yes	Yes	Yes	No	No
Pearson2	$\frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$	$[-1,1]$	Yes	Yes	Yes	No	No
Ochiai	$\frac{a}{\sqrt{(a+b)(a+c)}}$	$[0,1]$	Yes	Yes	Yes	No	No

Reference	Dissimilarity D	Range	Metric	Euclidean
Binary Euclidean	$(b+c)^{\frac{1}{2}}$	$[0,\infty)$	Yes	Yes
Hamming	$b+c$	$[0,\infty)$	Yes	No
Variance	$\frac{b+c}{4(a+b+c+d)}$	$[0,\infty)$	Yes	No
Bray-Curtis	$\frac{b+c}{2a+b+c}$	$[0,1]$	No	No
Binary size difference	$\frac{(b-c)^2}{(a+b+c+d)^2}$	$[0,\infty)$	No	No
Binary pattern difference	$\frac{bc}{(a+b+c+d)^2}$	$[0,1]$	No	No
Binary shape difference	$\frac{(a+b+c+d)(b+c)-(b-c)^2}{(a+b+c+d)^2}$	$[-\infty,\infty)$	No	No

Measures for categorical data. Let X be a categorical $n\times m$ data matrix and let the feature f_k take values in c_k categories such that $c=\sum_{k=1}^{m} c_k$. Dissimilarity measures defined for binary data, Table 5.1, can now be adapted for the categorical data, as well. To achieve that, one has to code each m-dimensional data vector $\mathbf{x}_i$ into a c-dimensional binary vector $\tilde{\mathbf{x}}_i=[\tilde{\mathbf{x}}_{(1)};\ldots\tilde{\mathbf{x}}_{(m)}]$. $\tilde{\mathbf{x}}_{(k)}$ is a vector of the length c_k consisting of all zeros except for 1 at the j-th position assuming that x_{ik} belongs to the j-th category [118].

Measures for ordinal data. Let X be an ordinal $n\times m$ data matrix such that the feature f_k has c_k categories and $c=\sum_{k=1}^{m} c_k$. In case of ordinal variables, the dissimilarity measure should take into account the positions of categories in the ordering, and it should be larger for more distant categories than for close ones. Here, a generalization of the Jaccard dissimilarity, Table 5.1, can be used for a comparison of the objects p_i and p_j, as follows:

$$d(p_i,p_j)=\frac{\sum_{k=1}^{m} x_{ik}+\sum_{k=1}^{m} x_{jk}+2\sum_{k=1}^{m}\min(x_{ik},x_{jk})}{\sum_{k=1}^{m} x_{ik}+\sum_{k=1}^{m} x_{jk}-\sum_{k=1}^{m}\min(x_{ik},x_{jk})}. \tag{5.2}$$

Table 5.2: Dissimilarity measures for quantitative features; $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$.

Reference	D	Dissimilarity $d(\mathbf{x}, \mathbf{y})$	Metric	Euclidean
Euclidean	D_E, D_2	$\sqrt{(\mathbf{x}-\mathbf{y})^T(\mathbf{x}-\mathbf{y})}$	Yes	Yes
City block	D_1	$\sum_{i=1}^m \lvert x_i - y_i \rvert$	Yes	No
Max norm	D_{max}	$\max_i \lvert x_i - y_i \rvert$	Yes	No
l_p or Minkowski	D_p	$[\sum_{i=1}^m \lvert x_i - y_i \rvert^p]^{1/p},\ p \geq 1,\ p \neq 2$	Yes	No
Mahalanobis	D_M	$\sqrt{(\mathbf{x}-\mathbf{y})^T C^{-1} (\mathbf{x}-\mathbf{y})};\ C$ is psd	Yes	Yes
Median distance	D_{med}	$D_{[n/2]-rank}$	No	No
Correlation-based	D_{corr}	$\frac{1}{2}\left(1 - \frac{\mathbf{x}^T\mathbf{y}}{\lvert\lvert\mathbf{x}\rvert\rvert^2 + \lvert\lvert\mathbf{y}\rvert\rvert^2}\right)$	No	No
Correlation-based	D_{corr2}	$\frac{1}{2}\left(1 - \frac{\mathbf{x}^T\mathbf{y}}{\lvert\lvert\mathbf{x}\rvert\rvert^2 + \lvert\lvert\mathbf{y}\rvert\rvert^2 - 2\mathbf{x}^T\mathbf{y}}\right)$	No	No
Cosine	D_{cos}	$\frac{1}{2}\left(1 - \frac{\mathbf{x}^T\mathbf{y}}{\lvert\lvert\mathbf{x}\rvert\rvert \lvert\lvert\mathbf{y}\rvert\rvert}\right)$	No	No
Divergence	D_{div}	$\sqrt{\sum_{i=1}^n \frac{(x_i - y_i)^2}{(x_i + y_i)^2}}$	No	No
Bray and Curtis	D_{BC}	$\frac{\sum_{i=1}^n \lvert x_i - y_i \rvert}{\sum_{i=1}^n x_i + y_i}$	No	No
Soergel	D_S	$\frac{\sum_{i=1}^n \lvert x_i - y_i \rvert}{\sum_{i=1}^n \max\{x_i, y_i\}}$	No	No
Ware and Hedges	D_{WH}	$\sum_{i=1}^n \left(1 - \frac{\min\{x_i, y_i\}}{\max\{x_i, y_i\}}\right)$	No	No

Another approach relies on coding the ordinal vectors into the binary ones. The object p_i can be represented as a c-dimensional binary vector $\mathbf{x}_i^* = [\mathbf{x}_{(1)}^* \ldots \mathbf{x}_{(m)}^*]^T$, where $\mathbf{x}_{(k)}^*$ is a binary vector of the length c_k consisting of first h_k ones, followed by $c_k - h_k$ zeros. The observation x_{ik} takes h_k-th of the c_k ordered values for the feature f_k. Now, any binary dissimilarity can be applied.

Measures for quantitative data. Many measures exist for quantitative variables, mostly constructed in an additive way after counting the differences for each variable separately; see [37, 72, 118, 120, 171]. Some of them are presented in Table 5.2. The basic measures come from the family of l_p distances. The l_p metric, for $p \geq 1$ is defined as $d_p(\mathbf{x}, \mathbf{y}) := ||\mathbf{x} - \mathbf{y}||_p = [\sum_{i=1}^m (x_i - y_i)^p]^{1/p}$, which for $p = 1$ becomes the city block distance and for $p = 2$, the Euclidean distance; see also Example 2.31.

A second order statistical dependence among m quantitative variables can be described by their covariance matrix Σ. Then, the Euclidean distance can be generalized into the Mahalanobis distance $d_M^2(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \Sigma^{-1} (\mathbf{x} - \mathbf{y})$. If Σ is unknown, its sample estimate C based on n objects is used. C is then estimated either as $C = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T$ or, when k classes of the cardinalities n_i are known, it becomes: $C = \frac{1}{n-k} \sum_{i=1}^k \sum_{j=1}^{n_i} (\mathbf{x}_j - \overline{\mathbf{x}}_{(i)})(\mathbf{x}_j - \overline{\mathbf{x}}_{(i)})^T$, where $\overline{\mathbf{x}}_{(i)}$ is the mean for the i-th class. For the transformed data with the identity covariance matrix, d_M^2 becomes Euclidean.

Measures for symbolic data. Symbolic objects are described by m variables f_i, each on the domain $\mathcal{D}_{f_i}$ and a logical statement of the form $[f_i \in \mathcal{X}_i]$, where $\mathcal{X}_i \subseteq \mathcal{D}_{f_i}$, e.g. $[\text{color} \in \{red, green, yellow\}]$ or $[\text{weight} \in (10, 20)]$. A symbolic object x is expressed as the Cartesian product of the values $x_i = f_i(x)$ with the total event being a conjunction of all the feature events. The dissimilarity between two objects $x = [f_1 \in \mathcal{X}_i] \wedge \ldots \wedge [f_m \in \mathcal{X}_m]$ and $y = [f_1 \in \mathcal{Y}_i] \wedge \ldots \wedge [f_m \in \mathcal{Y}_m]$ can be defined as [168]:

$$d(x, y) = \sum_{i=1}^m [d_p(x_i, y_i) + d_s(x_i, y_i) + d_c(x_i, y_i)], \tag{5.3}$$

where d_p, d_s and d_c, normalized to $[0,1]$, denote the components due to position, span and content, respectively. The component d_p, valid for quantitative variables only, indicates the relative positions of two variable values. By writing $\mathcal{X}_i = [x_i^l, x_i^u]$ and $\mathcal{Y}_i = [y_i^l, y_i^u]$ with the lower x_i^l and upper x_i^u limits, one has $d_p(x_i, y_i) = |x_i^l - y_i^l|/|D_{f_i}|$, where $|D_{f_i}|$ is the range of f_i over all the objects. The remaining two measures, d_s and d_c are defined for quantitative, symbolic or ordinal attributes. The component d_s indicates the relative sizes of the variable values without referring to the common parts between them as $d_s(x_i, y_i) = |l_x - l_y|/\text{span}\,(x_i, y_i)$. For quantitative values, $l_x = |x_i^u - x_i^l|$ and $l_y = |y_i^u - y_i^l|$, and the span, the length of the minimum interval containing both x_i and y_i, equals to $\text{span}\,(x_i, y_i) = |\max\{x_i^u, y_i^u\} - \max\{x_i^l, y_i^l\}|$. For other features, $l_x = |\mathcal{X}_i|$, $l_y = |\mathcal{Y}_i|$ and the span becomes $|\mathcal{X}_i \cup \mathcal{Y}_i|$. The component d_c measures the common parts between the variables: $d_c(x_i, y_i) = |l_x + l_y - 2\,\text{length}\,(\mathcal{X}_i \cap \mathcal{Y}_i)|/\text{span}\,(x_i, y_i)$. For other dissimilarity measures for symbolic objects, see for instance [54, 55, 202, 254].

Gower's generalized dissimilarity coefficient. A classical measure for data of mixed types is the Gower's [170] dissimilarity. First, a general similarity measure for m variables is introduced as:

$$s_{ij} = \frac{\sum_{k=1}^{m} w_k\, \delta_{ijk}\, s_{ijk}}{\sum_{k=1}^{m} w_k\, \delta_{ijk}}, \tag{5.4}$$

where $s_{ijk} = s\,(p_i, p_j)_k$ is the similarity between objects p_i and p_j based on the k-th variable only, and $\delta_{ijk} = 1$ if the objects considered can legitimately be compared and zero otherwise, as e.g. in case of missing values. For the dichotomous variables, $\delta_{ijk} = 0$ if $x_{ik} = x_{jk} = 0$ and $\delta_{ijk} = 1$, otherwise. The strength of feature contributions is determined by the weights w_k, which can also be omitted if $w_k = 1$ for each k. The similarity s_{ijk}, for $i, j = 1, \ldots, n$ and $k = 1, \ldots, m$ is then defined as:

$$s_{ijk} = s\,(p_i, p_j)_k = \begin{cases} 1 - \frac{|x_{ik} - x_{jk}|}{r_k}, & f_k \text{ is quantitative} \\ \mathcal{I}\,(x_{ik} = x_{jk} = 1), & f_k \text{ is dichotomous} \\ \mathcal{I}\,(x_{ik} = x_{jk}), & f_k \text{ is categorical} \\ 1 - g\left(\frac{|x_{ik} - x_{jk}|}{r_k}\right), & f_k \text{ is ordinal,} \end{cases} \tag{5.5}$$

where r_k is the range of the k-th variable and g is a chosen monotonic transformation.
Let $S_G = (s_{ij})$, then the Gower's dissimilarity matrix D_G is defined as $D_G = (\mathbf{1}\mathbf{1}^T - S_G)^{*1/2}$. The Gower's distance is Euclidean if no missing values occur [170].

Other heterogeneous measures. Cox and Cox [73] proposed an extension to the Gower measure, which can be used for both mixed and non-mixed data, producing simultaneously dissimilarities between pairs of objects and between pairs of variables. Also, many measures can be designed for mixture types, e.g. by combining the coefficients from Tables 5.1 and 5.2, either with, or without appropriate weighting. Some other heterogeneous distance measures are suggested in [421], which can handle missing data and nominal variables in the case where class labels are available. There, the performance of the 1-NN rule using the scaled Euclidean distance is compared to the alternative distance measures on a number of datasets, finding out that the former can be outperformed significantly.

A model of Tversky. A number of models studied in cognitive sciences assumes that human similarity assessment is based on the measurement of a distance in a psychological space [37, 164–166, 418]. Objects are treated as points in a perceptual space and their difference is expressed by a metric. Tversky [401] argued that from a human perception's point of view, metric requirements are not verified in practice. He claimed that a comparison of individuals is described by different sets of attributes. Hence, a *feature contrast model* was proposed, where instances are characterized

by sets of features, instead of interpreting them as points in a metric space. Assume feature sets $\mathcal{F}_i$ and $\mathcal{F}_j$ given for the instances x_i and x_j, respectively. Then, the similarity between x_i and x_j can be evaluated as $s_T(x_i,x_j) = \frac{f(\mathcal{F}_i \cap \mathcal{F}_j)}{f(\mathcal{F}_i \cup \mathcal{F}_j)+\alpha f(\mathcal{F}_i - \mathcal{F}_j)+\beta f(\mathcal{F}_j - \mathcal{F}_i)}$, where f is a non-negative function. This measure describes the contrast between the common and distinctive features. Depending on the choice of α,β and f, different models can be obtained. An underlying assumption is that objects are characterized either by binary features or by features whose values correspond to the presence or absence of some attributes. Consequently, if $\mathcal{F}_i$ and $\mathcal{F}_j$ are sets of dichotomous features and f is the cardinality of a set, $f(P) = |P|$, then the Tversky similarity can be expressed as $s_T(x_i,x_j) = a_{ij}/(a_{ij} + (1+\alpha)\, b_{ij} + (1+\beta)\, c_{ij})$, where a_{ij}, b_{ij} and c_{ij} are the counters defined before in the paragraph on dichotomous features. For suitable choices of α and β, some of the similarity measures presented in Table 5.1 can be obtained. Alternatively, the Tversky similarity can be expressed in the following form:

$$s_T(x_i,x_j) = f(\mathcal{F}_i \cap \mathcal{F}_j) - \alpha f(\mathcal{F}_i - \mathcal{F}_j) - \beta f(\mathcal{F}_j - \mathcal{F}_i). \tag{5.6}$$

The feature information can also be graded. This is achieved by the use of fuzzy features [332–334], represented as membership functions $\mu_f : \mathcal{D} \to$ degree, where each legal value of the domain $\mathcal{D}$ has a degree indicating to what extent this value is true. The membership functions can be subjected to arbitrary simplifications; usually continuous functions are used such as logistic, Gaussian or piecewise linear. Let ϕ_i correspond now to a set of measurements of the object x_i and $\mu_k(\phi_i)$ be the k-th fuzzy feature. Given m feature, $\mu(\phi_i) \equiv \mathcal{F}_i = \{\mu_1(\phi_i), \mu_2(\phi_i), \ldots, \mu_m(\phi_i),\}$. Let us denote $\mu_{ki} := \mu_k(\phi_i)$. The intersection and the difference between $\mathcal{F}_i$ and $\mathcal{F}_j$ can be then defined as: $\mathcal{F}_i \cap \mathcal{F}_j = \{\min(\mu_{ki}, \mu_{kj})\}_{1 \le k \le m}$ and $\mathcal{F}_i - \mathcal{F}_j = \{\max(\mu_{ki} - \mu_{kj}, 0)\}_{1 \le k \le m}$. The Tversky similarity (5.6) becomes then:

$$s_T(x_i,x_j) = \sum_{k=1}^{m} \min(\mu_{ki},\mu_{kj}) - \sum_{k=1}^{m} \alpha \max(\mu_{ki} - \mu_{kj}, 0) - \sum_{k=1}^{m} \beta \max(\mu_{kj} - \mu_{ki}, 0) \tag{5.7}$$

and the dissimilarity is given as $d_T(x_i,x_j) = m - s_T(x_i,x_j)$. The Tversky similarity relies on considerations from set theory. Still, the min and max operators can be approximated by smooth functions. If h is the Heaviside function: $h(x) = \mathcal{I}(x \ge 0)$, then a logistic function $h_\sigma(x) = 1/(1 + \exp(-\sigma x)$, approximates h with any desired error for any non-zero x (for $x = 0$, the error is 0.5, independently of σ). The min and max operators can be approximated[2] as $s_\sigma(x,y) = x\, h_\sigma(y-x) + y\, h_\sigma(x-y)$ and $l_\sigma(x,y) = x\, h_\sigma(x-y) + y\, h_\sigma(y-x)$, respectively. So, in formula (5.7), these operators can be replaced appropriately. This leads to the following dissimilarity d_T:

$$d_T(x_i,x_j) = \sum_{k=1}^{m} \alpha\, s_\sigma(\mu_{ki} - \mu_{kj}, 0) + \sum_{k=1}^{m} \beta\, s_\sigma(\mu_{kj} - \mu_{ki}, 0) - \sum_{k=1}^{m} l_\sigma(\mu_{ki},\mu_{kj})]. \tag{5.8}$$

The Tversky's idea can be adopted to define new dissimilarity measures for continuous features. For instance, the similarity between two instances w.r.t. the feature f_k can be measured as $s_{f_k}(x_i,x_j) = (r_k - |f_{ik} - f_{jk}|)/r_k$, where f_{ik} is the value of the k-th feature for the i-th object and r_k is the range of f_k. If $\alpha = \beta = 0$, then the original Tversky's model simplifies to $f(\mathcal{F}_i \cap \mathcal{F}_j)/f(\mathcal{F}_i \cup \mathcal{F}_j)$. Let f be a weighted linear combination of the features, then we can define the total similarity s_T as:

$$s_T(x_i,x_j) = \frac{\sum_{k=1}^{\mathcal{F}_i \cap \mathcal{F}_j} w_k\, s_{f_k}(x_i,x_j)}{\sum_{k=1}^{\mathcal{F}_i \cup \mathcal{F}_j} w_k}, \tag{5.9}$$

where w_k are suitable weights assigned to the features. An asymmetric similarity can be obtained in the same way as above, but for $\alpha = 0$ and $\beta = -1$. Yet, such a similarity should express the degree of inclusion of x_i into x_j by using $\sum_{k=1}^{\mathcal{F}_i} w_k$ in the denominator of formula (5.9) instead.

[2] This approximation is due to $\min(x,y) = x\, h(y-x) + y\, h(x-y)$ and $\max(x,y) = x\, h(x-y) + y\, h(y-x)$.

5.2 Measures between populations

To analyze the differences between populations described by vectors in a feature space, a number of dissimilarity measures can be considered. If the mean vectors are used to represent entire populations, they can be used to compute the between-group dissimilarities according to formulas from Table 5.2. The evaluation of the inter-population dissimilarity may also rely on the description of a population by a multivariate probability distribution function (pdf) $F(\mathbf{x})$. Then, the difference between two populations is measured by the dissimilarity between two pdf's F_1 and F_2. A Kolmogorov metric [146] is commonly used. For two distribution functions F_1 and F_2 it is defined as

$$D_K(F_1, F_2) = \sup_{\mathbf{x}} |F_1(\mathbf{x}) - F_2(\mathbf{x})|. \tag{5.10}$$

For some general probability measures and their relations, see [146] and the following sections.

Normal distributions

An assumption of normally distributed data is often made in practice, hence there is a need for proper dissimilarity measures. A classical measure between two normal distributions $\mathcal{N}(\boldsymbol{\mu}_1, \Sigma)$ and $\mathcal{N}(\boldsymbol{\mu}_2, \Sigma)$ with the equal covariance matrices Σ is the Mahalanobis distance D_M between their means:

$$D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma) = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T \Sigma^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \tag{5.11}$$

Since the true distribution parameters are hardly known, they are replaced by the sample estimates: $\overline{\mathbf{x}}_{(i)} = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_j$, $i = 1, 2$ and $C = \frac{(n_1-1)\, C_1 + (n_2-1)\, C_2}{n_1+n_2-2}$, where n_i denotes the sample sizes and $\overline{\mathbf{x}}_i$ and C_i, $i = 1, 2$, represents the sample mean vectors and sample covariance matrices, respectively. If $C = I$ or $C = \mathrm{diag}(\sigma_i)$, then the D_M^2 becomes the Euclidean or weighted Euclidean distance between the mean vectors, correspondingly. Note, however, that if the Mahalanobis distance is considered w.r.t. the space $X = \mathcal{N}(\mu, \Sigma)$, (X, d_M) is premetric; see Example 2.39.

The Mahalanobis distance is based on the assumption of equal covariance matrices. For heterogeneous covariance matrices, its generalization leads to the normal information radius [212]:

$$d_{NIR}(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, C_1, C_2) = \begin{cases} \frac{1}{2} \log_2 \frac{\frac{1}{2}|(C_1+C_2)| + \frac{1}{4}(\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)^T (\overline{\mathbf{x}}_1 - \overline{\mathbf{x}}_2)}{|C_1|^{1/2} |C_2|^{1/2}}, & \text{if } C_1 \neq C_2 \\ \frac{1}{2} \log_2 (1 + \frac{1}{4} D_M^2(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2; C), & \text{if } C \equiv C_1 = C_2 \end{cases} \tag{5.12}$$

All other measures for normal distributions are presented in the next section.

Divergence measures

Many classical measures expressing the difference between two probability distributions F_1 and F_2 with the density functions f_1 and f_2 are special cases of the ϕ-divergence proposed by Csiszár [76], which is based on the likelihood ratio $\lambda(\mathbf{x}) = f_2(\mathbf{x}) / f_1(\mathbf{x})$:

$$d_\phi(F_1, F_2) = E\left[\phi\left(\lambda(X)\right)\right] = \int_{\mathcal{D}} \phi\left(\lambda(\mathbf{x})\right) dF_1(\mathbf{x}) = \int_{\mathcal{D}} \phi\left[f_2(\mathbf{x}) / f_1(\mathbf{x})\right] f_1(\mathbf{x})\, d\mu(\mathbf{x}), \tag{5.13}$$

where $\phi(\lambda)$ is a real, convex function defined on $\mathbb{R}_+$ such that $\phi(1) = 0$, and μ is a measure over the domain $\mathcal{D}$. Note that by inverting the arguments F_1 and F_2 of $d_\phi(F_1, F_2)$, another ϕ-divergence is obtained, i.e. $d_\phi(F_2, F_1)$ becomes $d_{\lambda\, \phi(1/\lambda)}(F_1, F_2)$. Moreover, the symmetric divergence, $d_\phi(F_1, F_2) + d_\phi(F_2, F_1)$, can be considered as $d_{\phi(\lambda) + \lambda\, \phi(1/\lambda)}(F_1, F_2)$ [118].

Some well-known divergence measures [118] for continuous and univariate histogram-like distributions are presented below, together with the equivalent formulas in case of two normal distributions.

Formulations for discrete distributions are omitted since they are straightforward generalizations of the continuous ones. For brevity, let us denote $\mathcal{N}_i := \mathcal{N}(\boldsymbol{\mu}_i, \Sigma_i)$, for $i = 1, 2$, $\Sigma := \Sigma_1 = \Sigma_2$, for equal covariance matrices and the square Mahalanobis distance by D_M^2. The histogram-like distributions f_1 and f_2 are constant on disjoint intervals $I_1^{(1)}, \ldots, I_{N_1}^{(1)}$ and $I_1^{(2)}, \ldots, I_{N_2}^{(2)}$, respectively such that $f_i(x) = \sum_{\tau=1}^{N_i} h_\tau^{(i)}\, \mathcal{I}(x \in I_\tau^{(i)})$, $i = 1, 2$, where $h_\tau^{(i)}$ are positive weights. $J_{st} := I_s^{(1)} \cap I_t^{(2)}$ stands for the intersection of the two intervals $I_s^{(1)}$ and $I_t^{(2)}$ and $\mu(J_{st})$ is the length (Lebesgue measure) of J_{st}.

Kullback-Leibler divergence. This measure, known also as information distance or relative entropy [263], is obtained for $\phi(\lambda) = \lambda \log(\lambda)$, $\lambda > 0$ and $\phi(0) = 0$:

$$d_{KL}(F_1, F_2) = \int_{\mathcal{D}} \log\left(f_1(\mathbf{x})/f_2(\mathbf{x})\right) f_1(\mathbf{x})\, d\mathbf{x}. \tag{5.14}$$

The usual convention is $\log(0/b) = 0$ for all b and $\log(a/0) = \infty$ for all non-zero a. Hence, d_{KL} yields values in $[0, \infty]$. This measure is asymmetric, hence non-metric. For two m-dimensional normal distributions, d_{KL} becomes:

$$d_{KL}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2}\left(D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma_1) + \text{tr}\,(\Sigma_1^{-1}\Sigma_2 - I) + \log\frac{|\Sigma_1|}{|\Sigma_2|}\right) \tag{5.15}$$

or, in case of equal covariance matrices, one gets $d_{KL}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2} D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma)$.
For two histogram-like distributions, d_{KL} equals to:

$$d_{KL}(F_1, F_2) = \sum_{s=1}^{N_1}\sum_{t=1}^{N_2} \log(h_t^{(2)}/h_s^{(1)})\, h_t^{(2)}\, \mu(J_{st}).$$

J-coefficient. For $\phi(\lambda) = (\lambda - 1)\log(\lambda)$, we get a symmetric Kullback-Leibler divergence:

$$d_J(F_1, F_2) = d_{KL}(F_1, F_2) + d_{KL}(F_2, F_1). \tag{5.16}$$

For two m-dimensional normal distributions, d_J becomes:

$$d_J(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{2}\left[D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma_1) + D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma_2) + \text{tr}\,(\Sigma_1^{-1}\Sigma_2 - I) + \text{tr}\,(\Sigma_2^{-1}\Sigma_1 - I)\right] \tag{5.17}$$

or in case of equal covariance matrices, $d_J(\mathcal{N}_1, \mathcal{N}_2) = D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma)$.
For two histogram-like distributions, one has:

$$d_J(F_1, F_2) = \sum_{s=1}^{N_1}\sum_{t=1}^{N_2} \log(h_t^{(2)}/h_s^{(1)})\, (h_t^{(2)} - h_s^{(1)})\, \mu(J_{st}).$$

Information radius. This is a symmetric measure obtained for $\phi(\lambda) = -\frac{1}{2}(1+\lambda)\log(1 + \frac{\lambda}{2})$:

$$d_{IR}(F_1, F_2) \equiv d_\phi(F_1, F_2) + d_\phi(F_2, F_1). \tag{5.18}$$

For two normal distributions, d_{IR} becomes the normal information radius, as given by formula (5.12).

Hellinger coefficient. This similarity measure is obtained for $\phi(\lambda) = \lambda^t$, where $t \in (0, 1)$:

$$s_H^{(t)}(F_1, F_2) = \int_{\mathcal{D}} f_2(\mathbf{x})^t\, f_1(\mathbf{x})^{1-t}\, d\mathbf{x}. \tag{5.19}$$

For two m-dimensional normal distributions, $s^{(t)}$ becomes either

$$s_H^{(t)}(\mathcal{N}_1, \mathcal{N}_2) = \exp\left\{-\frac{t(1-t)}{2} D_M^2(\boldsymbol{\mu}_2, \boldsymbol{\mu}_1; t\,\Sigma_1 + (1-t)\,\Sigma_2) + \frac{1}{2}\ln\frac{|t\,\Sigma_1 + (1-t)\,\Sigma_2|}{|\Sigma_1|^t\,|\Sigma_2|^{1-t}}\right\} \tag{5.20}$$

or in case of equal covariance matrices: $s^{(t)}(\mathcal{N}_1, \mathcal{N}_2) = \exp\left\{-\frac{t\,(1-t)}{2} D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \Sigma)\right\}$.

Chernoff and Bhattacharyya coefficients. For $t = 1/2$, the Hellinger similarity becomes the Bhattacharyya symmetric coefficient [138]. The Bhattacharyya distance is then given as:

$$d_{BH}(F_1, F_2) = -\log\left(s_H^{(1/2)}(F_1, F_2)\right). \tag{5.21}$$

For two normal distributions, it becomes:

$$d_{BH}(\mathcal{N}_1, \mathcal{N}_2) = \frac{1}{8} D_M^2(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2; \frac{1}{2}(\Sigma_1 + \Sigma_2)) + \frac{1}{2} \log \frac{|\frac{1}{2}(\Sigma_1 + \Sigma_2)|}{|\Sigma_1|^{1/2} |\Sigma_2|^{1/2}}. \tag{5.22}$$

The Bhattacharyya distance is a special case of the Chernoff distance [138]:

$$d_{CH}^{(t)}(F_1, F_2) = -\log\left(s_H^{(t)}(F_1, F_2)\right). \tag{5.23}$$

The Chernoff and Bhattacharyya distances are important in the classification area since they provide upper bounds on the Bayes error of two classes described by normal distributions [97, 138].

Discrete probability distributions

Let us consider n objects, described by m categorical variables and belonging to two groups. The groups are then treated as separate distributions. Let $p_i^{kj} = n_i^{kj}/n$ be the relative frequency, where n_i^{kj} is the number of instances belonging to the j-th category present of the k-th variable for the i-th group, where $i = 1, 2$. Let $\mathbf{p}_i = [p_i^{11} .. p_i^{1c_1} p_i^{21} .. p_i^{2c_2} .. p_i^{mc_m}]$ and c_k be the number of different categories for the k-th variable and $c = \sum_{k=1}^{m} c_k$. The inter-group distance can be computed as follows:

$$d^2(\mathbf{p}_1, \mathbf{p}_2) = \sum_{k=1}^{m} \sum_{j=1}^{c_k} \frac{(p_1^{kj} - p_2^{kj})^2}{\frac{1}{2}(p_1^{kj} + p_2^{kj})}. \tag{5.24}$$

Another possibility is to extend the Mahalanobis distance by replacing the continuous variables by the categorical ones. If C is a $c \times c$ sample covariance matrix, such a measure is given by:

$$D_{M-cat}^2(\mathbf{p}_1, \mathbf{p}_2) = (\mathbf{p}_1 - \mathbf{p}_2)^T C^{-1} (\mathbf{p}_1 - \mathbf{p}_2). \tag{5.25}$$

The affinity coefficient can be used as well. It is related to the Hellinger similarity, formula (5.2), and it measures the resemblance between two categorical or modal features, or two histograms. Let $p_i^{kj} = n_i^{kj}/n$, as above. Thus, those frequencies generate a discrete probability distribution. The affinity between two frequency distributions for the variable f_k is expressed as $a_{f_k} = \sum_{j=1}^{c_k} \left(p_1^{kj} p_2^{kj}\right)^{1/2}$. This leads to the affinity dissimilarity between the groups defined as:

$$d_{\text{aff}}(\mathbf{p}_1, \mathbf{p}_2) = 1 - \sum_{k=1}^{m} w_k a_{f_k} \tag{5.26}$$

where w_k are appropriate weights.

5.3 Dissimilarity measures between sequences

Let A be an alphabet, i.e. a finite collection of symbols, also called letters, from which sequences or strings are composed.Let $\mathbf{s} = s_1 s_2 \ldots s_n$ be a sequence of letters from A. An empty word is denoted by ε and it has a null length. Such strings are used in the pattern recognition and machine learning areas for encoding objects of relatively homogeneous structure. Here, we will describe the most common distance measures. For a general framework, see [248] and for a universal definition in terms of Kolmogorov complexity, see the work of Vitányi and his colleagues [18, 246].

Hamming distance. This is one of the most simple measures: for two sequences of equal length, it counts the symbol positions in which they differ; see also Table 5.1. Without loss of generality, let $\mathbf{s}$ and $\mathbf{t}$ be binary sequences. Then, the Hamming distance can be expressed as $d_{Ham}(\mathbf{s},\mathbf{t}) = \sum_k \mathcal{I}(s_k \neq t_k)$. It is not a flexible measure, since it assumes sequences of a fixed length. In many problems, however, the sequences have a variable length and, moreover, there might be no fixed correspondence between their symbol positions. A small shift of the position in one of the two nearly identical sequences can lead to exaggerated values in the Hamming distance.

Fuzzy Hamming distance. A fuzzy Hamming distance [36] has been proposed to make the Hamming distance be sensitive to local neighborhoods. This is a type of an edit distance for sequences of equal length. Edit distance relies on transforming one sequence into another by using the so-called edit operations. The following edit operations are introduced: insertion, deletion and shift, with the costs $c_{\text{ins}}, c_{\text{del}}$ and c_{sub} assigned to them, correspondingly. The shift operation allows to transform a 1-bit in one string to the nearest 1-bit in the other string at smaller costs than by both deletion and insertion. The operations are now used to transform one string into another and the resulting dissimilarity d_{fHam} is computed by adding up the costs of the operations such that it has a total minimal cost. The fuzzy Hamming distance is metric [36] if $c_{\text{del}} = c_{\text{ins}}$ and for the absolute size of a shift $h \geq 0$, $c_{\text{sub}}(h) \geq 0$ and $c_{\text{sub}}(h) = 0$ iff $h = 0$, $c_{\text{sub}}(h)$ increases monotonically and it is concave on the integers.

Levenshtein distance. The most popular edit distance is the Levenshtein distance [244, 411], expressing a local similarity between the sequences of arbitrary lengths. It is based on three edit operations: insertion, deletion and substitution. The costs $c_{\text{ins}}, c_{\text{del}}$ and c_{sub} are associated to each of them, correspondingly, giving rise to a weighted version of this distance. In the edit distance, $c_{\text{sub}} > c_{\text{del}} + c_{\text{ins}}$, meaning that a deletion of a and an insertion of b are preferred to the substitution of a by b. If all the costs are such that a single one is not larger than the sum of two other costs, then d_L is a metric [48]. Similarly to d_{fHam}, the weighted Levenshtein distance d_L is determined by the minimal total cost related to the operations transforming a sequence $\mathbf{s}$ into $\mathbf{t}$. (Note that the solution might not be unique). Assuming that such a transformation requires n_{sub} substitutions, n_{ins} insertions and n_{del} dilations, d_L is expressed as:

$$d_L(\mathbf{s},\mathbf{t}) = \min_{n_{\text{sub}},n_{\text{ins}},n_{\text{del}}} (n_{\text{sub}}\, c_{\text{sub}} + n_{\text{ins}}\, c_{\text{ins}} + n_{\text{del}}\, c_{\text{del}}). \tag{5.27}$$

The traditional Levenshtein distance with all the costs equal to one is often considered. However, d_L depends on the lengths of the compared sequences. To make it independent of the lengths, a normalization can be used, yielding the normalized weighted distance [262, 410]:

$$d_{nL}(\mathbf{s},\mathbf{t}) = \frac{d_L(\mathbf{s},\mathbf{t})}{\max\{n,m\}}. \tag{5.28}$$

However, since the triangle inequality may not hold[3], d_{nL} is quasimetric.

Other distance measures. Two sequences can also be considered based on the common longest prefix, suffix or just a subsequence. Assume we are given two sequences $\mathbf{s}$ and $\mathbf{t}$ of the length n and $m \leq n$, respectively. Then, the distance can be defined as $d(\mathbf{s},\mathbf{t}) = m + n - 2\,|\text{common}(\mathbf{s},\mathbf{t})|$. The problem of finding of the common longest subsequence is complementary to determining the edit distance. It can also be solved by the use of dynamic programming. See also [380]. A survey to approximate string matching can be found in [281].

[3] Consider three sequences $\mathbf{s}_1$, $\mathbf{s}_2$ and $\mathbf{s}_3$ of 9, 10 and 15 zeros, respectively. Assume all the costs equal one. Then $d_{nL}(\mathbf{s}_1,\mathbf{s}_2) = \frac{1}{10}$, $d_{nL}(\mathbf{s}_2,\mathbf{s}_3) = \frac{1}{3}$ and $d_{nL}(\mathbf{s}_3,\mathbf{s}_1) = \frac{3}{5}$. Clearly, $\frac{1}{10} + \frac{1}{3} < \frac{3}{5}$, hence the triangle inequality is violated.

5.4 Dissimilarity measures between sets

Dissimilarities can also be considered between two closed and bounded subregions of a (Euclidean) space, sets of points or elements. Let us first formally introduce the Hausdorff distance [218, 318].

Hausdorff metric. Let (X,ρ) be a metric space and $\mathcal{C}(X)\subseteq X$ be a space of nonempty, closed and bounded subsets of X. Let $N_\varepsilon(A)=\bigcup_{x\in A}B_\varepsilon(x)$ be the cover of $A\in X$ by open ε-balls $B_\varepsilon(x)=\{y\in X:\rho(x,y)<\varepsilon\}$. Since $B_\varepsilon(x)$ is a neighborhood of x, Theorem 2.33, then $N_\varepsilon(A)$ is the neighborhood of A according to Def. 2.4. The Hausdorff distance between A and B is defined as the smallest ε-neighborhood of A which covers B and the other way around; see also Fig. 5.1. On the other hand, the directed Hausdorff distance between A and B, $d_H^{\triangleright}(A,B)$ can be expressed as the maximum taken over the collection of minimum distances between elements of A and the set B. Then, the Hausdorff distance $d_H(A,B)$ is the maximum over the two directed distances. Formally:

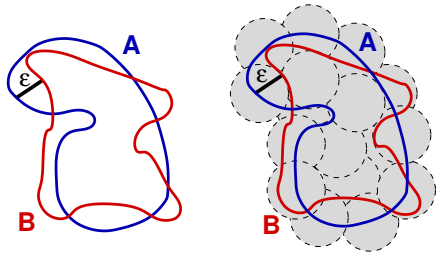


Fig. 5.1: An illustration of the Hausdorff distance; $d_H(A,B)=\varepsilon$.

Def. 5.3 (Hausdorff distance) In a (semi-)metric space (X,ρ), the Hausdorff distance with the base ρ is defined for all $A,B\in\mathcal{C}(X)$ in one of the following ways:

(1) $d_H(A,B)=\inf_{\varepsilon>0}\{A\subset N_\varepsilon(B)\ \&\ B\subset N_\varepsilon(A)\}$.

(2) $d_H(A,B)=\max\{d_H^{\triangleright}(A,B),d_H^{\triangleright}(B,A)\}$, where $d_H^{\triangleright}(A,B)=\sup_{a\in A}\inf_{b\in B}\rho(a,b)$.

If the domain of $d_H^{\triangleright}$ is restricted, then supremum becomes maximum and infinium becomes minimum, namely $d_H^{\triangleright}(A,B)=\max_{a\in A}\min_{b\in B}\rho(a,b)$.

Corollary 5.4 The two formulations of the Hausdorff distance given in Def. 5.3 are equivalent.

Proof. We start from definition (1) and by equivalent transformations, the formulation of definition (2) is reached. $\inf_{\varepsilon>0}\{A\subset N_\varepsilon(B)\}=\inf_{\varepsilon>0}\{\forall_{a\in A}\ a\in N_\varepsilon(B)\}=\inf_{\varepsilon>0}\{\forall_{a\in A}\ a\in\bigcup_{b\in B}(x:\ \rho(x,b)<\varepsilon)\}=\inf_{\varepsilon>0}\{\forall_{a\in A}\ \inf_{b\in B}\rho(a,b)<\varepsilon\}=\{\sup_{a\in A}\inf_{b\in B}\ \rho(a,b)\}=d_H^{\triangleright}(A,B)$. Based on this, we have: $d_H(A,B)=\inf_{\varepsilon>0}\{B\subset N_\varepsilon(A)\ \&\ A\subset N_\varepsilon(B)\}=\max\{\inf_{\varepsilon>0}\{B\subset N_\varepsilon(A)\},\inf_{\varepsilon>0}\{A\subset N_\varepsilon(B)\}\}=\max\{d_H^{\triangleright}(A,B),d_H^{\triangleright}(B,A)\}$, which finishes the proof. ■

Theorem 5.5 If (X,ρ) is a metric (semimetric) space, then d_H is metric (semimetric).

Proof. First, we will prove that if ρ is semimetric, then d_H is semimetric. We will make use of the second formulation in Def. 5.3. Since for all $a\in A$, $\inf_{a\in A}\rho(a,a)=0$, then $d_H(A,A)=0$. The max operation is symmetric, so d_H is symmetric. Let $A,B,C\in\mathcal{C}(X)$. Let $\rho(a,B)=\inf_{b\in B}\rho(a,b)$. If $a\in A$, then there exists b such that $\inf_{b\in B}\rho(a,b)\le\sup_{a\in A}\rho(a,B)=d_H^{\triangleright}(A,B)\le d_H(A,B)$. Given such b, we can also write $\rho(b,C)=inf_{c\in C}\ \rho(b,c)\le d_H(B,C)$. By applying the triangle inequality to ρ, for each $a\in A$ the following holds: $\rho(a,C)\le\rho(a,B)+\rho(B,c)\le d_H(A,B)+d_H(B,C)$. Since the above inequality remains true for all $a\in A$, then $d_H^{\triangleright}(A,C)=\sup_{a\in A}\rho(a,C)\le d_H(A,B)+d_H(B,C)$. Because the ordering of A and C is arbitrary, we also know that $d_H^{\triangleright}(C,A)\le d_H(A,B)+d_H(B,C)$. Hence, $d_H(A,C)\le d_H(A,B)+d_H(B,C)$.

To prove that d_H is metric if ρ is, the definiteness axiom, Def. 2.30, should be considered. Let $d_H(A,B)=0$. Then $d_H^{\triangleright}(A,B)=d_H^{\triangleright}(B,A)=0$. Consequently, for each $a\in A$, $\inf_{b\in B}\ \rho(a,b)=0$. This means that every neighborhood of a contains an element from B. We know that $a\in(^-B)=B$, since B is a closed set. Since this holds for all $a\in A$, then $A\subset B$. By symmetry of our definition, we also get $B\subset A$. Thus, $A=B$. ■

The Hausdorff distance is invariant w.r.t. a transformation only if the base metric is invariant. Thereby, every isometry in the base metric is an isometry in the Hausdorff metric. Moreover, two sets are within the Hausdorff distance d from each other if any point of one set is within the distance

d from some point of the other set. Such a distance is sensitive to single outliers. For instance, think of a case where a point a is at some large distance d_a to all points in the set A. Then, $d_H(A,B)=d_a$ is determined by this point. Therefore, generalizations of the Hausdorff distance have been considered, which are more robust against outliers or noise.

Variants of the Hausdorff distance. Let (X,ρ) be a metric space (usually Euclidean) and $\mathcal{C}(X)\subseteq X$ be a space of nonempty, closed and bounded subsets of X. Let $A,B\in\mathcal{C}(X)$ be sets of n_A and n_B elements, correspondingly. The distance between an element $a\in A$ and the set B can be defined as:

$$d(a,B)=d(\{a\},B)=\min_{b\in B}\rho(a,b). \tag{5.29}$$

The directed dissimilarities between two sets can be then found as [93]:

$$\begin{aligned} d^{\triangleright}_{min}(A,B) &= \min_{a\in A} d(a,B), & d^{\triangleright}_{0.5}(A,B) &= M^{0.5}_{a\in A} d(a,B),\\ d^{\triangleright}_{max}(A,B) &= \max_{a\in A} d(a,B), & d^{\triangleright}_{0.75}(A,B) &= M^{0.75}_{a\in A} d(a,B),\\ d^{\triangleright}_{avr}(A,B) &= \tfrac{1}{n_A}\textstyle\sum_{a\in A} d(a,B), & d^{\triangleright}_{0.9}(A,B) &= M^{0.9}_{a\in A} d(a,B), \end{aligned} \tag{5.30}$$

where $M^{s}_{a\in A}$ is k-th ranked distance such that $k=s\,n_A$. For instance, for $s=0.5$, $M^{0.5}_{a\in A}$ becomes the median of the distance sequence $d(\mathbf{x},Y)$ and for $s=0.75$, this is the upper quartile.

Since the values $d^{\triangleright}(A,B)$ and $d^{\triangleright}(B,A)$ are usually not identical, the symmetry is imposed by applying one of the following operators: $f_{min}(x,y)=\min\{x,y\}$, $f_{max}(x,y)=\max\{x,y\}$, $f_{avr}(x,y)=\frac{1}{2}(x+y)$ or $f_{wavr}(x,y)=\frac{1}{n_A+n_B}(n_A\,x+n_B\,y)$. Combining them with the distances defined by (5.30), 24 symmetric dissimilarity coefficients can be obtained, which all but one are non-metric. Two of them are of a significant importance, especially for the purpose of object matching in binary images [93], namely the Hausdorff distance (the only metric), already introduced in Def. 5.3, and the modified Hausdorff distance. The latter, although non-metric, has been found useful [93] and more robust against outliers. Also other variants obtained by replacing the max operation in the Hausdorff measure by a k-th rank are often less noise sensitive [201].

Def. 5.6 (Modified Hausdorff) In a (semi-)metric space (X,ρ), the modified Hausdorff distance with the base ρ is defined for all $A,B\in\mathcal{C}(X)$ as:

$$d_{MH}(A,B)=\max\{d^{\triangleright}_{avr}(A,B),d^{\triangleright}_{avr}(B,A)\},\ \text{where } d^{\triangleright}_{avr}(A,B)=\frac{1}{n_A}\sum_{a\in A}\min_{b\in B}\rho(a,b). \tag{5.31}$$

A Hausdorff-like distance can also be defined for fuzzy sets; see [60, 61] for details.

5.5 Dissimilarity measures in applications

There is a large arsenal of various measures developed for retrieval, clustering and classification purposes. Here, we will only mention some of them, limiting ourselves to the application areas, where objects under considerations are represented in a feature space, by shapes or by images.

In an information-theoretic sense, a universal definition of similarity, applicable to the domains which have a probabilistic model, was proposed by Lin [248]. It is based on the common sense observation that the similarity between two objects is connected to their commonality and their difference and that two identical objects reach the maximum similarity. This leads to the following assumptions [248]:

1. The commonality between A and B is measured by $I(\mathrm{com}(A,B))$, where I is the amount of information, usually the negative logarithm of the probability of the event it refers to.
2. The difference between A and B is measured by $I(\mathrm{desc}(A,B))-I(\mathrm{com}(A,B))\geq 0$, where $\mathrm{desc}(A,B)$ is a proposition that describes what A and B are.

3. The similarity is a function $f : \mathbb{R}_+^0 \times \mathbb{R}_+ \to [0,1]$ of commonalities and differences given as $\text{sim}(A,B) = f\left(I\left(\text{com}(A,B)\right), I\left(\text{desc}(A,B)\right)\right)$, such that $f(x,x) = 1$ and $f(0,y) = 0$.
4. The overall similarity of two objects is a weighted average of their similarities computed from different perspectives.

The similarity derived from these assumptions is measured as the ratio between the amount of information needed to state the commonality of two objects and the amount of information needed to describe them. It is given as $\text{sim}(A,B) = \log P\left(\text{com}(A,B)\right) / \log P\left(\text{desc}(A,B)\right)$. In [248] Line presents how this general definition is applied to a number of domains, resulting in a similarity between strings, words or concepts in taxonomy.

Another universal definition of a metric between sequences was proposed by Li et al. [246]. This measure is based on the notion of Kolmogorov complexity. For some other considerations, see [18].

Feature type data. For data represented in feature spaces, a number of distance measures have been designed to account for the distribution of points in local neighborhoods. Such distances are then used by the k-NN rule or by some variant of locally weighted learning; see Atkeson et al. [5] for a survey of methods. We will mention a few.

Friedman [136] proposed some techniques for flexible metric construction. These methods are based on a recursive partitioning strategy to adaptively shrink and shape rectangular neighborhoods around the test point. Also Hastie and Tibshirani [190] developed an adaptive NN rule that uses local discriminant information to modify the neighborhoods appropriately. The distance metric is the square Euclidean distance weighted by a product of suitably weighted between- and within-sum-of-squares matrices. They show that this metric approximates a chi-squared distance between true and estimated posterior probabilities for spherical Gaussian classes. Generalizing both previous approaches, Domeniconi et al. [89] estimated a flexible metric for computing neighborhoods based directly on the Chi-squared distance. The property of the neighborhoods is such that they are elongated along less informative features and compact along most influential ones. Also Avesani et al. [6] proposed two metric measures for the NN rule: a local asymmetrically weighted similarity metric and a minimum risk metric based on a probability estimation that minimizes the risk of misclassification. They found experimentally that the 1-NN rule based on their measures performs well. Lowe [251] introduced a variable kernel classifier based on a similarity metric, by combining the k-NN rule with smooth weighting defined by the Gaussian kernels. The Gaussian kernels are based on a weighted Euclidean distance, where the weights are learned in the cross-validation procedure.

All these approaches can be encompassed by a general framework based on similarities computed between the features, as proposed by Duch et al. [94–96]. Such a model involves the steps of selecting distinctive features, weighting them and scaling appropriately, and computing a distance suitable for the feature type and the problem at hand.

Text. Many of the information retrieval models make use of statistical properties of text [259]. For a collection of text documents, a vocabulary set is often chosen for the indexing purposes. Text documents are then represented as vectors of term weights for every term from the vocabulary set. The term weight is often proportional to the frequency of occurrence within the document and inversely proportional then number of documents the term occurs in. The similarity measure between the documents is often an appropriately weighted variant of a cosine similarity which measures the cosine of the angle between the document vectors or an l_p distance [385]. Many weighting schemes can be used, as well as binary measures focusing on the word occurrences see proceeding of the SIGIR conferences [364]. Another possibility are also information theoretic measures as described in [18, 246, 248]. When document collections are described by graphs, various graph dissimilarity related to the maximum common subgraph [50, 51] as well as to the graph union or minimum common supergraph [340], can be used.

Shapes. In computer vision, image processing and pattern recognition areas, many shape description techniques have been developed for both quantitative and qualitative measurements. Such descriptions mostly rely either on segmentation followed by external characteristics of the resulting binary shape defined by spatial arrangements of elements such as edges and junctions, or on internal shape characteristics, as texture or intensity-based features, in the given grey-level image. For a general introduction into shape description methods, see the book by Costa and Cesar [69].

Here, we are interested in the comparison of objects, hence in measures of their similarity. Many such measures exist, both general and application-specific, mostly developed for solving pattern matching problems. A typical example of a dissimilarity-oriented pattern matching relies on finding geometric transformations (from a specified class) of one pattern (shape, contour, image) into another one such that a predefined cost is minimized. For a survey of shape matching approaches, see [408] and for some similarity measures and algorithms, see [407].

For the purpose of matching of binary images (hence also contours), variants of Hausdorff distances can be used, as described in section 5.4. For some practical considerations, see [93, 201]. Since these measures are in fact measures between sets of points, some further extensions can be found in [117]. Also mathematical expressions for the distance between 2D point sets with known correspondences were suggested in [417]. They are invariant to either affine transformations or similarity transformations of the sets. First, images are normalized and aligned by the use of affine transformations and next, the square Euclidean distances between the points in images are computed. To our judgment, this is similar in formulation to the Procrustes analysis [37, 72].

A more general metric distance measure, the so-called *absolute difference* was introduced by Hagedoorn and Veltkamp [183]. This measure is invariant under affine transformations and deals well with objects having multiple connected components. It is robust against perturbation and occlusion.

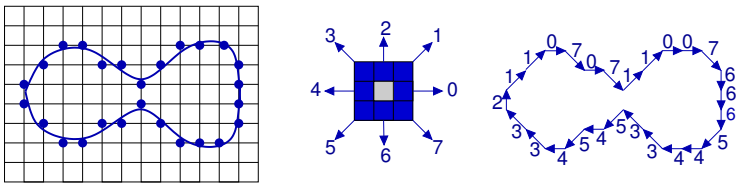
Basri et al. [9–11] tried to capture human judgments of similarity. For instance, in [10], the dissimilarity between image contours is studied as a cost of matching by summing up the costs of local deformations that reflect the differences between two contours. A cost function is proposed which depends on the local curvature and obeys the constraints of continuity, metric properties and invariance under some classes of transformations. The cost function should also grow with the increase of bending or stretching, but bending should be less costly at a point of high curvature.

Some other ideas of curve matching can be found in [144], the definition of elastic distance is considered in [423, 424] and the use of deformable templates for handwritten digits in [207].

Belongie et al. [16, 17, 274] developed a shape descriptor, the shape context, along with a framework for deformable matching. The shape context at a particular point location on the shape is defined by the histogram of the relative log-polar coordinates of all other points. Since corresponding points of two different shapes have similar characteristics, the alignment of shapes is simplified. The overall distance is given as the weighted average of three contributions: sum of the best shape matching costs, appearance distance due to the brightness differences and the bending energy.

Recently, Sebastian, Klein and Kimia [361] have proposed a novel approach to the alignment between two curves, which further serves for the definition of the dissimilarity between them. They reported that this method is robust under a variety of affine transformations, as well as viewpoint variations and small deformation and that it can be applied to object recognition problems. Algorithmically, the alignment is solved by the dynamical programming [15].

Another possibility of comparing two binary shapes is by the use of a distance transformation. It is the operation on a binary image which transforms it into a gray-level image, a distance map, where non-object pixels have a value corresponding to the distance to the nearest object pixel. Objects can be shapes, but also curves, edges or points. Matching relies on positioning the template shape at



(a) (b)

Fig. 5.2: Chain code representation. (a) Result of resampling. (b) Chain code based on the 8-connectivity.

various locations of the distance map. The matching cost, hence the dissimilarity between the object shape and the template, is determined by the pixel values of the distance map which lie under the data pixels of the template. The target is considered as detected when e.g. the average distance value is below a chosen threshold. The most common distance is Euclidean, but due to its computational cost, often chamfer distance, as its best approximation, is used; see the work of Borgefors [38]. An example of shape matching using chamfer distance transform can be found in [142, 143]. It covers the detection of arbitrary-shaped objects, either parametrized or not, like pedestrian contours.

Recently, Thayananthan et al. [395] have compared the methods of shape context and chamfer matching for the purpose of object detection as described by a contour. They found out that in case of cluttered scenes chamfer matching, based on a number of templates, is more robust than the shape context approach.

Shapes can also be described from the structural point of view. Then, a chain code [133] represents a digital boundary as a sequence of direction vectors based on the 4- or 8-connectivity principle; see Fig.5.2. In general, it is not unique, since it depends on the starting point. However, given a starting point, it reconstructs a shape perfectly. Unfortunately, chain codes become very long for complex objects, but more importantly, they reflect all the noise present on the boundary e.g. due to small disturbances. Still, for a comparison of two shapes, their chained codes can be compared. Since their starting points can be arbitrary, the matching should be performed between all their cyclic permutations. Let $\mathbf{s}$ and $\mathbf{t}$ be the chain codes of the two contours. Let S and T represent sets of all cyclic permutations of $\mathbf{s}$ and $\mathbf{t}$, respectively. Then, the comparison of two chain codes is based on the weighted Levenshtein distance as follows: $d_{\text{chain}}(\mathbf{s},\mathbf{t}) = \min\{d^{\triangleright}(\mathbf{s},\mathbf{t}), d^{\triangleright}(\mathbf{t},\mathbf{s})\}$, where $d^{\triangleright}(\mathbf{s},\mathbf{t}) = \min_{\mathbf{s}^* \in S, \mathbf{t}^* \in T} d_{wL}(\mathbf{s}^*,\mathbf{t}^*)$ is a directed distance. In this way, d_{chain} is robust against rotation of shapes, however, not against scaling.

Alternatively, a contour can be represented as a sequence of points $s = (x_1, y_1) \ldots (x_m, y_m)$ in a 2-dimensional space, resampled if necessary such that the distances between any consecutive pair of points are identical. Then, a string $z = \mathbf{z}_1 \ldots \mathbf{z}_m$, describing a contour, is derived such that $\mathbf{z}_i$ is the direction vector pointing from (x_i, y_i) to (x_{i+1}, y_{i+1}). The distance between the strings is an edit distance with fixed insertion and deletion costs and some substitution cost. Different substitution costs, e.g. based on an angle or the Euclidean distance between vectors lead to different distance measures; see [47, 48]. It is claimed [48] that such an approach has a number of advantages such as higher angular resolution, robustness to shape distortion under rotation and invariance under scaling.

Also Fourier descriptors [307, 428] for closed contours can be found for which some distance measures can be defined, such as a Minkowski distance.

A structural description of complete shapes is based on a coarse description of the geometric relations between the parts that compose them. Similarity between the shapes can be, therefore, evaluated as a metric edit distance between shock graphs representing the shapes as advocated by Kimia and his colleagues [218, 359, 360, 362]. This measure is computed as the optimal cost of the deformation path between two curves and it is robust against small deformations, occlusions and boundary disturbances. Also, a comparison between retrieval based on shock graphs (structural approach) and curve matching (metric approach) is presented by Sebastian and Kimia [357, 358].

Some other approaches based on the representation of shapes by medial axes can also be found in [249, 399, 429].

Finally, the statistical properties of the object's shape can also be used for comparison. This means that shape information can be encoded by moment descriptors, which describe center of mass, elongation aspects and overall orientation. Also, other, more specific features w.r.t. the overall shape can be found, such as: perimeter, area, boundary straightness, curvature in terms of the zero-crossing of the curvature around the shape contour or bending energy. All these quantitative features may be used to construct a dissimilarity measure as e.g. given in Table 5.2.

Histograms and spectra. Emission and reflectance spectra become more popular for the identification of certain materials, e.g. types of plastics or minerals and rocks. Also autofluorescence is emerging as a useful tool for the detection of cancer e.g. in oral cavity or in the bronchi. It relies on the spectroscopy of the tissues of interest. The measurements are usually performed on healthy and diseased tissues (in various stages of cancer) at several excitation wavelengths. The emission spectra are then analyzed to support the diagnosis of a doctor.

Histograms and spectra can be interpreted in the probabilistic framework, where their normalized versions are considered as probability distributions. This allows one to use divergence measures or general measures between distributions, where some them are mentioned in section 5.2. Since the structure of such data is organized by the underlying factor, such as the order of bins, wavelength or time, it might be beneficial to incorporate such knowledge into the measure. This is somewhat possible e.g. by computing the difference, such as the l_p-distance, between the approximated derivatives of the histograms or spectra [107, 286, 287, 305]. For instance, the distance between the first-order derivatives emphasizes the difference in positions between the local minima and maxima of the histograms. Also the distance between the cumulative histograms can be used as well e.g. as we used for the comparison of chromosome band profiles in [293].

Images. Assume that grey-level images are represented as vectors in a space. Simard et al. [365, 366] proposed a *tangent distance*, which is locally invariant to any set of chosen transformation (such as rotation and thinning) and is relatively cheap to compute. It was found to be especially effective in the domain of handwritten digit recognition [365]. When an image is transformed (e.g. scaled and rotated) with a transformation that depends on some parameters (like the scaling factor and rotation angle), the set of all transformed patterns create a manifold of a dimension at most equal to the number of free parameters in the vector space. The distance between two image patterns can be now defined as the minimum distance between their respective manifolds and by this invariant w.r.t. the considered transformations. Such a distance is hard to compute. The compromise is offered by the tangent distance which is defined as the minimum distance between the tangent subspaces that best approximates the non-linear manifolds. See [365] for details.

Since two gray-value images can be considered as fuzzy sets (by rescaling them to the range $[0, 1]$), for their comparison the fuzzy Hausdorff (or modified-Hausdorff) distance can be used. Also binary images can be regarded as fuzzy sets in the following manner: white pixels have zero membership values and a black pixel takes a value of $k/(K^2 - 1)$ if it has k black neighbors in its $K \times K$ neighborhood. In this way, noisy black pixels will have either zero or very small membership value. If the binary images are converted to the fuzzy sets as described, Chaudhuri [61] has reported that the noise has much less effect on the fuzzy Hausdorff distance than on the original Hausdorff distance between binary images. Consequently, the fuzzy Hausdorff distance is relatively robust to noise.

On the other hand, grey-value images can be interpreted from the probabilistic point of view, e.g. as bivariate histograms. This allows one to use various divergence measures or general measures between distributions, see also section 5.2. Since the intensity of the images might be different, some preprocessing of normalizing the intensities might be crucial.

The description of images can also be simplified to univariate histograms, for instance intensity histograms. Then, the distance between two images A and B can be computed e.g. based on the intersection between two intensity histograms with b bins is $d_I(A,B) = 1 - \frac{\sum_{i=0}^{b-1} \min(h_i(A), h_i(B))}{\#\text{ pixels}}$, where $h_i(A)$ describes the number of pixels whose intensity equals to the value assigned in i-th bin. Note that the intersection is the estimation of the Bayes error, i.e. the overlap between two probability density functions $P(A)$ and $P(B)$ approximated by histograms. An extension of such a measure has been proposed by Cha and Srihari [57], which takes into account similarity of both overlapping and non-overlapping parts.

There are a number of dissimilarity measures to support the content-based image retrieval. For a brief summary, see e.g. [404]. In the probabilistic framework, usually dissimilarities defined between distributions such as Kullback-Leibler divergence, Bhattacharyya distance, Mahalanobis distance, are used. For a brief analysis of their inter-relations, see [405]. Also, the earth mover's distance (EMD) [326] was designed to evaluate dissimilarity between two distributions based on the so-called ground distance measure between single features. Loosely speaking, one distribution can be interpreted as a mass of earth spread in space, while the other distribution as a collection of holes in the same space. Then, the EMD defines the least amount of work needed to fill the holes with earth. Computing the EMD is based on a solution to the transportation problem [326]. This measure has been successfully applied for an evaluation of texture and color similarities in images [324–326]. It has, however, a rigorous probabilistic interpretation, as shown by [245].

In the probabilistic framework, also Puzicha and colleagues empirically investigated some dissimilarity measures for the purpose of texture segmentation and image retrieval [309] and for color and texture [311]. In both papers, images are compared by distribution-based dissimilarity measures, of Gabor coefficients in the filtered images in the first paper, and between histograms in the latter.

An approach to incorporate human similarity assessment in the dissimilarity measure is based on the extensions of the Tversky's model [401] by fuzzy logic; see the work of Santini [332–334].

5.6 Discussion and conclusions

This brief overview of similarity and dissimilarity measures indicates not only their variability, but also their different origins and the principles lying underneath them. The use of dissimilarity (proximity) is especially popular in computer vision and pattern matching applications, information retrieval and the evaluation of human judgments. In the pattern recognition area, it is widely accepted to use the k-nearest neighbor rule (usually considered for a given feature representation), at least as a reference method when solving a classification task. Still, more and more attention is devoted to the assessment of dissimilarity as a natural means of comparison of objects. For instance, in computer vision, Edelman recognized the importance of proximity by stating that 'representation is representation of similarities' [115]. He advocated the use of dissimilarities in [114–116] and in his book [113].

The universality of a dissimilarity lies in the fact that it can be approached from both statistical and structural points of view. Conventionally, one tries to develop either a measure based on statistical, hence quantitative or metric, properties of object representations (examples are measures in feature spaces, between sets of points and probabilistic measures) or based on structural, hence qualitative, properties (examples are measures based on chain codes, graphs and trees). Various attempts have been made to combine these two research lines as addressed already by Fu [137]. They are, however, often hybrid in a sense that subproblems of a larger problem are tackled separately by either one on the other approach and the complete system is optimized part by part. The significance of finding new measures, unifying these two approaches was emphasized e.g. in [110, 413]. Also, a number

of researchers made an attempt to define universal or general dissimilarity measures, as for instance in the framework proposed by Duch [94–96], Griffiths [178], Lin [248] or Vitányi [18, 246].

Most of the dissimilarities are defined for the problem at hand. Still, there is a number of them which allows for the existence of some free parameters, like weights of particular contributions, to be learned (adopted) in the training (usually off-line) process. Assume that one deals with objects that possess such a structure, such as spectra, time-signal, images or text documents. A completely novel way of thinking, trying to unify the statistical and structural lines, has been promoted by Goldfarb. A dissimilarity measure is determined in a process of inductive learning realized by the so-called evolving transformation systems [153, 157, 160]. Such a system is composed of a set of primitive structures, basic operations that transform one object into another or which generate a particular object and some composition rules which permit the construction of new operations from the existing ones [155–157, 160, 161] (which is a structural contribution). The statistical component is is defined by the means of a dissimilarity. Since there are costs related to the operations, the dissimilarity is determined by the minimal total cost of transforming one object into another. In this sense, the operations play the role of features and the dissimilarity, dynamically learned in the training process, combines the objects into a class. How to realize that is an open issue.

A simpler approach is to first define a small set of fundamental structural detectors, yet general enough to be applicable to many problems, independent of a specific expert knowledge of the application. This means that such detectors work for the given measurement domain, e.g. spectra or images. The useful subpatterns should be then identified by the detectors when applied to the consecutive measurement values. The inter-relationships between the subpatterns should be captured in some relational intermediate representation (e.g. by a graph or by a string). These would be the basis for the matching process and the derivation of the final dissimilarity. The learning relies then on the learning of proper weights (contributions) assigned to the identified subpatterns such that the specified dissimilarity is optimal for the discrimination between the classes. The most simple example is the edit-distance between string descriptions of objects, however, more general approaches are needed to be developed. Note that one may also consider statistical feature extractors (such as wavelets or Gabor filters), which work on the consecutive measurements, to be the building blocks of the learned dissimilarity. How to learn such measures is open for a future research.

PART II

Practice

In theory, there is no difference between theory and practice, but in practice, there is a great deal of difference.

ANONYMOUS

Planning, observation, and conclusion
Gathering information, it behooves us.
Test and experiments bring about
True results without a doubt.

”EXPERIMENTS”, DELOIS SYKES

6. Visualization

... when you are describing
A shape, or sound, or tint,
Don't state the matter plainly,
But put it in a hint;
And learn to look at all things
With a sort of mental squint.

"POETA FIT, NON NASCITUR", LEWIS CARROLL

This chapter begins the experimental part, where dissimilarity data are practically analyzed. Here and in the subsequent chapters, we would like to present a systematic approach to such an analysis, so we start from the most basic questions concerning the data understanding. In order to gain some insights about the data, one usually uses tools to represent the data and their relations in some visual forms to be subjected to a human judgment. Therefore, we investigate a number of well-known visualization techniques and their usefulness for the dissimilarity data.

The most simple display of the dissimilarity relations is achieved by plotting a dissimilarity matrix as an intensity image, where the increase in pixel intensity corresponds to growing dissimilarity values (starting from black). If the data items are grouped, then possible clusters can be seen by dark rectangular areas. An example is given below:

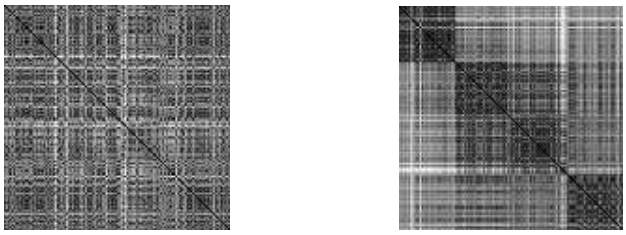


Fig. 6.1: Intensity images of a symmetric square dissimilarity representation. On the left, the order of objects is random, while on the right, the matrix is permuted such that the objects are grouped. This allows one to observe some cluster tendencies. The black diagonal line corresponds to the zero dissimilarities.

The dissimilarity relations can also be represented in a lower-, usually two- or three-dimensional space. This can be achieved by continuous spatial models, which rely on linear and nonlinear projections of the dissimilarities such that the configuration determined in an output space preserves most of some of the dissimilarities under the specified criterion. Usually, a Euclidean space is used, but other l_p-normed spaces can also be considered. The basic theory of spatial representations realized by the means of multidimensional scaling (MDS) techniques and more general models referring to pseudo-Euclidean spaces has been discussed in section 3.4. Here, for the completeness of the overall presentation, the basics of MDS are briefly recapitulated in section 6.1. The focus is, however, on some illustrative examples. Other types of spatial representations are obtained by nonlinear mappings concentrating e.g. on the preservation of dissimilarities in local neighborhoods or by approximating geodesic distances on a manifold. These and others alternative projection methods are briefly summarized in section 6.2.

Dissimilarity relations can also be represented by weighted, fully connected graphs, where the vertices correspond to individual objects and weights coincide with the given dissimilarity values. This

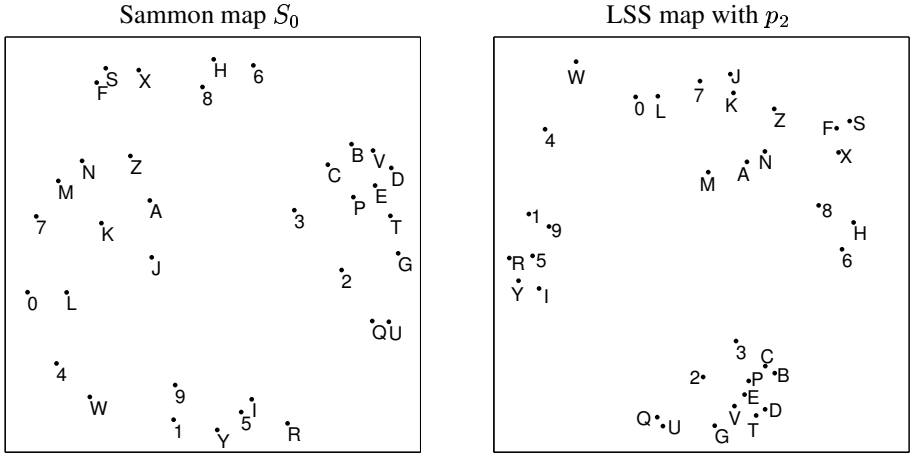


Fig. 6.2: The MDS maps, Sammon (left) and LSS (right), of auditory confusion measurements for letters and numerals. For the LSS map, $\hat{\delta}_{ij} = p_2(\delta_{ij})$, where p_2 is a second order polynomial.

can be further structured by tree models, usually understood in terms of the shortest paths between the vertices. Here, particularly important are the additive and ultrametric trees, which are discrete spatial models. They are widely used in data analysis, since they support the hierarchical clustering schemes and by this, they elevate the process of structuring of the data. The tree models are presented in section 6.3. The overall summary is given in section 6.4.

This chapter partly relies on our previous work [289, 295, 297–299], however, as such the study is mostly new (all the plots are made by us). Our contributions here refer to the presentations of nonlinearity of various variants of Sammon mappings, the formulation of the MDS techniques for missing data, and the explanations of generalization possibilities (adding new objects to the existing maps) for a number of projection algorithms, including the Sammon mappings. We also considered the use of LLE and Isomap for non-Euclidean dissimilarities and proposed to correct the local Gram matrices by adding a suitable constant. So, this chapter serves not only for the illustrative purposes, but tries to give some intuition on how the dissimilarity data can practically be explored.

6.1 Multidimensional scaling

Multidimensional scaling (MDS) [37, 72, 228] is a collection of techniques providing spatial representations of the objects by representing them as points in a low-dimensional space. This is achieved by (non)linear projections which aim to preserve all pairwise, symmetric dissimilarities between data objects. A spatial configuration is usually found in a Euclidean space, although any other l_p space ($p \geq 1$) can also be considered [37, 72]. Such a map is believed to reflect significant characteristics, as well as 'hidden structures' of the data. Therefore, objects judged to be similar to one another result in points being close to each other in a projected space. The larger the dissimilarity between two objects, the further apart they should be in the resulting map. In general, the dissimilarities describe the relations between objects, originally represented in a high-dimensional space (so, MDS is then treated as a dimension reduction technique), measured as costs of pattern matching in a template matching procedure, similarity between text documents or road distances, or just given, like human judgments.

Here, metric MDS methods [37] are used. They rely on quantitative dissimilarities, assuming that both the input data and the output configuration are metric. These techniques are realized by the linear methods of classical scaling or FastMap and the nonlinear methods of LSS and Sammon mapping variants, as already discussed in section 3.4. In brief, the goal of a metric MDS is to find a faithful representation X in a low-dimensional space such that the approximated distances $d_{ij}, i, j =$

$1, 2, \ldots n$ between n points match the disparities $\hat{\delta}_{ij}$ as well as possible. Disparities are functional dependencies (e.g. continuous monotonic functions) of the original dissimilarities, i.e. $\hat{\delta}_{ij} = f(\delta_{ij})$. Depending on the way the structure of the data is preserved, somewhat different techniques arise; see section 3.4 for details. Basically, the MDS methods, called *least squares scaling* (LSS) mappings, minimize a normalized version of the raw stress $\sum_{i<j}(\hat{\delta}_{ij} - d_{ij})^2$ as:

$$S_t^{LSS} = \frac{1}{\sum_{i<j} d_{ij}^{t+2}} \sum_{i<j} d_{ij}^t(X)\,(\hat{\delta}_{ij} - d_{ij}(X))^2, \quad t = \ldots, -2, -1, 0, 1, 2, \ldots$$

A similar technique is the Sammon mapping [329], originally proposed in the pattern recognition area as a method of nonlinear projection to a lower-dimensional space by optimizing the normalized square differences between the original and approximated distances. Assuming that $\hat{\delta}_{ij} = \delta_{ij}$, the variants of the Sammon stress functions are:

$$S_t = \frac{1}{\sum_{i<j}^{n} \delta_{ij}^{t+2}} \sum_{i<j} \left(\delta_{ij}^t\,(\delta_{ij} - d_{ij}(X))^2\right), \quad t = ..., -2, -1, 0, 1, 2, ...$$

Due to the obvious similarity to the LSS techniques, we account them as the MDS examples. Since the optimization of the Sammon stress functions is easier to define in gradient terms, Sammon mappings are preferred.

A standard MDS example is the reconstruction of a map of a country, given either the road or air distances between main cities; see [37, 258]. One important thing to realize about an MDS map is that the axes are, in themselves, meaningless. In case of a Euclidean space, additionally, the orientation of the projection is arbitrary, since any rotation of the configuration does not change the distances. This means that the MDS map of the cities need not be oriented such that north is up and east is right. What is important is the relative positions of the objects; the retrieved configuration may be mirrored or rotated. In general, MDS serves for exploration of the data, e.g. finding possible clusters, i.e. groups of points which are close together in the represented space. As an example, let us consider auditory confusion (dissimilarities) between 25 letters (all excluding 'O') and 10 Arabic numerals, computed by Lee [252]. The spatial Sammon and LSS maps obtained by us are shown in Fig. 6.2. Although they give somewhat different results (see section 3.4.2), the basic characteristics are the same. Clusters of similarly sounded letters or numerals can be clearly observed. For instance, we can justify the 'closeness' of 'I','5','1' and 'Y' since, when spoken, there is an obvious resemblance between their sounds.

Another purpose of the MDS is to find rules that would explain the observed dissimilarities and would help to describe the data structure in simple terms. This may be especially useful for data describing human judgments of similarity between objects. In such a case, interpreting an MDS configuration entails making a link between geometrical properties of such a map and prior knowledge about the objects represented as points [37]. By identifying points which are far apart, a line between them can be drawn, defining a perceptual axis, which describes a direction of a change between opposite or significantly different characteristics. This involves some data-guided speculation.

An example is based on human judgments of dissimilarities [252] between various sports. Our MDS representation is given in Fig. 6.3. To interpret why humans consider some sports to be more alike than others, we distinguish one perceptual axis, the

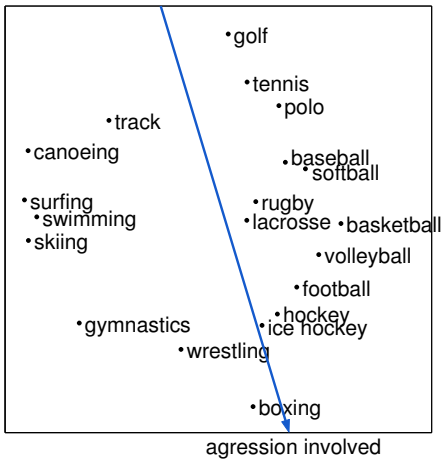


Fig. 6.3: The MDS map of human dissimilarity judgments on sport.

degree of aggression involved. The axis was added by us as a possible (not unique) interpretation, as a help to understand the relations better. Another possibility could be to tell the difference on the basis of whether a ball is used in a sport or not or the difference between a team sport and an individual sport. (Note that such axes do not need to be perpendicular.) A more scientific approach is to find a perceptual axis as a regression line in the projected space and then, given additional knowledge, attach a meaning to it.

Illustrative examples

To understand the properties of the MDS, one needs to study various linear and nonlinear techniques for dissimilarity data with some particular structure. The examples given below illustrate the difference between the linear and nonlinear methods.

Artificial data. The data sets describe 200 points lying on two circles, both of the radius of 1.0, in a 3D space. The circles are placed either on the planes parallel to the yz-plane with the distance of 1 or on two perpendicular planes. The data and the (non)linear metric MDS projections onto a 2D space are presented in Fig. 6.4. The projections are based on the 200×200 distance matrices, either Euclidean or city block distance. If the distance is Euclidean, then the mapped result is identical to the principal component projection (PCA) in the original 3D space [72]; see also section 3.3.1. If the l_1 distance is used, the output Euclidean distances approximate the original l_1 distances in a 3D space. Therefore, the 3D spatial representations are only shown for the l_1 distances, since for the Euclidean distances, the retrieved configurations are rotations of the original data. Note that since the l_1 distances have larger values than the Euclidean ones, the projected circles are also larger.

In case of classical scaling, two corresponding points from two parallel circles are mapped onto a single point in $2D$. It seems, therefore, that the data describe *one* circle. In case of perpendicular circles, one of them is reduced to a line. Therefore, some important information is lost in classical scaling, namely the existence of the second oval. Note, however, that FastMap, section 3.4.1, reveals two closed curves. Although for this example, it may seem that FastMap is superior to classical scaling in discovering the structure, it is not true for more complex, multi-class real data; see for instance Fig. 6.8. The nonlinear Sammon mapping outputs clearly illustrate two ovals of a similar shape. In general, nonlinear mappings reveal more 'hidden' structure in the data.

To illustrate the differences between the stress measures, as well as nonlinearity aspects of the projections involved, an artificial example of points lying on three non-crossing and non-parallel lines in a $5D$ space is considered with the Euclidean distance representation. Fig. 6.5 shows $2D$ linear MDS maps and $2D$ nonlinear MDS maps obtained by optimization of various S_t stresses. On the basis of either the classical scaling or FastMap results, one can draw a false conclusion that the data set represents three straight crossing lines in a higher-dimensional space. On the contrary, the Sammon maps suggest that the data consist of three *non-crossing* curves, but of course, not necessarily straight lines. Therefore, linear and nonlinear mappings are useful while studying them together, hence they complement each other.

The Sammon maps are ordered with respect to the nonlinearity involved in projections. By minimizing the S_{-2} stress, one focuses on preserving very small distances, by which local perturbations may appear as observed in Fig. 6.5, top row, third plot from the left. By optimizing the S_2 stress, on the contrary, one tries to preserve large distances, and as a result, the curves start to somewhat resemble straight lines. The stress S_0 keeps the balance between preserving small and large distances. The choice of a stress function depends on required geometric properties that an MDS map should have. When no preferences are given, our experience suggests that the stress S_0 can be recommended.

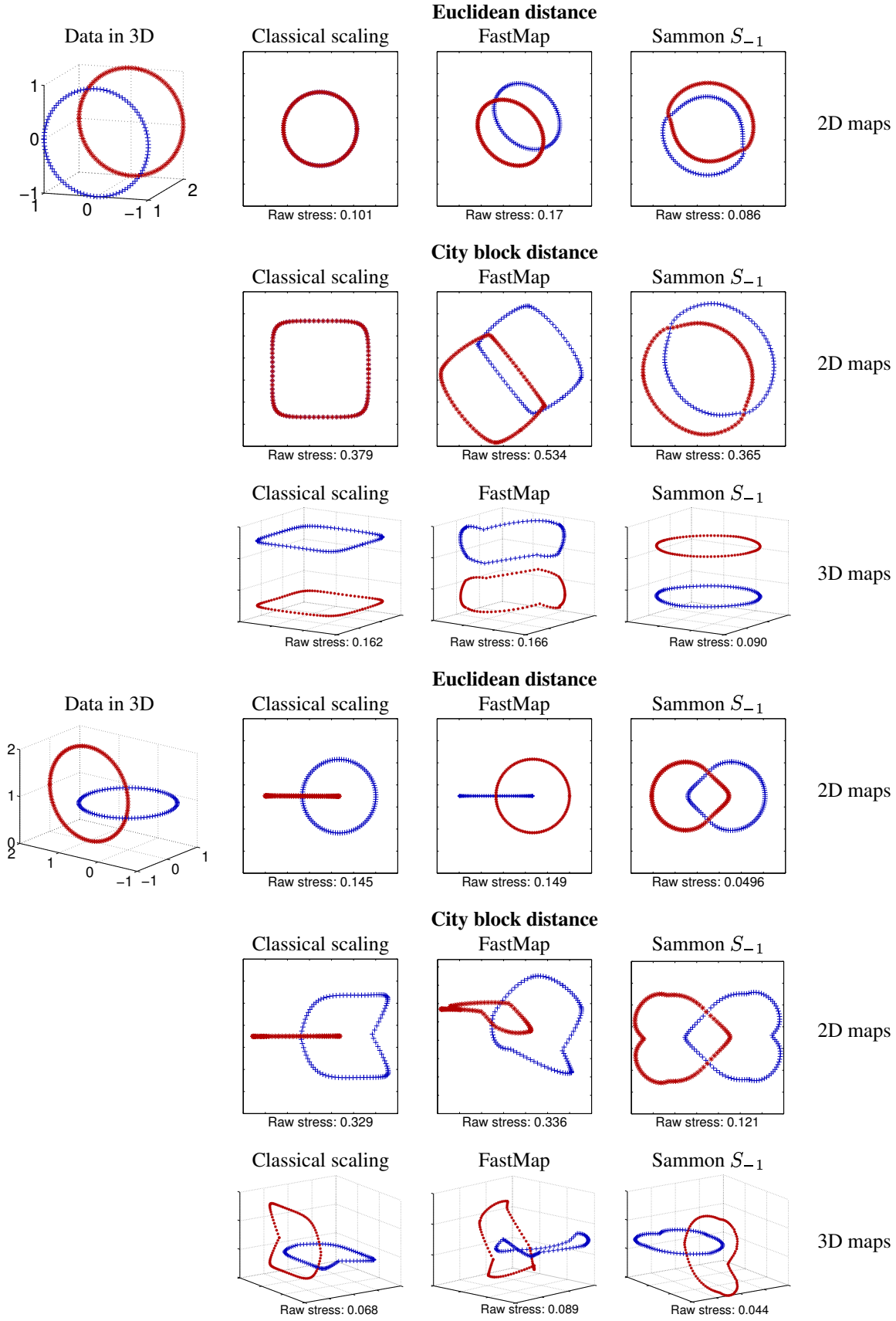


Fig. 6.4: Two circles in 3D (left) and their 2-or 3- dimensional MDS maps based on either Euclidean or city block distance representations. The scales within $2D$ and $3D$ maps are identical.

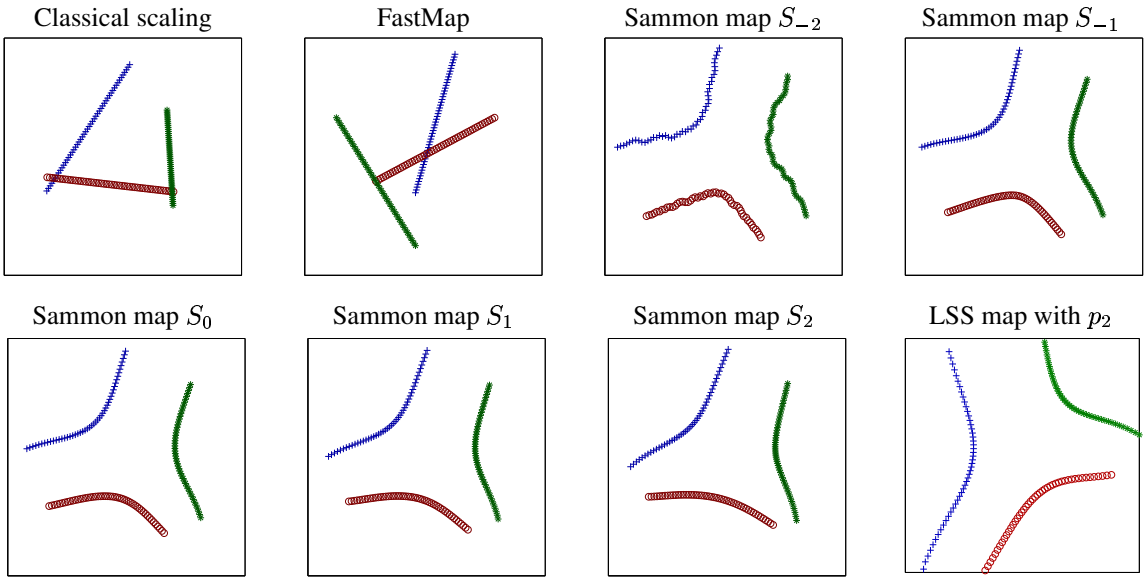


Fig. 6.5: MDS maps of the three straight non-crossing lines in 5D represented by Euclidean distances. The LSS map is provided as a reference. The scale is preserved in all the subplots.

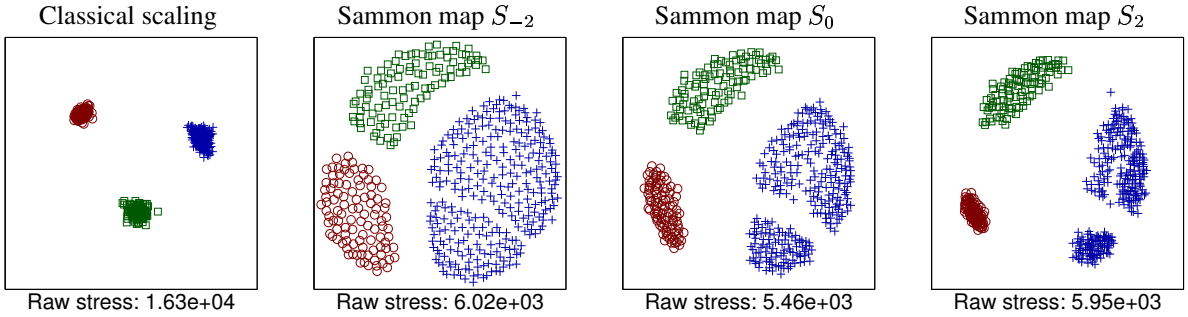


Fig. 6.6: MDS maps of the l_1 distance representation of the *Pump* data. Three operating states are distinguished: normal, marked in circles, imbalance, marked in squares and bearing failure, marked in crosses. The result of the FastMap is not presented, since it looks very similar to the classical scaling output. The scale is preserved in all the plots.

Pump vibration data. The *Pump* data set consists of 500 observations with 256 spectral features of the acceleration spectrum [247]; see also appendix A. It is known [427] that the data have a low intrinsic dimensionality. The MDS projections based on the city block distances are shown in Fig. 6.6. Both classical scaling and FastMap reveal three non-overlapping clusters, while the Sammon mappings with the stresses S_0 and S_2 show, however, much more structure in the data. From the Sammon results, new information can be obtained: the class of bearing failure (marked in '+') is composed of two or even three subclasses, corresponding in fact to the three operating speeds used.

The MDS maps can provide additional insight into the data, especially when the data are highly nonlinear. In practice, this means that many dimensions are necessary to explain a high percentage, like 80% or 90%, of the total variance in the data by the classical scaling approach. In case of the l_1 distance representation of pump vibration, two dimensions explain about 36% of the total variance and 107 dimensions would be needed to reach 80%. Basically, in order to judge the intrinsic dimensionality, a series of the MDS mapping should be performed to a space of a growing dimensionality. Then, the plot of the stress as a function of dimensionality can be obtained, as in Fig. 6.7. From such a figure, one can find

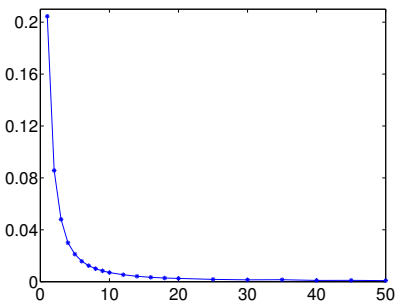


Fig. 6.7: S_0 stress versus the dimensionality for the *Pump* data.

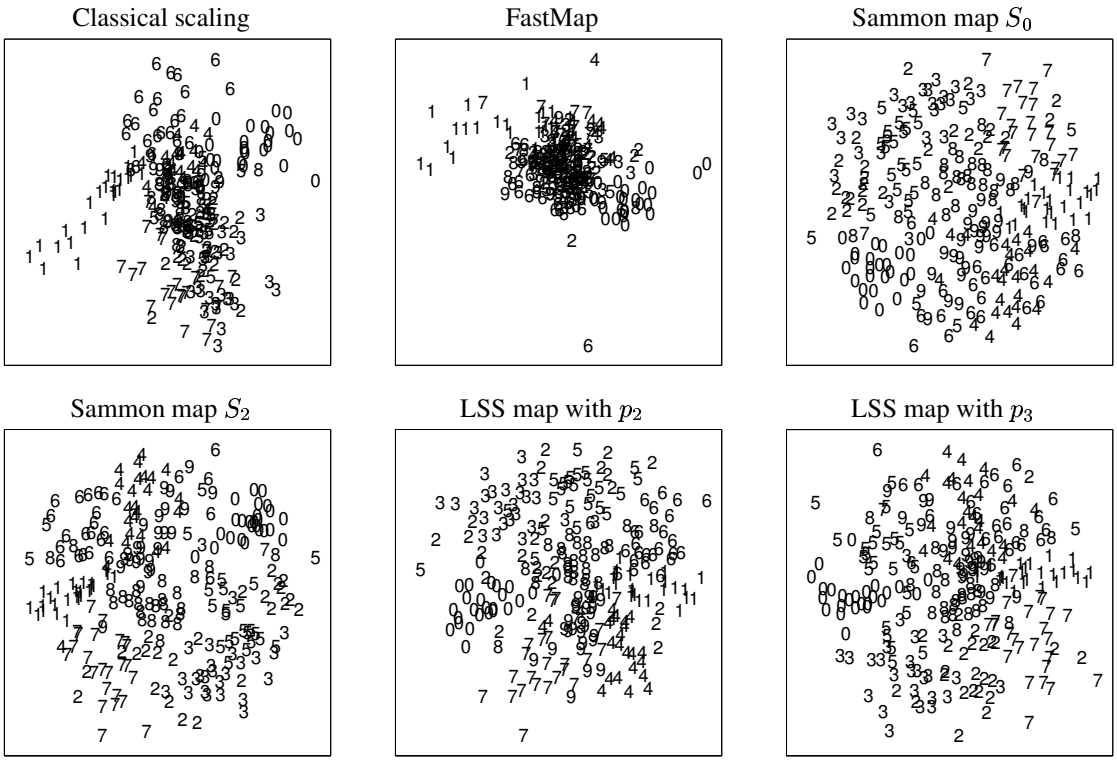


Fig. 6.8: MDS outputs for the *Zongker* dissimilarity data, describing dissimilarities between digit images; see appendix A.2. The disparities in the LSS maps are modeled by p_2 and p_3, i.e. polynomials of the second and third orders, respectively. The scales in the plots are not comparable.

a potential intrinsic dimensionality, which corresponds to a point where the rapid decrease in the stress function stops. For the *Pump* data, it would be around 6 or 7 dimensions. Another indicator of the intrinsic dimensionality can be provided by the number of significant eigenvalues found in classical scaling, however, nonlinear MDS techniques usually need much less.

Zongker digit data. The data set describes the NIST digits [420], originally provided as 128×128 binary images. Here, the similarity measure, based on deformable template matching, as defined in [207], is used; see also appendix A.2. For visualization purposes, a random subset of 25 digits per class has been chosen. Fig. 6.8 presents 2D MDS maps. It can be observed that according to the classical scaling result, the classes of '0', '1' and '6' are the most distinguishable. The first two classes are also mostly separated in the other MDS results. On the other hand, in nonlinear MDS outputs, the class of '2' is the most scattered overall.

Missing values. Any nonlinear MDS can handle missing values. This can be implemented by incorporating extra weights w_{ij} of zeros and ones, as given in formula (3.31), such that zeros account for the missing information. Provided that the data items are labeled, it is even possible to consider a case, where only the dissimilarities between classes are available; see Fig. 6.9. In general, even a large amount of the data can be missing, as shown in Fig. 6.9 and 6.10.

Implementations

Initialization. Different starting configurations are important as they can influence the resulting projections. Each initialization gives potentially a possibility to end up in a different configuration. It follows from our experience that initializing a Sammon projection by classical scaling (CS) often gives good results [297–299]. Another advantage is that the minimization process is also relatively short. Therefore, such an initialization is applied in most cases. It is, however, always useful, to analyze the MDS result based on a pseudo-random initialization. The optimization procedure ini-

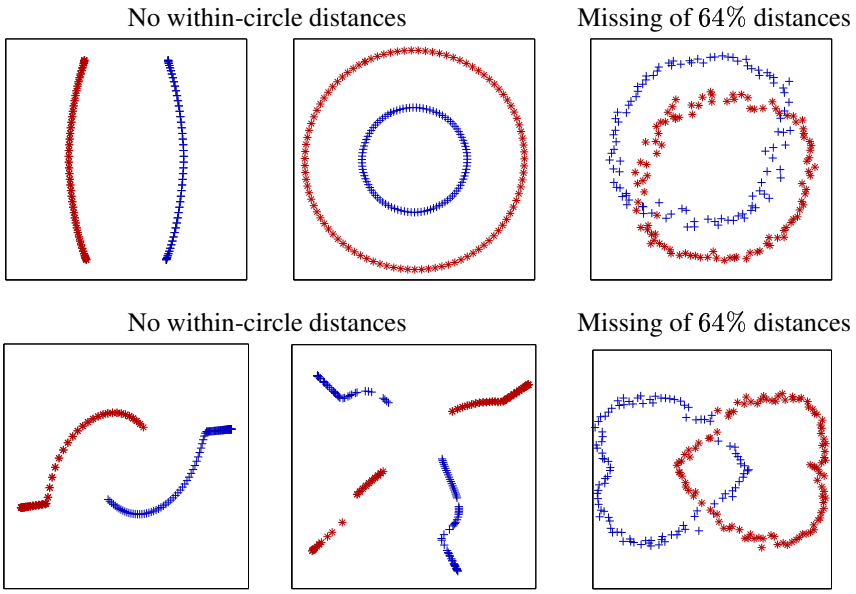


Fig. 6.9: MDS outputs of the Sammon mapping S_{-1} based on the l_1 distance representations between two circles, either parallel (top row) or enclosing each other (bottom row), with missing values. The first two plots, starting from the left, present the results where only distances between the circles where supplied. The differences are due to different initializations. The rightmost plots show the results, when around 64% of distances where randomly removed from the data.

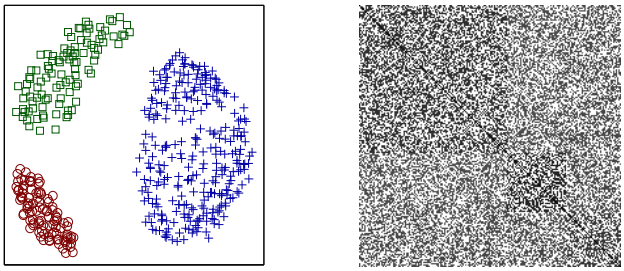


Fig. 6.10: MDS output (left) of the Sammon mapping S_0 on the l_1 distance representation with 50% of missing values for the *Pump* data and the corresponding dissimilarity matrix presented as an image (right), where white pixels denote the missing information.

tialized by the classical scaling may, in some cases, got stuck easily in a local minimum. One may also add some noise to the output of classical scaling. Recently, also Malone at al. [256] argued that a better initialization than the one offered by the CS can be considered by finding a proper term β scaling the output of the CS; see also section 3.4.2.

Algorithms. There exists a number of different MDS implementations ready for use; see [37, 72] for an overview. From our experience with the variants of Sammon mappings [297–299], we have found out that both the Newton-Raphson minimization technique [308] with a line search algorithm and scaled conjugate gradients (SCG) [272] provide good results. The Newton-Raphson method, besides the gradient information, uses also the second order information, approximating the full Hessian by its diagonal matrix. The SCG is a combination of a nonlinear conjugate gradients technique [308] with a trust-region variant. In the beginning, it attains a very large decrease of the stress function, slowing down considerably after the first few iterations. Therefore, it might be beneficial to start the optimization process from the SCG method and switch to the Newton-Raphson technique after a while for a better combination of the efficiency and performance. To stop the iteration process, the following criteria can be used: ${S_t}^i - {S_t}^{i+1} < \varepsilon_{prec}\,(1 + S_t^{i+1})$ or $||X^{i+1} - X^i|| < \sqrt{\varepsilon_{prec}}\,(1 + ||X_{i+1}||)$, where ε_{prec} stands for a chosen precision value and $||\cdot||$ is the

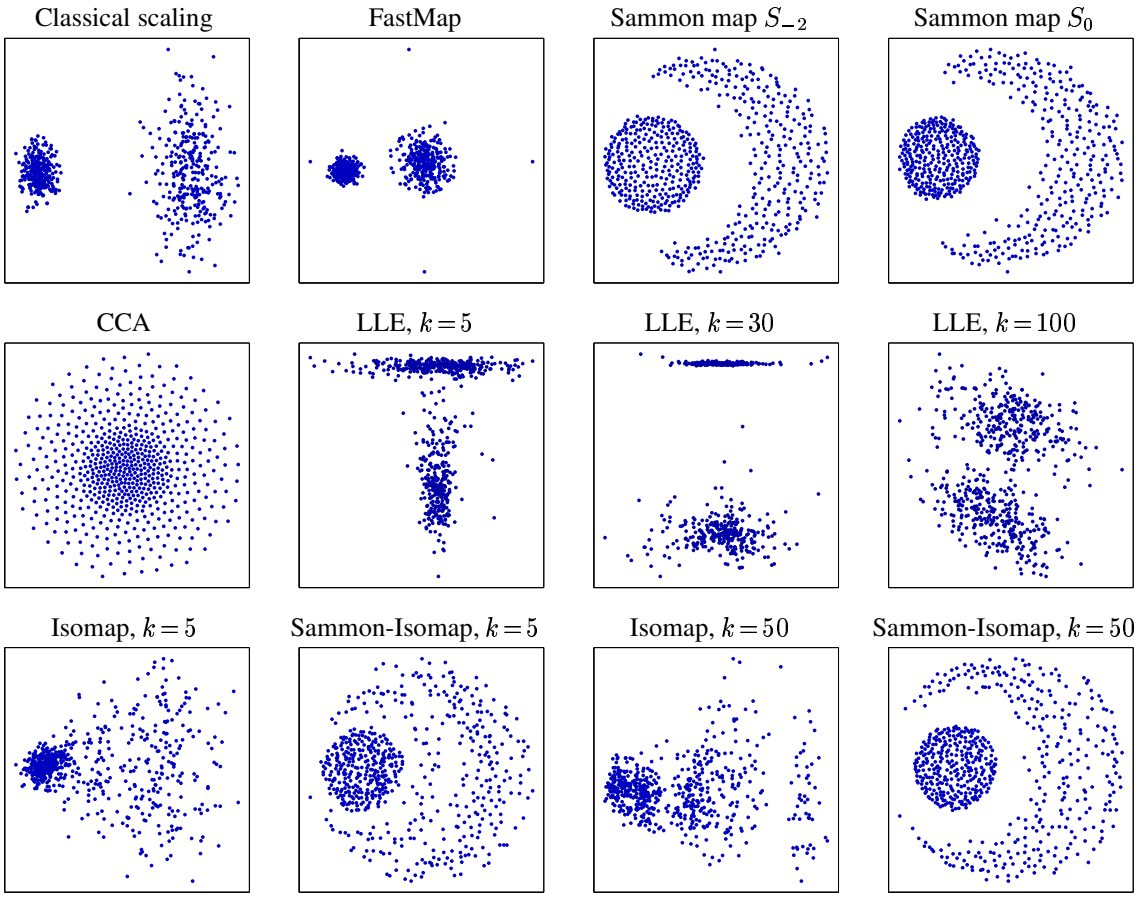


Fig. 6.11: The outputs of various projection methods based on the Euclidean distance representations of the *Hypercube* data in 100D. The scales are not comparable.

Euclidean or max norm. The superscript indicates the iteration number. All our results presented here are based on the first criterion with ε_{prec} equal to 10^{-6} or 10^{-7}.

6.2 Other mappings

In real applications, large high-dimensional data can be modeled as points lying close to a nonlinear low-dimensional manifold or a linear subspace. Examples include image vectors of the same digits, scaled, thickened and tilted or image vectors of the same objects under different camera positions and lighting conditions. Another example is given by document vectors in the complete database related to a specific topic. Usually, such feature representations live in very high-dimensional spaces (described e.g. by the number of image pixels or the number of terms/phrases in the vocabulary of the text database). The intrinsic dimensionality, however, is often limited due to e.g. physical constraints or the degrees of freedom of the measuring tools.

This observation has recently led to a growing interest in developing algorithms for finding nonlinear low-dimensional manifolds (or subspaces) from data represented in high-dimensional spaces. This can serve the purpose of data visualization as well as identification of the underlying variables, such as the degree of tilting, angle of elevation or direction of light, given the high-dimensional data.

Two main directions can be identified: one based on the preservation of the geodesic distances between the data points (or objects in general) with respect to the assumed underlying manifold and the other direction describing the global structure in terms of (overlapping) local structures. The latter research line follows the already established methodology of self-organizing maps (SOMs) [220], generative topographic mappings [29, 30], principal curves [189] or topology-preserving networks

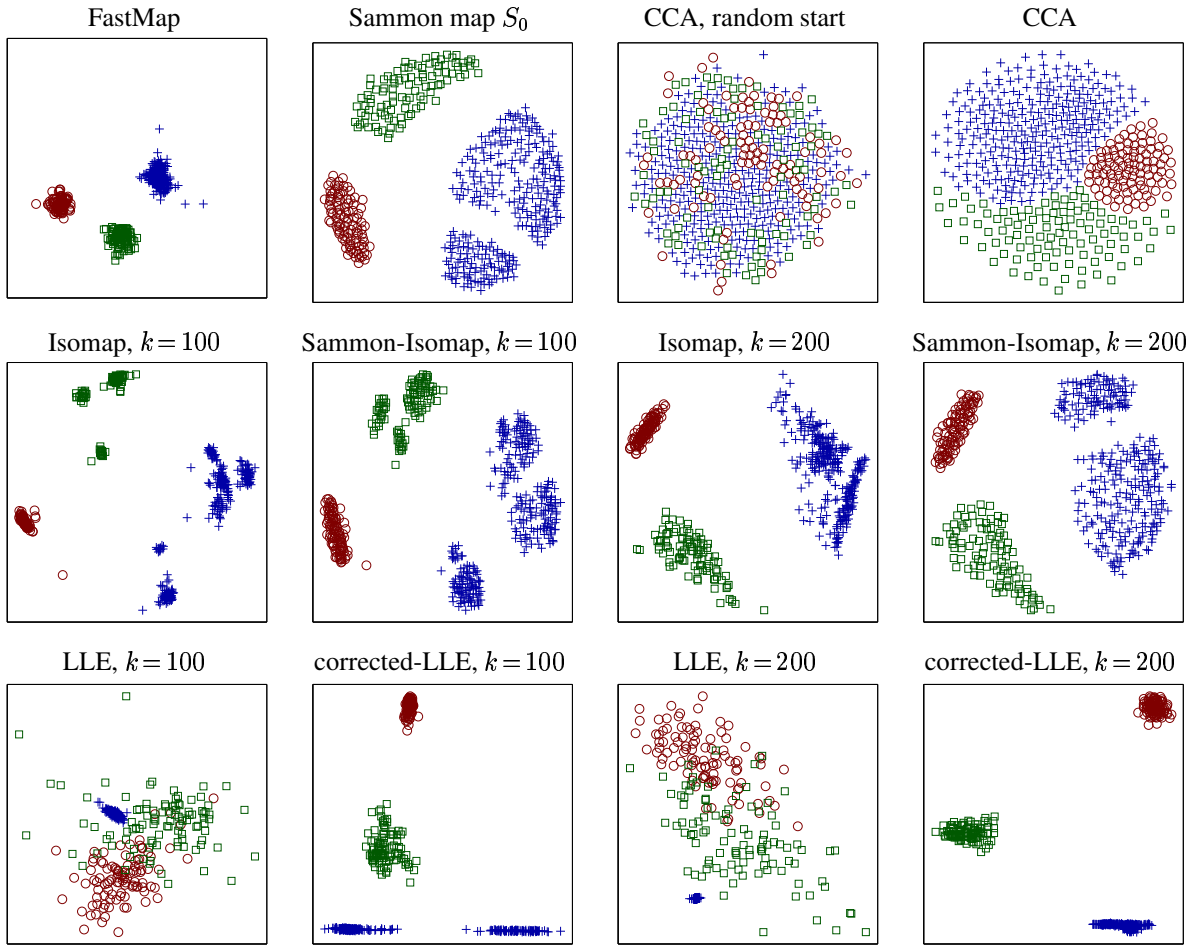


Fig. 6.12: The outputs of various projection methods based on the l_1 distance representation of the *Pump* data. k denotes the numbers of neighbors taken into account for the definition of local neighborhoods. For $k<100$, the LLE projection reduces to three points, while Isomap determines the geodesic distances between 300 vibration spectra only. The scales are not comparable.

[261], however, with emphasis on simple and reliable implementation. Two recent examples of both research lines will be discussed.

Locally linear embedding (LLE). This technique [321, 322, 335, 392] has recently gained a lot of attention. It constructs a manifold by preserving local geometric structures, collectively analyzed, which are invariant to rigid transformations in a neighborhood of each point. In brief, the algorithm can be summarized in three steps:

(1) Compute k neighbors of each data point.
(2) Find the weights that best reconstruct each point from its neighbors by constrained linear fits.
(3) Determine the vectors in a low-dimensional space which are best reconstructed by the derived weights in terms of some constrained least-square problem.

It turns out that the solution to the LLE resolves into the problem of finding eigenvectors of some large, yet sparse, matrix, which encodes information on local neighborhoods. By this sparsity, such an implementation can be made efficient. The difficulty, however, arises since the weights in the step (2) rely on the inverses of the local Gram (inner product) matrices, which should be regularized to avoid singularities.

Since for the LLE the computation of weights is based on local Gram matrices, there exists a straightforward implementation of the LLE method based on Euclidean distances [335]; see also section 3.2.1 on the linear relation between the Gram matrix and square Euclidean distances. Based

on the same principle, non-Euclidean distances can also be used. For small neighborhoods, non-Euclidean distances will approximate the Euclidean ones well. For large neighborhoods, however, the deviation from Euclideaness might be significant. Still, the derived information can be used as an approximation. Here, our proposal is to use the local corrections of the indefinite Gram matrices to make them positive definite by adding proper constants, as discussed in section 3.3.2. We will denote this method as the *corrected-LLE*.

Isomap. This technique [393] has also become popular. It shares some virtues of the LLE, however its philosophy is different, since it is based on notion of geodesic distances (the shortest distance between two points on a manifold). Geodesic distances can be approximated by summing a sequence of distances between neighboring points. These approximations are computed efficiently by finding shortest paths in a graph with edges connecting neighboring data points. Roughly speaking, the Isomap algorithm has three steps:

(1) Determine k nearest neighbors for each data point based on the given distances.
(2) Estimate geodesic distances between all pairs of points by computing their shortest-path distances in the weighted graph with edges weighted by distances between neighboring points.
(3) Apply classical scaling to the geodesic distance matrix.

In this sense, Isomap is a nonlinear extension of the classical MDS, in which embedding is optimized to preserve geodesic distances. Isomap is asymptotically guaranteed to recover the true dimensionality and geometric structure of a class of nonlinear manifolds, whose intrinsic geometry is of a convex region of Euclidean space, but however, the manifold might be highly folded, twisted or curved in a high-dimensional space; see [393] for proofs.

Both Isomap and LLE refer to the construction of low-dimensional manifolds in a nonlinear way. Their applicability to general data represented by dissimilarities will be, however, limited due to an underlying assumption of a densely sampled manifold. Moreover, the choice of a proper neighborhood size (i.e. the number of neighbors or equivalently, the ε-neighborhood) might be problematic; we have observed that especially the LLE is sensitive to this aspect. The possible failure of the LLE is then to map far away points to the nearby outputs in the projected space. On the other hand, Isomap is dominated by the preservation of far away (geodesic) distances (since the classical MDS minimizes the raw stress) at the expense of distortions in local geometry. Consequently, their usefulness is justified for well sampled data. From that point of view, the traditional MDS techniques might be preferable to get insight into the structure of the, possibly undersampled, data. Still, we think that the preservation of geodesic distances can reveal additional aspects of the data. Then, we would propose to perform a nonlinear MDS on the approximated geodesic distances instead of classical scaling as done in Isomap. The reason is to put more emphasis on local geometry. We will denote it as *Sammon-Isomap*.

Another technique trying to discover an underlying spatial structure of the data is the kernel-PCA [353, 354], which loosely speaking, performs the PCA in the space defined by the kernel map. It starts from a positive definite kernel matrix K interpreted as a generalized inner product matrix, which serves further for finding the principal directions of the space it describes. If one starts from a square Euclidean dissimilarity matrix, the kernel-PCA on the corresponding Gram matrix is equivalent to a process of an approximate embedding of the distances to an underlying Euclidean space, as discussed in sections 3.3.1 and 3.3.6. This is exactly the classical scaling projection to a few dimensions. Although any other positive definite kernel can be used, this can always be interpreted from a classical scaling point of view for an appropriate distance matrix, formula (4.28). For that reason we will not investigate its performance here.

Another research line focuses on unfolding a nonlinear structure present in the data. It has been started by Curvilinear Component Analysis (CCA) [82, 180, 194], which draws an inspiration from

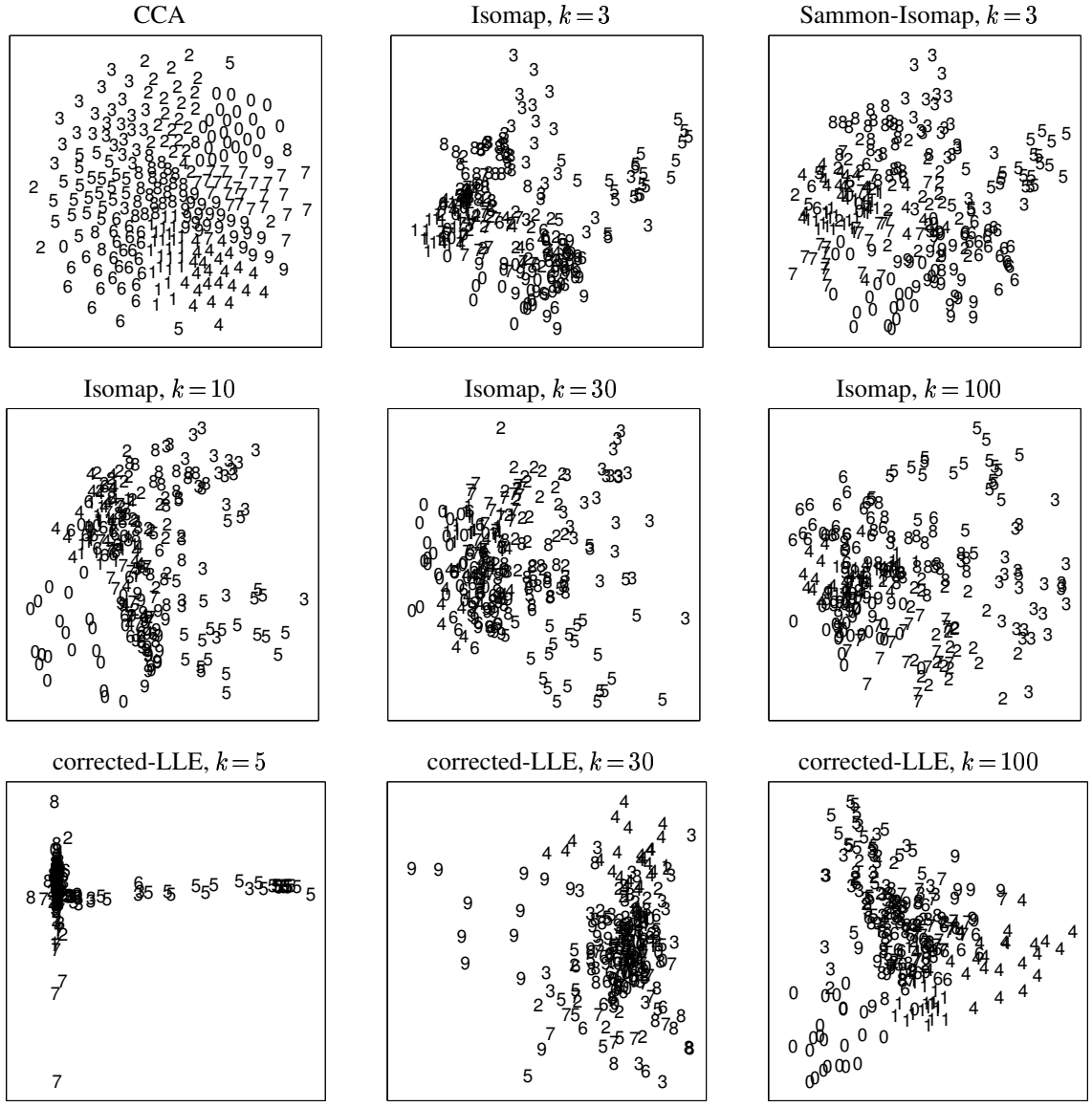


Fig. 6.13: The outputs of various projection methods for the *Zongker* data. The dissimilarities between the digit images are computed in a template matching process. k denotes the numbers of neighbors taken into account for the definition of local neighborhoods. The scales are not comparable.

the MDS techniques and Kohonen SOM [220]. It is based on the minimization of the least-square loss function (similarly to MDS), but making use of an additional weight function F which depends on the current estimates of the approximated distances in a Euclidean space. F is a decreasing and bounded function of its argument, such as exponential, sigmoid or a step function, so it is used to favor local topology preservation (similarly to SOM). Consequently, the CCA tries to reproduce the short distances first and then, the large ones. An additional value is the efficient implementation; see [82] for details. Basically, the loss function is given as $E(X)=\frac{1}{2}\sum_{i<j}(\delta_{ij}-d_{ij}(X))^2F(d_{ij}(X),\lambda_X)$, where δ_{ij} are the given dissimilarities, d_{ij} are Euclidean distances in the projected space and λ_X is the neighborhood parameter. By the focus on distances in local neighborhoods, unfolding of a manifold is reveled more significantly than in case of the MDS techniques. This means that for large dissimilarities, d_{ij} tends to be larger than δ_{ij}, on average. It is emphasized in [82] that due to the special loss function, the CCA method is able to better preserve local topology when mapping data from dissimilarities to a Euclidean space. On the other hand, although the CCA might be very beneficial for well sampled manifolds, it may locally get into too much details of reproducing the dissimilarity structure, especially for data yielding some clusters. The information can be lost.

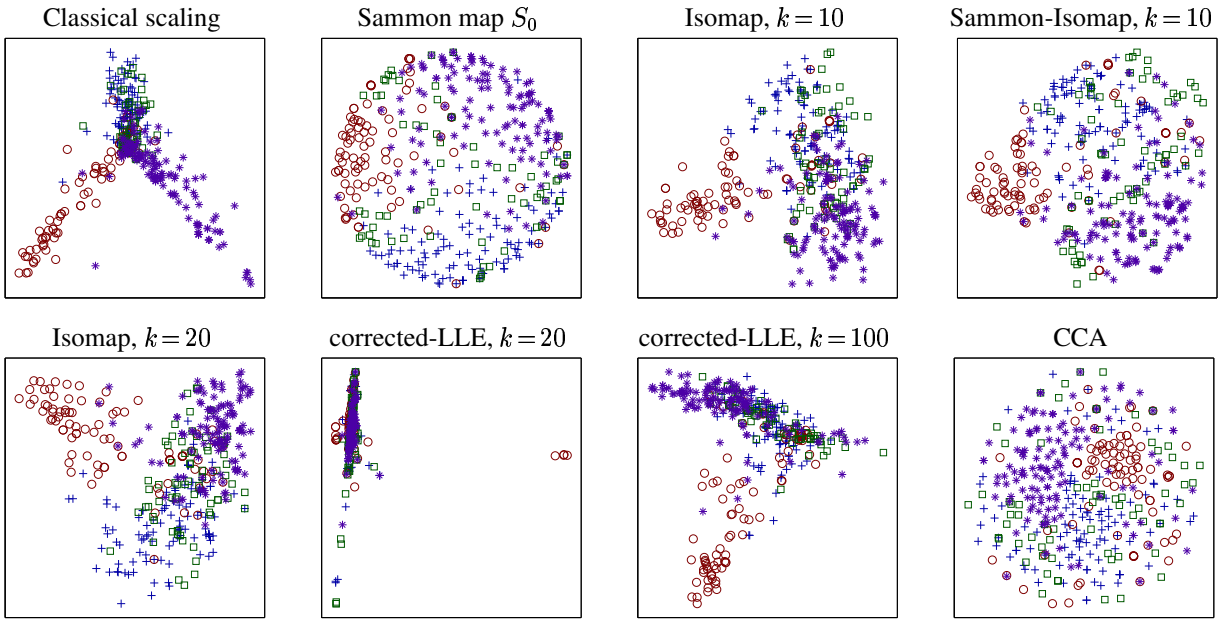


Fig. 6.14: The outputs of various projection methods based for the *News-cor* data defined by the correlation-based dissimilarities between the text newsgroups: 'comp.*', marked in crosses, 'rec.*', marked in circles, 'sci.*', marked in squares and 'talk.*', marked in stars. k denotes the numbers of neighbors taken into account for the definition of local neighborhoods. The CCA result is presented after 200 iterations, however, even 2000 iterations did not change the results significantly. The scales are not comparable.

An extension of the CCA method is offered by Curvilinear Distances Analysis [241, 242]. The novelty relies on the use of curvilinear distances, expressing the distance measured along the structure, instead of the original distances δ_{ij}. Such curvilinear distances are computed as the shortest path between two chosen prototypes, after their quantization and linking.

Examples. Here we will present some embedding examples of artificial and real dissimilarity data. The LLE and Isomap routines come from the specially dedicated web pages, see [205, 250]. Since, in general, Isomap or the LLE are suitable for locally linear, but globally nonlinear, embeddings, the dissimilarity data should be more complex than representing the distances between two circles. By Sammon-Isomap, we mean that the embedding procedure follows the Isomap routine until the estimation of geodesic distances, but then it uses the Sammon mapping S_0 (instead of classical scaling) to find the $2D$ representation. The CCA result is presented after 50 or 100 iterations, being initialized by the classical scaling result. We have also noticed that in a number of cases, when the random initialization was used for the CCA, 2000 iterations were not sufficient to discover the structure in the data; see also Fig. 6.12.

Let us consider the Euclidean distance representation of the *Hypercube* data as described in appendix A.1 for details. The data points are generated inside two enclosing hypercubes in a 100-dimensional space. The results of various mappings are presented in Fig. 6.11. Concerning the distance data, Fig. A.2, according to our judgment, the Sammon mapping and Isomap reveal the data structure most appropriately, namely one compact cluster (corresponding to a smaller hypercube) with points around this cluster, possibly building a cloud. Of course, there is an inherent side effect of the Sammon stresses to give more sphere-like shapes than squares, which has been already mentioned in section 3.4.2.

Pump vibration spectra represented by the l_1 distances (see appendix A.2 for the data description and section 6.1 for the MDS results) is a difficult case for both the LLE and Isomap, since they describe well separated classes. The sampled 'manifold' is not continuous, hence many nearest neighbors have to be taken into account in order to discover such a structure. For less than $k = 100$ nearest

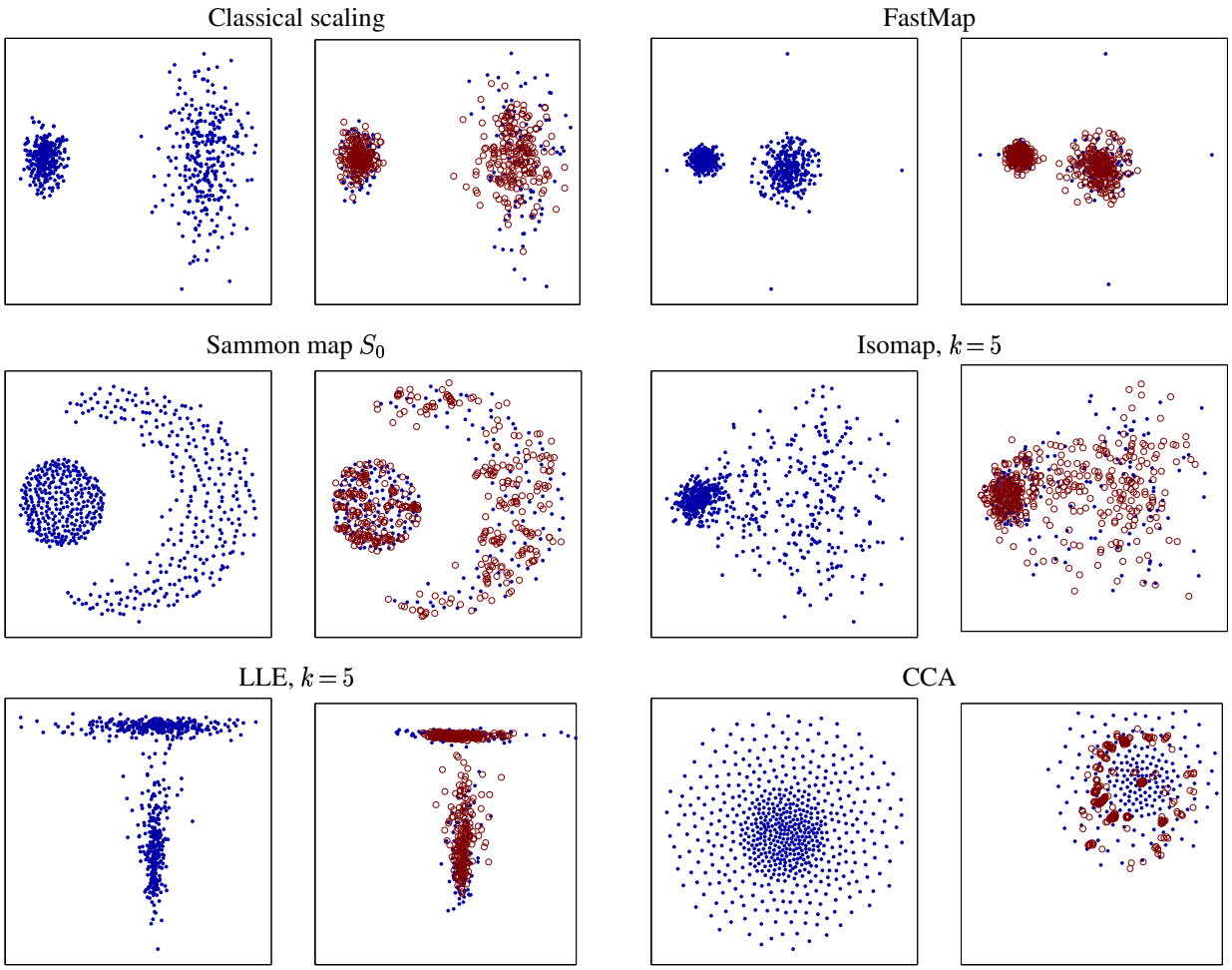


Fig. 6.15: An illustration on the generalization abilities of each of the projection methods: classical scaling, Sammon mapping, LLE, Isomap and CCA. Each two subsequent plots correspond to one method and their scales are identical. From each pair, the left plot presents the projection of the Euclidean distance representation of the *Hypercube* data based on *all* points, the right plot shows the result when first the map was established by 200 randomly selected points (marked by dots) and then the remaining 400 points were added to the existing map (marked by circles).

neighbors, the LLE method collapses to the result of three points in a $2D$ space. For such a case, also Isomap projects points on the top of each other. Many nearest neighbors have to be included and, as a side effect, both methods become more costly than the nonlinear MDS mappings. The results of various mappings are shown in Fig. 6.12. Note that the CCA concentrates on the locality so much that it looses the ability to show the separateness of the classes. It has also difficulties to present a good solution when the initialization is random; see second plot, top row in Fig. 6.12. From all the plots, the Sammon map S_0 is the only one which detects three subclusters in the bearing failure mode of the pump; see Fig. 6.6 for the MDS results.

Spatial representations of the *Zongker* dissimilarity data are presented in Fig. 6.8 (MDS maps) and Fig. 6.13 (other maps). Note that these data are an example of highly non-metric dissimilarities. While the MDS methods find, in general, the classes of '0' and '1' as the most confined, Isomap considers the classes of '3', '5' (and '0' for $k = 10$) as the most distinguishable. The remaining classes are heavily overlapping as judged from the Isomap result. The LLE could not detect any sensible structure in the data, also for larger neighborhoods (not presented here). Depending on the neighborhood size, the corrected-LLE distinguishes the classes of '5','9','4' and '0'. Still, the results vary tremendously with the increasing locality, hence it is hard to draw clear conclusions. According

to the CCA map, '8' is the central class, similar to all other classes, '1' is the most compact class and '2' is the most confusing (since single examples of '2' appear in various places). Since the CCA method 'unfolds' the data, it is hard to judge which classes are potentially overlapping, hence difficult for the classification task.

The last example refers to the *News-cor* data, the newsgroups data, for which the non-metric correlation-based dissimilarity representation was computed; see appendix A.2 for the data description. The results of the mappings are presented for randomly chosen 100 objects per class. They can be observed in Fig. 6.14. In general, the newsgroup 'rec.*' is the most well-defined class, followed by the 'talk.*' group, as revealed by the MDS maps and Isomap. The corrected-LLE seems to detect the cluster of 'rec.*', however for a large neighborhood.

Generalization abilities. Classical scaling and Isomap can naturally be extended such that the new data is added to an existing map. This is due to the fact that such a generalization relies on an orthogonal projection, which can be easily applied; see section 3.3.5 for details. The possibility of adding new points to the Sammon map, by an iterative minimization of a modified stress function, has already been discussed in section 3.4.3. The extension of the LLE is straightforward by finding for each object its k nearest neighbors and determining the weights in a lower-dimensional space such that the projected point can be in the best way represented as a linear combination of its neighbors. The generalization of the CCA is also apparent and described in [82]. In principle, this suggests that any of the mappings described so far, can be used for the classification purposes. An example of their generalization abilities is presented in Fig. 6.15. In our example, however, the CCA does not seem to generalize well.

6.3 Tree models

A tree structure of the dissimilarity data enhance a natural interpretation of relations between the objects. It is a useful tool utilizing the understanding of the data structure, as by the inference of the organization of objects, especially for a smaller number of them. Moreover, trees support the hierarchical clustering scheme based on proximities. Such discrete models can be considered as complementary to the continuous spatial representations obtained e.g. by the MDS techniques. The key discrete model is the additive tree model, which represents objects by nodes of a tree and defines dissimilarities as path lengths between two nodes.

An additive tree is a connected, undirected graph where each pair of nodes is joined by a unique path. An $n \times n$ dissimilarity matrix D defines a unique additive tree if D is additive, hence l_1-embeddable. This means that the distance between two points is a path metric realized by the sum of positive weights along the path connecting the points; see section 3.1.2. From an algorithmic point of view, the additivity of D stands for D being a metric and fulfilling the four-point inequality as presented in Def. 3.12. A special case of an additive tree is an ultrametric tree, which is intimately related to the hierarchical clustering of the data. It is an additive rooted tree in which the distance from the root to every leaf is identical, as in dendograms. Formally, an $n \times n$ dissimilarity matrix D defines a unique ultrametric tree if the ultrametric inequality as in Def. 3.14 holds.

Note that in additive trees the root is not determined, hence different interpretations may be suggested by choosing different roots. Basically, the root helps in distinguishing of some clusters in the data, so it could be chosen to enhance the interpretability of the data. This, however, requires some prior knowledge. Another possibility is to place the root at a node which minimizes the variance of the distances from the root to the leaf nodes, so it splits the data into homogeneous clusters.

In practice, there might be no tree metric coinciding exactly with the given dissimilarity matrix D, hence no representation by an additive or ultrametric tree. This means that a tree metric $\tilde{D}$ can

be sought which provides the best approximation of D under some criterion, e.g. given by a loss function such as the l_1, l_2 or l_∞ norms. This is a formulation of the numerical taxonomy problem; see e.g. [8, 216]. Such tasks of fitting an additive or ultrametric tree are known to be NP-hard under the l_1 and l_2 loss [64, 216, 370]. In case of the l_∞ norm, the same holds for an additive tree [1], however the optimal ultrametric tree can be computed in a polynomial time [123]. There exists a number of other methods trying to construct such trees so that the path distances approximate the given distances as well as possible; see e.g. [1, 66, 123, 139, 140, 373, 374] for specific algorithms. Below we briefly mention some of such tree fitting techniques.

Approximation under the l_2 norm. The dissimilarity data D can be approximated by an additive or ultrametric tree in terms of the least square error. If $D = (d_{ij})$ are the original dissimilarities, the dissimilarities $\tilde{D} = (\tilde{d}_{ij})$ defining either an additive or ultrametric tree are sought such that in terminology of the MDS, the raw stress $\sum_{i<j}(\tilde{d}_{ij} - d_{ij})^2$ is minimized. This can be formulated as:

	Additive tree	Ultrametric tree	
Minimize	$L(\tilde{D}) = \sum_{i<j}(\tilde{d}_{ij} - d_{ij})^2$	$L(\tilde{D}) = \sum_{i<j}(\tilde{d}_{ij} - d_{ij})^2$	
s.t.	$\tilde{d}_{ij} + \tilde{d}_{kl} \leq \max\{\tilde{d}_{ik} + \tilde{d}_{jl}, \tilde{d}_{il} + \tilde{d}_{jk}\}$	$\tilde{d}_{ij} \leq \max\{\tilde{d}_{ik}, \tilde{d}_{jk}\}$	$\forall i, j, k, l$

De Soete [371, 372] has proposed a practical algorithm to solve these constrained optimization problems by transforming them into a series of unconstrained problems.

Approximation under the l_∞ norm. It is known [169] that given a distance matrix D, there exists a unique ultrametric distance matrix D_U such that $D_U(i,j) \leq D(i,j)$ for all pairs (i,j) and D_U is maximal, i.e. all other ultrametric distance matrices are dominated by D_U. One way to find D_U is to construct a minimum spanning tree[1] T_D on the complete graph whose weights become the distances of D. Then, D_U is built from maximum weights of the edges in T. The same tree is obtained in a greedy agglomerative approach of the *single-linkage* (SL), algorithm, which is of quadratic complexity. It first starts with all objects in their own clusters. Then, repetitively, it finds the two clusters with the closest distance and merges them into one cluster until there is one cluster left. After every merging, the distance between the new clusters is recomputed and all other distances are reduced. Due to its simplicity, the SL algorithm has become popular and it is widely used in cluster analysis.

A possibility to fit an additive tree to a given dissimilarity matrix D is by using the neighbor joining heuristic [328]. Conceptually, the method is related to the SL algorithm, but without resorting to the assumption of an ultrametric tree. The idea here is to join the clusters that are not only close to one another, but are also far from the rest. The method begins with all objects in their own clusters (leaves). In each step, the algorithm attempts to find the direct parent of the two nodes in the tree. For the i-th node, its average distance to the other nodes is estimated as $m_i = \frac{1}{n-1}\sum_{j \neq i} D(i,j)$. In order to minimize the sum of all branch lengths, the nodes i and j that are clustered next are those for which $D(i,j) - m_i - m_j$ is smallest. The distances between the nodes are recomputed appropriately. The algorithm stops when all objects belong to one cluster. Its time complexity is $\mathcal{O}(n^2)$.

Another approach to fitting an additive tree relies on the property that an additive metric D_A can be characterized by an associated ultrametric via a centroid metric. A *centroid metric* D_C is a metric which is realized by a weighted tree with a star topology (i.e. a tree with all leaves but one) and edge weights w_i. Then, $D_C(i,j) = w_i + w_j$. More formally, for a chosen a, let $m_a := \max_i D_A(a,i)$. Then, the centroid metric is defined by the weights $w_i = m_a - D_A(a,i)$ such that $D_C(i,j) = w_i + w_j =$

[1] A *minimum spanning tree* (MST) is a tree T_{mst} that spans all the nodes and minimizes the total weight of the tree, i.e. $\sum_{e \in T} w_e$. An MST constructing algorithm starts from an arbitrary root node and grows until the tree spans all the nodes. The algorithm is greedy since the tree is augmented, step by step, with an edge that contributes the minimum amount possible to the total weight cost. MSTs can be used to solve tree optimization problems.

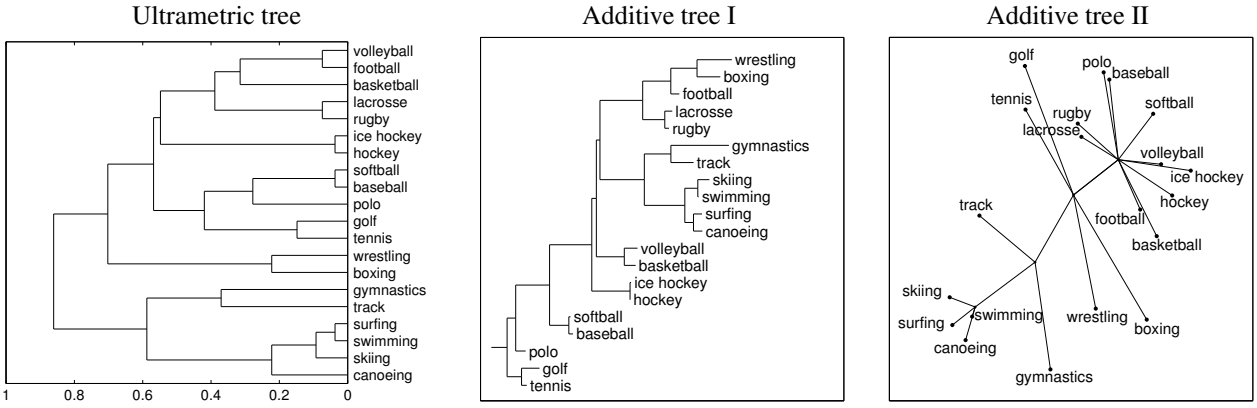


Fig. 6.16: Tree models for human dissimilarity judgments on various sports.

$2m_a - D_A(a,i) - D_A(a,j)$. D_A is an additive metric iff $D_A + D_C$ is ultrametric [1, 64, 88]. Since, the nearest ultrametric can be found in a quadratic time by the SL algorithm, this suggests a general strategy for fitting an additive metric D_A to D in a quadratic time. Loosely speaking, given D, a centroid metric D_C is chosen and added to D. Then, an ultrametric D_U approximating $D + D_C$ is found. The additive metric D_A is determined as $D_U - D_C$, which should serve for the reconstruction of the tree; see [1, 64] for specific algorithms.

Generalization abilities. It is not clear to us how new objects can be added to the existing trees. To our knowledge, it has not been discussed in the literature, although it is possible to think of constructing additive and ultrametric trees for rectangular dissimilarity matrices $D(T, R)$, where the sets R and T are distinct. Conceptually, the most reasonable approach would be to construct again a tree based on all the dissimilarities, including these of newly coming objects. Yet, this is not a generalization. Surely, one can think of some approaches of adding objects to the existing trees, e.g. by appending them to the objects for which the distances are the smallest, but then the complete additive structure of the tree may be destroyed. So, this remains an open issue.

Two examples. Let us consider the auditory confusion (dissimilarity) measurements for letters and numerals and the human judgments on sports. The fitted ultrametric and additive trees [252, 381] are presented in Fig. 6.17 and 6.16. The same figure contains also a representation of a minimum spanning tree pictured between the points of the MDS map.

In an additive tree the root is not determined, and choosing different roots may suggest different interpretations. Therefore, two different additive trees are shown in the figure: the first one (I) is found such that the root is placed at a node which minimizes the variance of the distances from the root to the leaves and the second tree (II) is unrooted and determined such that it has three or four apparent clusters (or in fact internal nodes). All the presented trees agree in some basic interpretations e.g. on the existence of a clear cluster composed of 'I', '5', 'R', '1', 'Y' and a bit more remote '9' and also identification of generally remote objects as '4' and 'W' in case of Fig. 6.17 or on a basic division of sports into team sports versus individual sports as observed in Fig. 6.16.

6.4 Summary

Spatial models of the dissimilarity data can be realized either by linear and nonlinear projections to an output space or by tree representations of the relations between the objects.

In the first group of methods, multidimensional scaling (MDS) techniques play a special role, since they aim to preserve all pairwise, symmetric dissimilarities, resulting in a faithful, low-dimensional representation, usually in a Euclidean space, of the geometrical relations between the points. Other methods concentrate on the preservation of dissimilarities in local neighborhoods, like locally linear

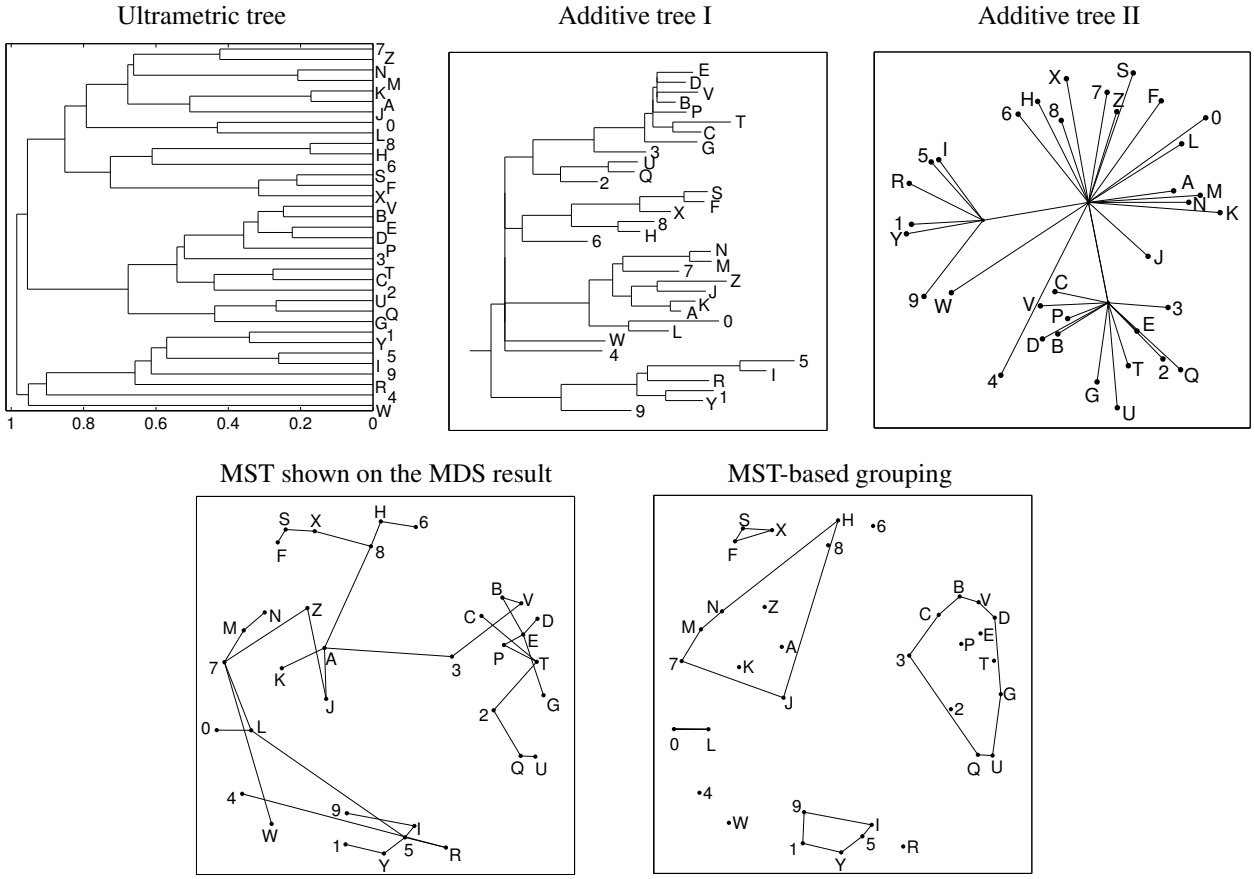


Fig. 6.17: The plots in the top row present tree models of the auditory confusion measurements for letters and numerals. The plots in the bottom row show the minimal spanning tree models drawn on the 2D MDS maps applied to the same data. These plots are made by using some of the routines available at [252, 381].

embedding and curvilinear component (distance) analysis or the preservation of locally estimated geodesic distances, like Isomap. Nonlinear methods can reveal more structure and cluster tendencies than the linear ones. They are, however, much more time consuming. To understand the data better, both of them should be used since they integrate with each other. Classical scaling (a linear projection), accompanied by the Sammon map S_0 and Isomap can provide a good insight into the data. Due to an inherent property of nonlinear MDS techniques to project the data onto spherical shapes, some judgments might be biased. Therefore, both classical scaling and Isomap are useful. Isomap is dominated by the preservation of far away geodesic distances at the expense of distortions in local geometry, while Sammon mapping tries to penalize large distances to maintain the local geometry, hence they complement each other.

Our conclusion, therefore, is that for general dissimilarity representations of possibly undersampled problems, the most revealing projections are the ones based on the MDS principles (including kernel-PCA) and Isomap. Other techniques such as locally linear embedding and curvilinear component (distance) analysis seem to need dense samplings and a clearly identifiable low intrinsic dimensionality, hence their usage is limited.

Tree models focus on the organizational aspects of the dissimilarity data. They enhance understanding of the data in terms of hierarchical or nested structures and, moreover, they are easy to interpret. However, to make the interpretation a feasible process, the objects should be distinct from each other and not too many. Trees naturally support evolutionary processes in which all the objects have an initial structure in common and additional distinctive features are developed later on. Examples is the evolution of the species or languages in time, so, these are clear cases of their applicability.

7. Further data exploration

If you torture data sufficiently, it will confess to almost anything.

FRED MENGER

Understanding data is crucial in the process of designing and validation of learning algorithms. Visualization is often the first step. In the previous chapter, continuous and discrete spatial representations of the dissimilarity data were described, attained by vector configurations in some low-dimensional spaces or by weighted fully connected graphs which facilitate the visualization. Subsequent steps require a more profound comprehension of interrelations among the data instances. Therefore, this chapter focuses further on methods that help in the exploration of the dissimilarity data so that an assessment of the organization and the (underlying) structures can be made.

Three main issues are discussed here concerning both the structure and the complexity in the dissimilarity data representation: clustering techniques, intrinsic dimensionality and the sampling issue. Initially, all given objects are the candidates for the representation set, hence the analysis starts from an $n \times n$ dissimilarity matrix $D(R, R)$. The first question investigates in section 7.1 cluster tendencies in the data. Since the clustering problem has gained a great deal of attention over the years, we are not able to study numerous existing methods; this would be a research issue in itself. Hence, we limit ourselves to the presentation of some essential algorithms, related to the dissimilarities. The second question moves on to the intrinsic dimensionality of the data, to be indicative for the complexity of the class or classes the dissimilarity describe. This is discussed in section 7.2. The third question refers to the sampling issue, i.e. whether the dissimilarity data are represented by a sufficient number of objects. The ideas presented in section 7.3 rely on our earlier work [102].

7.1 Clustering

Clustering has been addressed in many contexts and in many disciplines, reflecting its significance in exploratory data analysis. The purpose of clustering [188, 209, 211, 212, 370] is to improve understanding and to enhance interpretation of the data by organizing them in some meaningful groups such that examples within one group are more closely related than those from different groups. Therefore, such techniques are often used to analyze the structure in the data. Some of the most important applications are image segmentation, data mining and information retrieval or categorization.

The clustering task is subjective, since the data can be partitioned differently depending on what is taken into account. Basically, it reflects the user's needs. For instance, one can be interested in finding 'natural clusters' in the data, representatives of homogeneous clusters, some useful (i.e. easily interpretable) data groupings or even outliers. Consequently, there is no universally applicable technique that would be able to uncover the variety of structures present in the data. Depending on the final aim, a suitable method should be applied.

7.1.1 Standard approaches

Two basic strategies have been developed for clustering: hierarchical and partitioning methods, both encompassing a variety of algorithms. Most of them rely on the notion of a (dis)similarity and a criterion specifying how the clusters are formed. However, the dissimilarity is not just the relative

dissimilarity between pairs of objects, but also the *conceptual* dissimilarity instead, comparing objects (or concepts) and concepts. So, the objects are grouped according to their fit to the specified concepts; see also section 4.2. The concept can be given, for instance, as a density model of a cluster or as an average dissimilarity within the cluster.

Hierarchical clustering. Hierarchical clustering proceeds successively either by merging smaller groups into larger ones or by splitting larger groups into smaller ones. Hence, the methods are either agglomerative or divisive. The final result is a tree of nested clusters, a *dendogram*, such that the complete set is represented by the root, while the leaves are the individual examples. The internal nodes are defined as the union of their children. Hence, each level of the tree represents a partition of the set into several (nested) clusters. By cutting a dendogram at a specified level, a clustering into disjoint groups is obtained. The way the current clusters are merged (or split) depends on the criterion which defines the dissimilarity between the clusters (which is the conceptual dissimilarity). Divisive methods often rely on constructing neighborhood graphs such as the minimum spanning tree and using some principle to remove edges and create the clusters. Agglomerative methods start from the partition where each example forms a cluster and proceed by repetitively merging two clusters with the smallest conceptual dissimilarity until one cluster is left (or a specific number of clusters is reached). Due to the sequential nature of such algorithms, i.e. objects once assigned to a cluster cannot change its label later on, they will not necessarily produce the optimal clustering, even with the prior knowledge of a desired number of clusters.

Hierarchical methods are often applied in Euclidean feature spaces by using the square Euclidean distance as a basic measure. The reason behind this is the interpretability of the results, since the Euclidean distance captures the (imposed) geometry between the clusters in a Euclidean space. Yet, the techniques can be applied to any dissimilarity measure.

Partitioning clustering. Partitioning methods usually operate in (Euclidean) feature spaces. They split the objects into (a priori specified) k groups according to some criterion. They are often model-based techniques, where the clusters are described by some parametric or non-parametric distributions, by the use of representatives or by assuming a specific type of geometrical structures like planes, spheres etc. Hence, the conceptual dissimilarity is the goodness of fit of an object to an assumed cluster model. The primary difference to the hierarchical methods is the need to specify k. Given a hypothesized number of clusters, a general representative-based partitioning procedure chooses the cluster representatives with some strategy. The remaining objects are then assigned to the clusters according to the conceptual dissimilarity, which may be calculated based either on the initial cluster members or on their merged versions such as the average. New representatives are estimated for each cluster and the whole procedure is repeated until a stable solution is reached. Methods differ primarily in the choice of initial representatives, the assignment of objects to clusters and the estimation of representatives.

A typical method is the k-means algorithm [253], where new representatives are estimated by the cluster means and the conceptual dissimilarity is the distance to them. The EM-clustering, based on the expectation-maximization (EM) algorithm (a general maximum likelihood optimization procedure for problems with hidden variables or missing data[1] [83]) is an extension of this basic approach;

[1] To maximize the likelihood, the EM algorithm iterates between the E-step and the M-step until convergance. In the E-step, a posterior probability distribution on the hidden or unobserved variables is estimated, which serves for a further estimation of the model parameters in the M-step, where the likelihood is maximized. EM is usually employed for finding the parameters of a mixture-of-Gaussian distribution. Assume K Gaussian models, where the i-th model is given as $M_i := \{\mu_i, \Sigma_i, \pi_i\}$ and the total model structure is $M := \{M_1, \ldots, M_K\}$. Then, the mixture-of-Gaussian is described as $p(\mathbf{x}, M) = \sum_{i=1}^{K} \pi_i \, p(\mathbf{x}|\mu_i, \Sigma_i)$, where $\pi_i \geq 0$ and $\sum_i \pi_i = 1$. Given a population $\mathcal{X} := \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the optimized log-likelihood becomes then $LL(\mathcal{X}) = \log p(\mathcal{X}|M) = \sum_{z=1}^{N} \log p(\mathbf{x}_z, M) = \sum_{z=1}^{N} \log \left\{ \sum_{i=1}^{K} \pi_i \, p(\mathbf{x}_z|\mu_i, \Sigma_i) \right\}$.

see also [27]. It computes probabilities of cluster memberships based on the assumed probability distribution models. Then, the goal is to maximize the overall probability or the likelihood of the data, given the (final) clusters. Usually, one hypothesizes the number of clusters and the Gaussian cluster models [265], yet, other probability distributions, like multinomial, can be used. Note that in contrast to the k-means algorithm, the EM-clustering uses 'soft' assignments (memberships) to the clusters.

The EM-clustering can also be interpreted as an approach, where starting from an initial partition to k clusters, a normal density based classifier is trained, changing the given assignments accordingly. This proceeds iteratively until stable assignments are achieved. A generic EM-clustering can be considered when any probabilistic classifier is employed instead of the normal density based classifier. One can go even further, by using an arbitrary classifier (e.g. logistic discriminant, decision tree, support vector classifier) with crisp label assignments. We will denote this approach as the *classifier-clustering*, in particular, the NMC-clustering, NQC-clustering, etc. (Note that the k-means is in fact the NMC-clustering.) One must realize and take some precautions in judging the obtained partition, because the results of the EM- and classifier-clustering depend on the initialization. The initial labels are often provided by an another clustering algorithm such as a hierarchical clustering.

Cluster validity. Finding the right number of clusters to retain is often difficult, since the answer depends on the scale (size of clusters) one is interested in. One usually chooses a criterion capable of recognizing the 'correct' number of clusters, where an optimum is reached, when evaluated for a growing number of clusters. If the true cluster labels are known, the evaluation measures include the confusion matrix, classification accuracy, average entropy or mutual information. Some cluster validity proposals can be found in [25, 128, 132, 181, 184, 396].

In probabilistic approaches to clustering, the likelihood-ratio measures are used. In the framework of the k-means and EM-clustering, new patterns can be assigned to the known clusters, so the classifiers are indirectly designed. The clustering can proceed in an N-fold cross-validation fashion to determine the the average distance to the cluster means (in terms of conceptual dissimilarity) for the k-means or the average log-likelihood for the EM-clustering. Such values may indicate the right number of clusters (according to the assumed cluster distributions). For hierarchical approaches (except for the centroid linkage), the change in the dissimilarity between the merged clusters (the gap) can be inspected (since it will grow). A large value indicates that two dissimilar clusters are merged, as further exploited by Fred and Leitão [132].

Cluster ensembles. Ideally, a clustering algorithm should posses a number of useful properties, such as an ability to discover clusters of arbitrary shapes, easily determined input parameters, handling noise and outliers, an ability to find the right number of clusters, interpretability and usability. However, the basic difficulty of clustering algorithms lies in their limitation to find clusters of specific shapes or structures (e.g. hyper-spherically shaped), failing to reveal clusters whose shapes do not match the assumed models. To address the above-mentioned requirements more adequately, cluster ensembles are an appealing alternative. Indeed, there has been a growing interest in studying cluster ensembles to discover clusters of variable shapes and to improve the robustness of the clustering techniques. Examples of such a work can be found in [7, 129–131, 382–384, 397]. An interesting approach is to transform data partitions resulting from various clustering methods into the co-associations [129, 131] encoding the co-occurrences of pairs of objects in the same cluster. In fact, a new higher-level similarity representation is created, where each similarity value is the numerical vote towards gathering a pair of objects together. The final grouping is then derived from such similarities e.g. by a single linkage [129, 131].

Other views on clustering. The division of the clustering techniques into hierarchical and partitioning methods is not the only possibility. Another way to inspect the arsenal of clustering approaches is to consider them as hard and fuzzy algorithms. In a hard clustering process, each object is allocated to a single cluster, which is indicated by a crisp label. A fuzzy clustering method [24, 198] assigns to each object degrees of cluster memberships. The final crisp result is obtained by assigning objects to the clusters which yield a maximum membership degree.

One may also distinguish deterministic and stochastic approaches, applicable to criterion-based minimization techniques. The deterministic methods are often greedy descent approaches and EM, while stochastic methods often rely on simulated annealing or mean field annealing [46, 211, 310]. Yet, another possibility are the incremental versus non-incremental algorithms. The former methods are especially important for the organization of huge data sets when they are designed to be efficient with respect to both the execution time and memory.

7.1.2 Clustering techniques on dissimilarity representations

Here, we will mention some clustering techniques derived for dissimilarity representations. This is not meant as a thorough investigation of the subject, rather as a brief survey and adaption of the basic existing techniques. The dissimilarity representations will now be interpreted in three frameworks: neighborhood-based, embedded spaces and dissimilarity spaces. For the sake of simplicity, symmetric representations $D(R, R)$, where $R = \{p_1, p_2, \ldots, p_n\}$ are considered.

Neighborhood relations. The rationale is to group objects characterized by small dissimilarities to other objects or which are in a close neighborhood of some selected representatives. Let C_k and C_l be two clusters of the cardinalities N_k and N_l, respectively and let ρ_{kl} be the dissimilarity between them. Concerning the hierarchical clustering, the basic criteria for the agglomerative methods are:

- **Single linkage (SL)**. The dissimilarity ρ_{kl} between two clusters is determined by the dissimilarity between their nearest neighbors, i.e. $\rho_{kl} = \min_{p_i \in C_k} \min_{p_j \in C_l} d(p_i, p_j)$. This rule emphasizes cluster connectedness, resulting in elongated clusters.
- **Complete linkage (CL)**. The dissimilarity ρ_{kl} is defined by the furthest neighbors of the two clusters, i.e. $\rho_{kl} = \max_{p_i \in C_k} \max_{p_j \in C_l} d(p_i, p_j)$. This usually performs well when the objects form naturally distinct clouds; since it emphasizes the compactness. It is inappropriate if the clusters are somehow elongated or of a chain type.
- **Average linkage (AL).** The dissimilarity ρ_{kl} becomes the average between-cluster dissimilarity, i.e. $\rho_{kl} = \frac{1}{N_k N_l} \sum_{p_i \in C_k} \sum_{p_j \in C_l} d(p_i, p_j)$. This performs well in both cases, when the objects form natural distinct clouds and when they form elongated clusters. It tends to produce clusters of a similar variance.
- **Density linkage.** This criterion computes a new dissimilarity d^* based on the density estimates and adjacencies, which is further used by the single linkage clustering. For instance, in the k-nearest neighbor approach, the estimated density $f(p_i)$ at p_i is the number of objects within the k-ball divided by its volume. The new d^* is then computed as $d^*(p_i, p_j) = \frac{1}{2}(\frac{1}{f(p_i)} + \frac{1}{f(p_j)})$ if $d(p_i, p_j) \leq \max\{d_{k-NN}(p_i), d_{k-NN}(p_j)\}$ and ∞, otherwise.

Concerning the implementation issues, a general recurrence formula for hierarchical clustering methods has been developed [119, 238]. It is useful since at any level of the hierarchy, the dissimilarity between newly created cluster and other clusters can be computed from the current grouping.

The methods mentioned above work directly on the dissimilarities. Two other popular criteria, the centroid linkage and the Ward linkage [119] require a Euclidean feature space representation, since they work with the estimated cluster means. We may, however, propose an extension for arbitrary symmetric dissimilarity representations that indirectly makes use of the centroids in the embedded

pseudo-Euclidean space:

- **Generalized centroid linkage (GCL).** In the centroid linkage, the dissimilarity between clusters is the square Euclidean distance between their mean vectors. Although our GCL extension refers to the embedded pseudo-Euclidean space, such an embedding does not need to be performed explicitly. Making use of Corollary 4.2 and formula (4.23), the square pseudo-Euclidean distance between the cluster means can be approximated by $\rho_{kl} = \rho_{avr}(C_k, C_l) - \frac{1}{2}\rho_{avr}(C_k, C_k) - \frac{1}{2}\rho_{avr}(C_l, C_l)$ where ρ_{avr} is the average square dissimilarity $\rho_{avr}(C_k, C_l) = \frac{1}{N_k N_l} \sum_{p_i \in C_k} \sum_{p_j \in C_l} d^2(p_i, p_j)$. This becomes our merging criterion.
- **Generalized Ward linkage (GWL).** In the Ward linkage, in each step, the two clusters are merged that give the smallest increase in the within-cluster sum of squares, which is the sum of the squared Euclidean distances between vectors and their cluster means. This tends to create clusters of similar sizes. In our extension, the pseudo-Euclidean distance of the embedded cluster configuration can be used. Based on formula (4.20), the pseudo-Euclidean distance of a single point to the mean of cluster C_k in an embedded space is determined as $d^2(p_i, \mathrm{me}_k) = \frac{1}{N_k} \sum_{p_j \in C_k} d^2(p_i, p_j) - \frac{1}{2\,N_k^2} \sum_{p_z \in C_k} \sum_{p_t \in C_k} d^2(p_z, p_t)$. Hence, we can propose the GWL criterion relying on the estimated within-cluster sum of squares as $\sum_{p_i \in C_k} d^2(p_i, \mathrm{me}_k) = \frac{1}{2\,N_k} \sum_{p_z \in > C_k} \sum_{p_t \in C_k} d^2(p_z, p_t)$.

Remember that the dendogram built in an agglomerative clustering process is an additive tree (or an ultrametric tree as e.g. in the case of single linkage) approximating the original dissimilarity matrix; this has been introduced in sections 3.1.2 and 6.3.

Concerning the partition methods, the k-centres [427] and the mode-seeking [63] will be described.

k-centres. This technique works on $D(R, R)$ directly. It looks for k objects from R such that they are approximately evenly distributed with respect to the dissimilarity information. The algorithm proceeds as follows:

1. Select an initial set $J := \{p_1^{(j)}, p_2^{(j)}, \ldots, p_k^{(j)}\}$ of k objects, e.g. randomly chosen from R.
2. For each object $p_z \in R$ find its nearest neighbor in J. Let J_i, $i = 1, 2, \ldots, k$, be a subset of R consisting of objects that yield the same nearest neighbor $p_i^{(j)}$ in J. This means that $R = \cup_{i=1}^{k} J_i$.
3. For each J_i find its center c_i, i.e. an object in J_i for which the maximum distance to all other objects in J_i is minimum (this value is called the radius of J_i).
4. For each center c_i, if $c_i \neq p_i^{(j)}$, then replace $p_i^{(j)}$ by c_i in J. If any replacement is done, then return to 2, otherwise STOP.

Except for the step 3, this routine is identical to the k-means, performed in a vector space. The result of the k-centres procedure heavily depends on the initialization. For that reason we use it with some precautions. To determine the set J of k objects, we start from a chosen center for the entire set and then more centers are gradually added. At any point, a group of objects belongs to each center. J is enlarged by splitting the group of the largest radius into two and replacing its center by two other members of that group. This stops, when k centers are determined. The entire procedure is repeated M times, (say 50) resulting in M potential sets from which one yielding the minimum of the largest final subset radius is selected. Note that if we continue with the splits, the k-centres can also be seen as a hierarchical divisive method.

Mode-seeking. The mode-seeking method [63] focuses on the modes in dissimilarity data determined in the specified neighborhood size of s. The algorithm proceeds as follows:

1. Set a relative neighborhood size as an integer $s > 1$.
2. For each object $p_i \in R$ find the dissimilarity $d(p_i, nn_s(p_i))$ to its s-th neighbor.

3. Find a set J of all $p_j \in R$ for which $d(p_j, nn_s(p_j))$ is minimum within its set of s neighbors.

The objects from the set J are the estimated modes of the class distribution in terms of the given dissimilarities. They are used to constitute the modes. The final number of clusters k depends on the choice of s. The larger s, the smaller k.

Embedded spaces. The symmetric dissimilarity representations can be represented in the complete or approximated embedded spaces, where the standard partition methods, such as the k-means and the classifier-clustering can be used. Here, embedded spaces are understood broadly as either pseudo-Euclidean spaces or Euclidean spaces. The embedding may focus on the preservation of all original dissimilarities or on the preservation of the dissimilarities only in local neighborhoods. Such spaces are determined by the use of multidimensional scaling methods or some other techniques, such as Isomap or local linear embedding, described in section 6.2. In fact, by performing an approximate embedding, some information, possibly reflecting the noise in the data, is neglected. This might be seen as a purification of the dissimilarity information[2]. Here, we will use an approximate linear embedding to a pseudo-Euclidean space.

Dissimilarity spaces. In a dissimilarity space, traditional clustering algorithms can be applied. From the efficiency (computational) point of view and from the representational point of view (using only informative objects as the representatives), it is beneficial to use a reduced representation $D(R, R_r)$, where $R_r \subset R$. The cardinality of R_r can be specified as e.g. $5 - 20\%$ of $|R|$ (depending also on the hypothesized number of clusters to be retrieved) or as the estimated intrinsic dimensionality of $D(R, R)$. R_r can be selected randomly or by using the k-centres or mode-seeking procedures. Additionally, to ensure that the objects in R_r convey various dissimilarity information, they can be chosen in the following way. First for each object in R, the average dissimilarity to all other objects is computed resulting in a sequence $a_i := a(p_i) = \frac{1}{|R|} \sum_{p_z \in R} D(p_i, p_z)$. The sequence is then sorted in a decreasing order and the objects are then selected, which correspond to each q-th sorted value. First objects can be disregarded as possible outliers. We will refer to it as the *sparse average* selection. Alternatively, one may also retrieve principal components from the dissimilarity space (treating it as a usual vector space) reflecting e.g. 90% of the variance. We will call it a PCA-dissimilarity space.

The generic EM-clustering or classifier-clustering approach in a dissimilarity space is advantageous for reasonably sampled clusters of significantly different radii (i.e. the maximum dissimilarity between the objects in a cluster) or where at least one cluster is very sparse in comparison to other compact clusters. In such cases, the neighborhood-based clustering approaches (e.g. AL or CL hierarchical clustering or k-centres) tend to fail. In a (reduced) dissimilarity space, clusters might be well separable. Note, however, that if the dissimilarity between two objects does not capture the cluster characteristic, the dissimilarity space will not help in detecting such clusters[3]. So, a possible solution is to consider a nonlinear monotonic transformation of the dissimilarities, such as a sigmoid $f_{\text{sigm}}(x) = 2/(1 + e^{-x/s}) - 1$, applied in an element-wise way. Such a transformation will change the neighborhoods perceived in the dissimilarity space, although, it will not influence the methods based on the neighborhoods relations directly such as hierarchical methods or the k-centres.

[2] It is also possible to re-compute the dissimilarity representation derived from the approximate embedding. Hence, the embedding can be treated as a de-noising step in obtaining a more discriminative dissimilarity representation, which can be further used by the neighborhood-based clustering approaches.

[3] Imagine e.g. artificial banana data in 2D with a Euclidean distance representation, Fig. A.3. The curved banana clusters will be even more pronounced in a distance space, so no EM-clustering algorithm would be able to find such a structure without a perfect initialization. To detect curved structures, the dissimilarities should be recomputed appropriately, e.g. along the path.

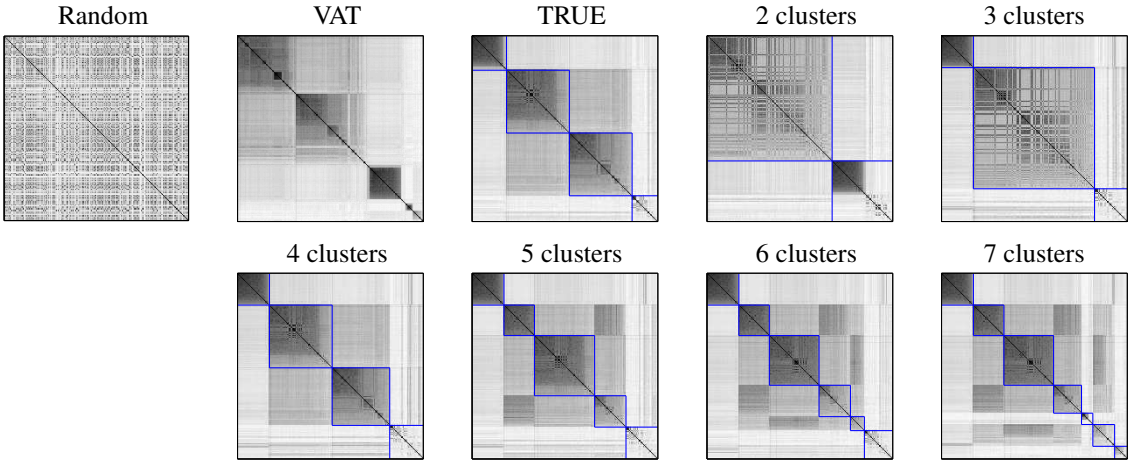


Fig. 7.1: Intensity images of the permuted protein dissimilarity representation. The upper leftmost intensity image corresponds to a random representation of the original data. The second upper intensity image is an implementation of the visual assessment cluster tendency algorithm (VAT) [23], which in fact reorders the data items with respect to the within-cluster dissimilarity. The third upper intensity image shows the true classes present in the data. The remaining intensity images are created based on the assignments to a number of clusters varying from 2 to 7. The data objects are grouped by the NQC-clustering (mixture-of-Gaussians clustering) in the PCA-dissimilarity space. To make an intensity image, the detected clusters are presented in the order of a growing within-cluster average dissimilarity, which is a simple visualization proposed by us below. From the visualization of the two-cluster clustering, one may already detect two more clusters present.

Related work. An interesting approach to a general proximity-based (neighborhood-based) partitioning, both partitioning and hierarchical, has been advocated by Buhmann, Hofmann and Puzicha, where the clustering is formulated as a combinatorial optimization problem; see e.g. [45, 46, 197, 310]. The authors specified an objective function, incorporating a suitably weighted average of the within-cluster and between-cluster dissimilarities, and derived some optimization heuristics based on annealing. Another idea has been proposed in [126] discussing a *path-based pairwise clustering*, which emphasizes the within-cluster connectivity by the use of graph methods. Two objects are considered as similar if there exists a within-cluster path between them without any edge of a large dissimilarity. As a result, a new dissimilarity is developed that is further used for grouping.

Recently, another proximity-based algorithm, called *evidential clustering* (EVCLUS) has been proposed by Denœux and Masson in [85]. The method relies on the evidence theory and attach to each object a mass function such that the degree of conflict between the masses of any two objects reflect their proximity, which is measured by a suitable stress function from the metric multidimensional scaling (see section 3.4.2). Practically, it relies on the optimization of the stress function penalized by some entropy measure, added to prevent the resulting model from being too complex.

The applications of spectral graph theory to the clustering problem resulted in *spectral clustering* algorithms; see e.g. [14, 284]. Such procedures rely on finding the eigenvectors of some similarity matrix derived from a feature-based representation of a set of objects. This is a suitably scaled Gaussian similarity matrix (based on Euclidean distances). The interesting property of spectral clustering is the ability to pull out non-convex or even disjoint clusters. In the final stage, however, partitioning algorithms perform the final grouping. The specification of σ in the Gaussian function, as well, as the number of clusters are the main questions to be solved. In fact, such algorithms determine a specific embedded Euclidean space of an appropriately transformed similarity representation used later for traditional partitioning clustering methods.

Visual cluster validity. Since clustering is subjective, one must not forget to visually inspect the results. The most appealing approach is to represent the dissimilarity representation D as an intensity

image, in which each pixel value corresponds to a dissimilarity between a pair of objects. To observe the detected clusters, D should be permuted according to the cluster assignments. If a fuzzy or soft clustering method is used, then the objects within a cluster can be sorted based on their membership values. To keep it simple and general, we propose to permute the objects within one cluster based on their growing average dissimilarity to all other objects. Assume that P is the final permutation matrix. Hence, one needs to display PDP as an intensity image. Example displays for a growing number of clusters is shown in Fig. 7.1, where the results for the protein dissimilarity data, grouped by the NQC-clustering in the PCA-dissimilarity space are presented; see next section for details. A more profound way to visualize the cluster validity methods was proposed in [23, 192]. Additionally, one may analyze the clustering results by labeling the objects accordingly in the 2- or 3-dimensional spatial maps obtained by multidimensional scaling techniques.

7.1.3 Clustering examples of dissimilarity representations

Four dissimilarity data sets are considered here for which true labels are known: the 400×400 Euclidean distance representation of the artificial two-class ringnorm data describing two somewhat overlapping Gaussian clouds in a 20-dimensional space, 65×65 cat-cortex dissimilarity data (four classes), 213×213 protein dissimilarity data (four classes) and 400×400 newsgroup correlation-based dissimilarity data *News-cor2* (four classes); see Appendix A for the data description. The protein dissimilarity data set is nearly Euclidean, while the cat-cortex data and the newsgroup data are non-Euclidean; see also Our assumption is that the dissimilarity measure used is able to capture the underlying cluster difference, so we should perceive the clusters as Gaussian-type clouds either in embedded or dissimilarity spaces. We assume that the number of clusters is known and we will try to find out whether the true classes given in the data can be detected.

The following clustering methods are used: evidence clustering (EVCLUS) [85], the standard hierarchical clustering such as single linkage (SL), average linkage (AL) and complete linkage (CL), the k-centres, mode-seeking and the NQC-clustering (which is a Gaussian-of-mixture EM clustering for soft labels) both in the pseudo-Euclidean embedded space and in the dissimilarity space. The NQC has been chosen, since it is an appropriate classifier for detecting all types of Gaussian-like clusters. To avoid singular covariance matrices for small clusters, the NQC is slightly regularized with $\lambda = 10^{-6}$; see section 4.4.1 for details. The dimensionality of an embedded space is chosen based on a small number of significant eigenvalues determined in the embedding. The dissimilarity space $D(R, R)$ is reduced to $D(R, R_r)$ by the sparse average selection, as described above. We will denote this procedure as the NQC-clustering in DS. Another possibility is to extract the largest principal components in the dissimilarity space. Here, as default, the dimensionality is chosen based on the preservation of 90% of the total variance. We will denote this approach as the NQC-clustering in PCA-DS. If a square dissimilarity is used instead, it will be indicated by DS^{*2}.

The EVCLUS has been used here since it was applied to the cat-cortex and protein data in [85], where the authors claimed that their fuzzy-like method performed the same or much better than other state-of-the-art fuzzy techniques. Since EVCLUS is initialization-sensitive, we followed the authors' suggestions by running their code [84] 50 times and determining the final result as the one for which their penalized stress objective function was minimum. In this way, we compare our results to a good method.

The NQC-clustering depends on the initial labeling, as well. Therefore, a criterion is needed for the selection of the final result. Let k be the number of clusters and n_i be the i-th cluster cardinality, with n being the total number of objects. Inspired by [310], we propose to use the following goodness-

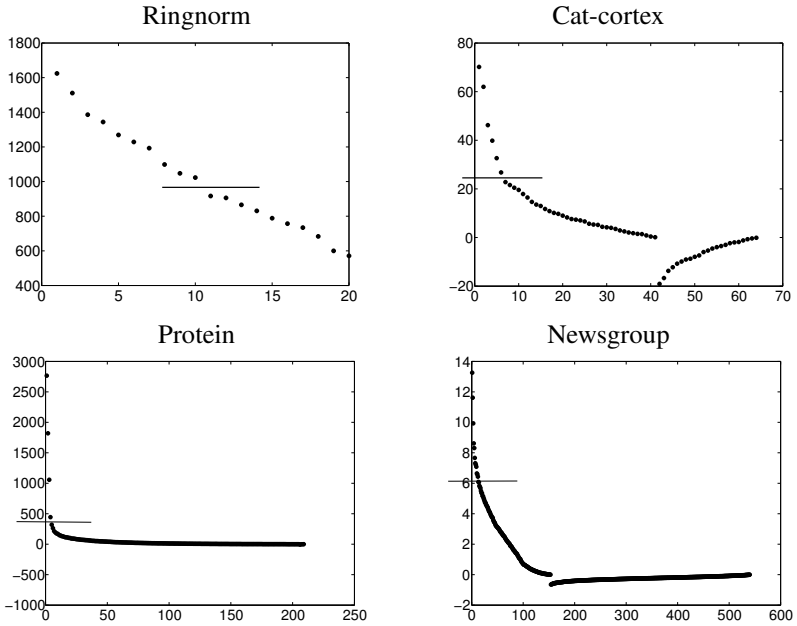


Fig. 7.2: Eigenvalues determined in the linear embeddings of the four dissimilarity data. The number of most significant eigenvalues describes the effective intrinsic dimensionality, here denoted by the black lines. For the Ringnorm distance data, only first 20 non-zero eigenvalues are shown.

of-clustering measure J_{GOC} relating the cluster separability and cluster compactness as:

$$J_{\text{GOC}} = \frac{\sum_{i=1}^{k} n_i \sum_{j \neq i} \frac{n_i}{n - n_i} A_{ij}}{2 \sum_{i=1}^{k} n_i A_{ii}}, \tag{7.1}$$

where A_{ij} is the average dissimilarity between the i-th and j-th clusters. So, A_{ii} is the average within i-th cluster dissimilarity. In our approach, the NQC-clustering is run 50 times in chosen embedded or dissimilarity spaces. The final result is chosen as the one corresponding to the maximum of J_{GOC}.

Other clustering methods provide deterministic results. Only in case of mode-seeking, a proper neighborhood size should be detected to retrieve a specified number of clusters.

In our clustering approaches, the number of clusters k is assumed to be known. Concerning the k-centres, hierarchical clustering and the mode-seek algorithms, a larger number of clusters is sometimes retrieved, since these methods suffer either from outliers (objects with large dissimilarities) or have difficulties to accommodate sparse clusters. For the NQC-clustering in the embedded and PCA-dissimilarity spaces, we notice that the results depend on the space dimensionality. In our understanding, the dimensionality should be chosen close to the effective intrinsic dimensionality of the problem, i.e. the smallest dimensionality, which can reveal the structure in the data (note that this reasoning is valid for clustering, but not necessarily for classification). Since in an embedded space and a PCA-dissimilarity space, all the determined dimensions depend on the dissimilarities to *all* objects, a small dimensionality is preferred.

For each dissimilarity data set, all eigenvalues of the pseudo-Euclidean embedding have been found and plotted, as observed in Fig. 7.2. The dimensionality of an embedded space has been chosen according to the number of *very* significant eigenvalues, understood as eigenvalues being apart from the 'continuous stream' of eigenvalues (as judged visually by us). So, the effective intrinsic dimensionality is chosen to be: 10 for the ringnorm data 6 for the cat-cortex data, 4 for the protein data and 12 for the newsgroup *News-cor2* data. We admit that this might not be the best approach, since it is not automatic, yet, it makes sense intuitively. We need to develop an automatic procedure, based e.g. on a spline interpolation of the eigenvalue plot for which the change in the speed of

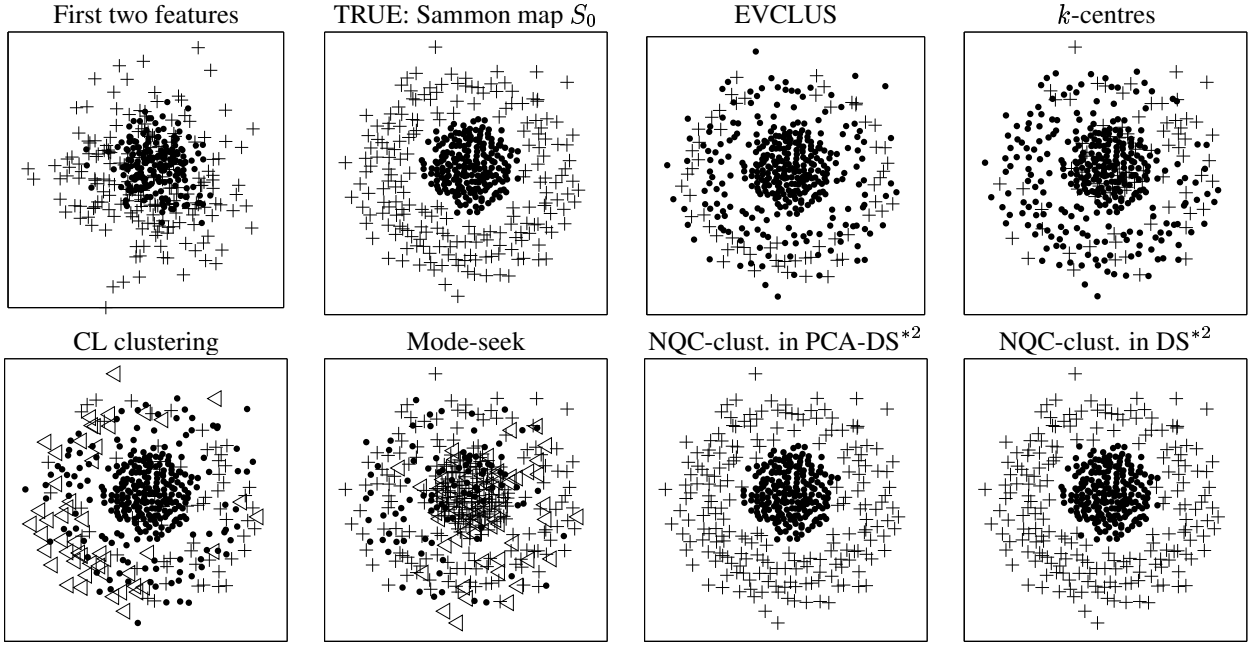


Fig. 7.3: Clustering results of the ringnorm Euclidean distance data as visualized by a proper labeling of the 2D Sammon map S_0 obtained on the unlabeled data. The objects are labeled according to the specified clustering algorithms. The number of clusters is fixed to 2, however for the CL and mode-seek clusterings, the results for three clusters are presented, because the two-cluster groupings find one cluster of a few objects only. TRUE stands for the true class labels. Note that the Sammon map is only a visualization of the dimensionality effects in the original 20-dimensional space. The first two features of the Gaussian clusters are shown in the top, leftmost plot. See text for details.

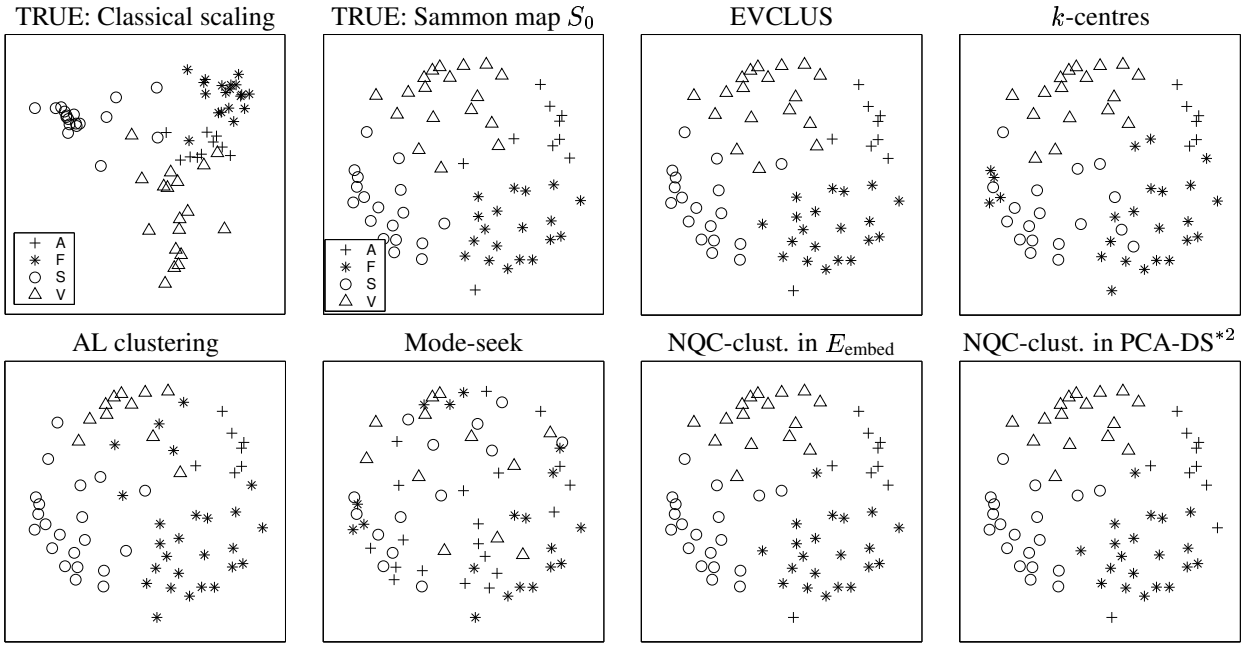


Fig. 7.4: Clustering results of the cat-cortex dissimilarity data as visualized by a proper labeling of the 2D Sammon map S_0 obtained on the unlabeled data. The objects are labeled according to the specified clustering algorithms. The number of clusters is fixed to 4. TRUE stands for the true class labels. See text for details.

its decline (from a fast to moderate steepness) should be determined. This requires some future attention. To simplify our procedures, the same dimensionality, as reported above, was used for the PCA-dissimilarity space (which might be not optimal at all).

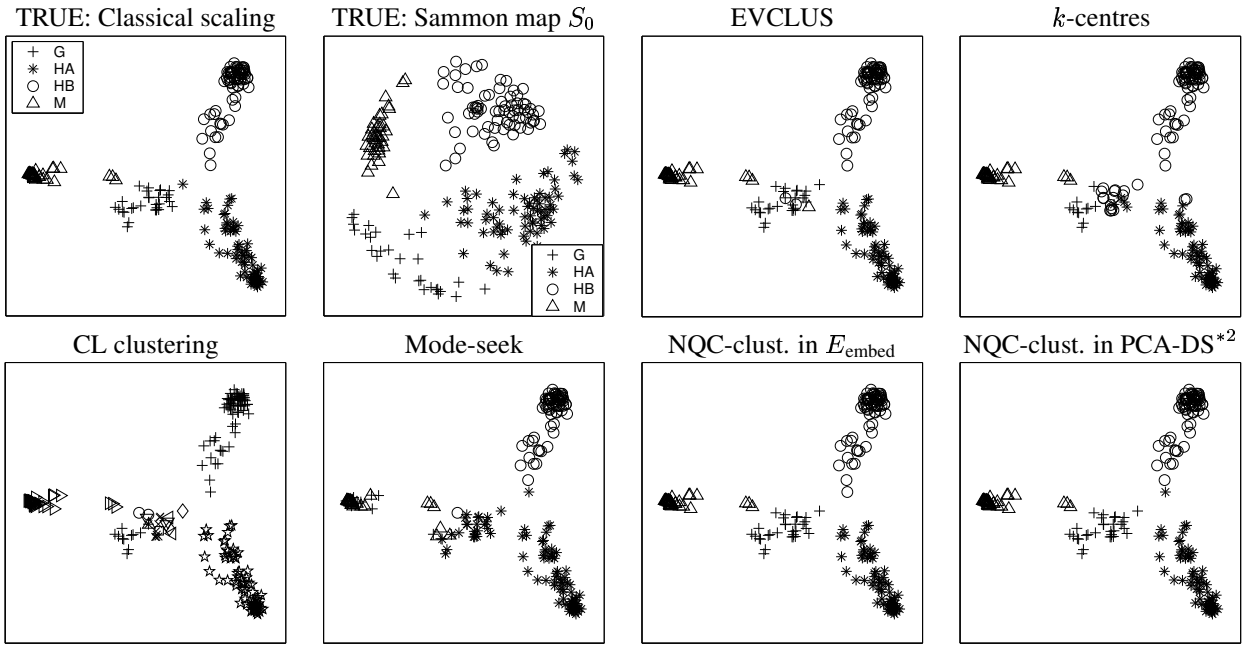


Fig. 7.5: Clustering results of the protein dissimilarity data as visualized by a proper labeling of the 2D classical scaling representation obtained on the unlabeled data. The objects are labeled according to the specified clustering algorithms. The number of clusters is fixed to 4, however for the CL clustering, the results for nine clusters are presented, because the groupings found for less clusters, detect two clusters only. TRUE stands for the true class labels. See text for details.

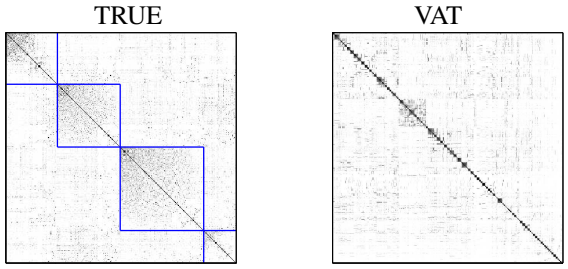


Fig. 7.6: Intensity images of the newsgroup *News-cor2* dissimilarity data. On the left, the data objects are permuted according to the true cluster memberships. On the right, the visual assessment cluster tendency (VAT) [23], meant to detect clusters in the dissimilarity data is shown. These intensity images suggest that there is no strong structure in the dissimilarity data.

Concerning the ringnorm Euclidean distance data, the two Gaussian clusters are not discovered by the EM-clustering algorithms in the initial Euclidean space. This is caused by the sparseness of one of the clusters. Presumably, the path-based clustering or the spectral clustering should be able to detect these clouds. In a dissimilarity space, however, the clusters are better separated, since the distances to the objects from the compact cloud are discriminative for both clusters. The clustering results of some algorithms are presented in Fig. 7.3.

Cat-cortex dissimilarity data set is difficult, since it is not only small, but the dissimilarities are ordinal values. This makes it hard to build a NQC in both the embedded and the dissimilarity spaces. In fact, the dissimilarities should be de-noised or smoothed out. Since the dissimilarities are not very discriminative (only five different dissimilarity values are used, integers from 0 to 4), the task becomes difficult. Some of the clustering results are illustrated in Fig. 7.4.

Protein dissimilarity data are reasonably well clustered, so the clusters can be recovered. Some of the results are shown in Fig. 7.5.

Newsgroup dissimilarity data are defined on weak and poor word occurrence vectors. The dissimi-

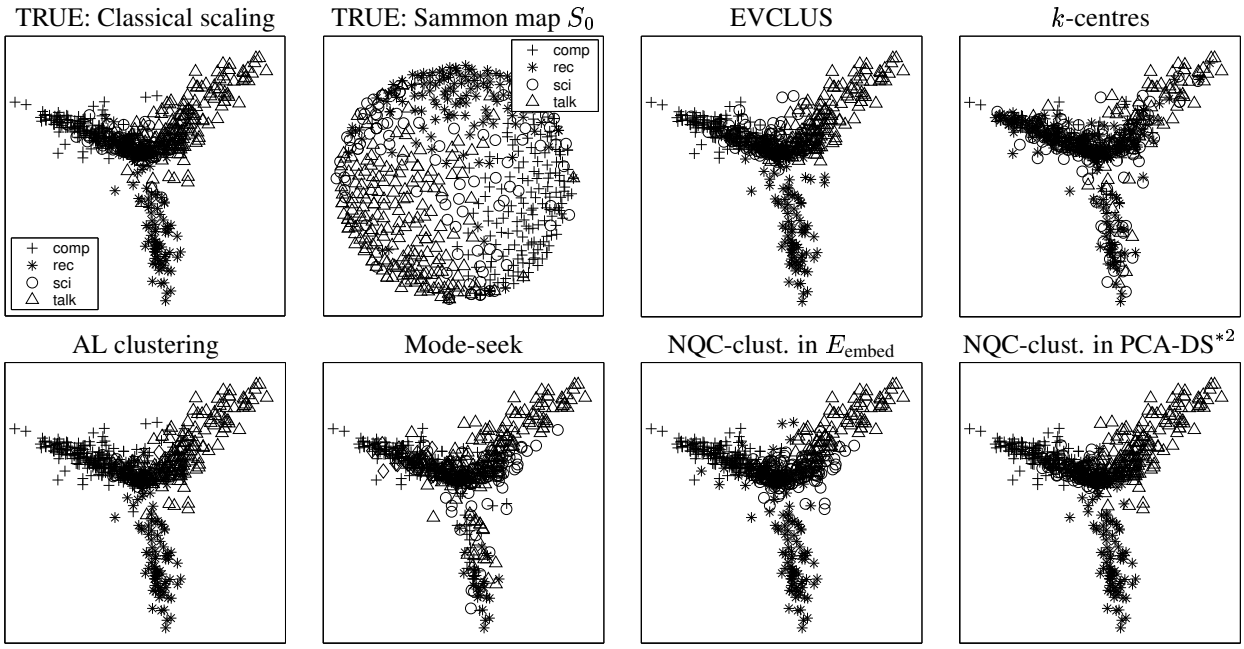


Fig. 7.7: Clustering results of the newsgroup *News-cor2* dissimilarity data as visualized by a proper labeling of the 2D classical scaling representation obtained on the unlabeled data. The objects are labeled according to the specified clustering algorithms. The number of clusters is fixed to 4. TRUE stands for the true class labels.

larities are not very discriminative between the clusters, so it is difficult to discover them properly. The within-cluster dissimilarity of the 'sci.*' news group are of the same order as the between-cluster dissimilarities; see Fig. 7.6. Majority of these objects is assigned to other clusters. Some clustering results are illustrated in Fig. 7.7.

The overall numerical results are shown in Table 7.1. It can be observed that indeed, EVCLUS performs well, provided that a suitable trade-off parameter is chosen. If the parameter deviates from the optimal value, very bad results are found. Additional disadvantage of EVCLUS is the high computational burden, which for the protein dissimilarity data the 50 groupings takes about 90 minutes, while the NQC-clustering (with the embedding included) takes about 0.5 min. The authors of EVCLUS reported in [85] that their algorithm competes with other state-of-art fuzzy clustering algorithms. We must, therefore, report that our handcrafted NQC-clustering approach in the PCA-dissimilarity space (or in an embedded space) performs similarly or better than EVCLUS (especially for the ringnorm data). Although our results are preliminary, they indicate that possibly more can be gained if the methods will be improved further by designing an automatic selection of the parameters.

7.2 Intrinsic dimensionality

If a certain phenomenon can be described (or if it is generated) by k independent variables, then its intrinsic dimensionality (ID) is k. In practice, however, due to noise and imprecision in measurements or some other uncontrolled factors, such a phenomenon may seem to have more variables. If all these factors are not 'too strong' that they completely disturb the original phenomenon, one should be able to re-discover the proper number of variables. So, the intrinsic dimensionality is a minimum number of variables that explains the phenomenon in a satisfactory way. In pattern recognition, one usually discusses the intrinsic dimensionality with respect to a collection of data vectors in a feature space. The intrinsic dimensionality can then be defined as the minimum number

Table 7.1: Clustering results compared with respect to the true classes for four dissimilarity data. The numbers below describe the absolute number of mismatches (hence a small number in the clustering procedures indicates a faithful grouping).

Clustering method	Ringnorm	Cat-cortex	Protein	Newsgroup
TRUE	400	65	213	600
EVCLUS	137	2	5	190
Hierarchical clustering	159	10	21	195
K-centres	176	14	22	402
Mode-seek	230	38	41	348
NQC-clustering in PE	168	4	1	229
NQC-clustering in PCA-DS	2	2	4	253
NQC-clustering in PCA-DS^{*2}	4	5	2	199
NQC-clustering in reduced DS^{*2}	4	23	1	298

of features needed to obtain a similar classification performance as by using the total number of features. In a geometrical sense, the ID can be defined as a dimension of a manifold that approximately (due to noise) embeds the data. In fact, the estimated ID of a sample depends on a chosen criterion (e.g. whether one searches a linear or nonlinear manifold) and it may vary from one criterion to another. Therefore, the estimated ID is relative for the task. Usually, the determination of the ID is done by the use of (nonlinear) feature reduction techniques, either by a selection or extraction.

For the study of dissimilarity representations, one may choose an embedding method and estimate the ID appropriately or perform the dimensionality reduction of the dissimilarity space. Some of such techniques were briefly explained in chapter 6, where the projections of the dissimilarity data were considered. Here, we will focus on two linear techniques: the pseudo-Euclidean embedding of a symmetric $D(R,R)$ and PCA applied in a dissimilarity space. In the embedding process, the number of informative eigenvalues describes the ID. If the data are labeled, then the intrinsic dimensionality might be judged for each class separately, as well as for the complete set. For unlabeled data, one may first determine some meaningful groups and then proceed as if with the labeled case. Using this particular embedding, the estimated ID cannot be larger than the number of objects considered.

Statistical estimation of the intrinsic dimensionality for a Gaussian sample. Assume first a Euclidean m-dimensional space and a variable $\mathcal{X}$, normally distributed with a zero mean vector, zero covariances and equal variances $\frac{\sigma^2}{2}$ in all dimensions. Hence, $\mathcal{X} \sim \mathcal{N}(\mathbf{0}, \frac{\sigma^2}{2} I)$. Consider now the square Euclidean distance variable τ which for a particular pair two realizations $\mathbf{x}_k$ and $\mathbf{x}_l$ of $\mathcal{X}$ is expressed as $\tau_{kl} = \sum_{i=1}^{m} (x_{ki} - x_{li})^2$. Since $y = \frac{\tau}{\sigma^2}$ is χ^2_m distributed[4], then after straightforward calculations one obtains that $E[y] = m$ and $E[y^2] = (m^2 + 2m)$, where $E[\cdot]$ denotes the expectation. Hence $E[\tau] = m\sigma^2$ and similarly $E[\tau^2] = (m^2 + 2m)\sigma^4$. Using these results, we find that

$$2\,\frac{(E[\tau])^2}{E[\tau^2] - (E[\tau])^2} = 2\,\frac{m^2\sigma^4}{(m^2+2m)\sigma^4 - m^2\sigma^4} = 2\,\frac{m^2\sigma^4}{2\,m\sigma^4} = m \quad \text{and} \quad \frac{E[\tau]}{m} = \sigma^2. \tag{7.2}$$

In this way, both the dimensionality m and the variance of the spherical Gaussian variable $\mathcal{X}$ can be estimated from the square Euclidean distance variable τ only. Given a sample of $\mathcal{X}$, i.e. a finite

[4] A basic statistical fact is that given m independent one-dimensional variables $\mathcal{Y}_i \sim \mathcal{N}(0,1)$, the variable $\mathcal{Y} = \sum_{i=1}^{m} \mathcal{Y}_i^2$ is χ^2_m distributed with m degrees of freedom (df). The probability density function of χ^2_m is $p_{\chi^2}(y) = \frac{y^{m/2-1} e^{-y/2}}{2^{m/2}\,\Gamma(m/2)}$, where $\Gamma(y) = \int_0^\infty t^{y-1} e^{-t} dt$. If $\mathcal{Z}_i \sim \mathcal{N}(0, \sigma^2)$, then $\mathcal{Z} = \sum_{i=1}^{m} \mathcal{Z}_i^2 = \sigma^2 \sum_{i=1}^{m} \mathcal{Y}_i^2 = \sigma^2 \mathcal{Y}$. Consequently, $\mathcal{Z}$ is $\sigma^2 \chi^2_m$ distributed. Consider now two independent m-dimensional variables $\mathcal{X}, \mathcal{Y} \sim \mathcal{N}(\mathbf{0}, \frac{\sigma^2}{2} I)$, which means that for one dimensional variables $\mathcal{X}_i$ and $\mathcal{Y}_i$, we have $\mathcal{X}_i, \mathcal{Y}_i \sim \mathcal{N}(0, \frac{\sigma^2}{2})$. The square Euclidean distance τ between $\mathcal{X}$ and $\mathcal{Y}$ can be expressed as $\tau = \sum_{i=1}^{m} (\mathcal{X}_i - \mathcal{Y}_i)^2 = \sum_{i=1}^{m} \mathcal{X}_i^2 + \sum_{i=1}^{m} \mathcal{Y}_i^2 + 2 \sum_{i=1}^{m} \mathcal{X}_i \mathcal{Y}_i$. Therefore, τ is $2\,\frac{\sigma^2}{2} \chi^2_m = \chi^2_m$-distributed, since the variables $\mathcal{X}_i$ and $\mathcal{Y}_i$ are independent.

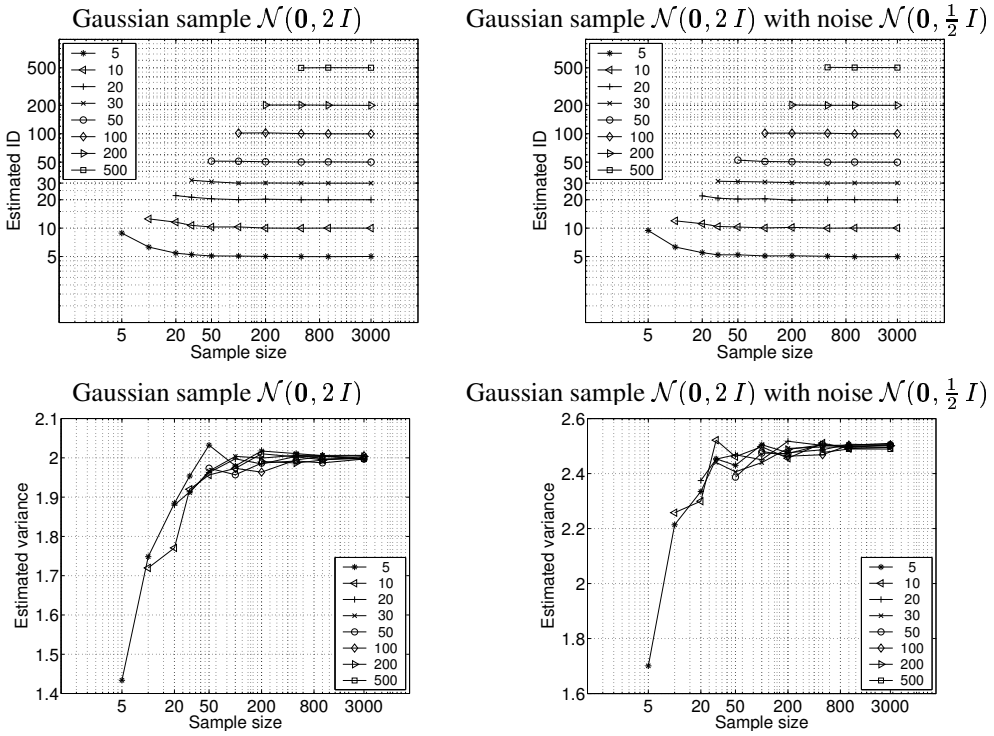


Fig. 7.8: Estimated intrinsic dimensionality (top row) and variance (bottom row) for various Gaussian samples. Different marks correspond to different dimensionalities, as described in the legend.

set of examples $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ and the corresponding square Euclidean distance matrix D^{*2}, $E[\tau]$ and $E[\tau^2]$ can be expressed as $E[\tau] = \frac{1}{n^2-n}\mathbf{1}^T D^{*2}\mathbf{1}$ and $E[\tau^2] = \frac{1}{n^2-n}\mathbf{1}^T D^{*4}\mathbf{1}$. Consequently, the dimensionality m and σ^2 can be estimated as

$$\hat{m} = 2\,\frac{(\mathbf{1}^T D^{*2}\mathbf{1})^2}{n(n-1)\mathbf{1}^T D^{*4}\mathbf{1} - (\mathbf{1}^T D^{*2}\mathbf{1})^2} \qquad \text{and} \qquad \hat{\sigma}^2 = \frac{1}{2}\frac{\mathbf{1}^T D^{*4}\mathbf{1}}{\mathbf{1}^T D^{*2}\mathbf{1}} - \frac{\mathbf{1}^T D^{*2}\mathbf{1}}{n(n-1)}. \tag{7.3}$$

The goodness of these estimates is illustrated in Fig. 7.8, where the Gaussian samples drawn from $\mathcal{N}(\mathbf{0}, 2\,I)$ in spaces of various dimensionalities are considered. The results are good even for a small sample (with respect to the space dimensionality), such that from the $n \times n$ Euclidean distance matrices (for $n > 10$), an estimation sufficiently close to the true value can be found. Also, when noise is added, the estimation of intrinsic dimensionality does not significantly change, although, the estimated variance is naturally influenced (it becomes larger).

Assume further a Gaussian variable $X \sim \mathcal{N}(\mathbf{0}, \frac{\sigma_i^2}{2} I)$. Then, the square Euclidean distance variable τ is described by $\sum_{i=1}^{m} \sigma_i^2 \kappa_i$, where $\kappa_i \sim \chi_1^2$. Consequently, τ is a linear combination of χ_1^2 distributions with one degree of freedom. Note that if $\sigma_i^2 \approx \sigma_j^2$, then $\kappa_i + \kappa_j \sim 2\,\sigma_i^2 \chi_1^2$. One can, therefore, describe τ approximately as $\sum_{i=1}^{k} \sigma^2 \kappa_i = \sigma^2 \sum_{i=1}^{k} \kappa_i$, where $\sigma^2 = \frac{1}{k}\sum_{i=1}^{m} \sigma_i^2$ and k is equal to m or less depending on the number of dominant variances σ_i^2. So, τ is approximately distributed as $\sigma^2 \chi_k^2$. Effectively, the number of degrees of freedom is determined by the dominant variances[5].

Examples. We will illustrate the difficulty of determining the 'true' intrinsic dimensionality for the square dissimilarity representations. Here, we are only concerned with the data describing a single class. Even though artificial Gaussian samples are considered, they already give some indication of the difficulties to be met in real problems, especially when different dissimilarity measures are used. Three different Gaussian samples are drawn in a 30-dimensional space:

[5] For instance, let $\mathcal{X} \sim \mathcal{N}(\mathbf{0}, \mathrm{diag}\,(\mathbf{v}))$, where $\mathbf{v} = [3\ 3\ 3\ 3\ 3\ 3\ 3\ 1\ 1\ 1]^T$. Then , by the formulation, $\sigma_i^2 = 2\,v_i$. The variable τ can be described as $6 \cdot 7\chi_1^2 + 3 \cdot 2\chi_1^2 = 48\chi_1^2$ and approximated by $5.33\,\chi_9^2$, since $5.33 \approx 48/9$. Effectively, $\hat{\sigma}^2$ should be then 2.67. $\hat{m}$ and $\hat{\sigma}^2$ can be derived for each sample realization of $\mathcal{X}$.

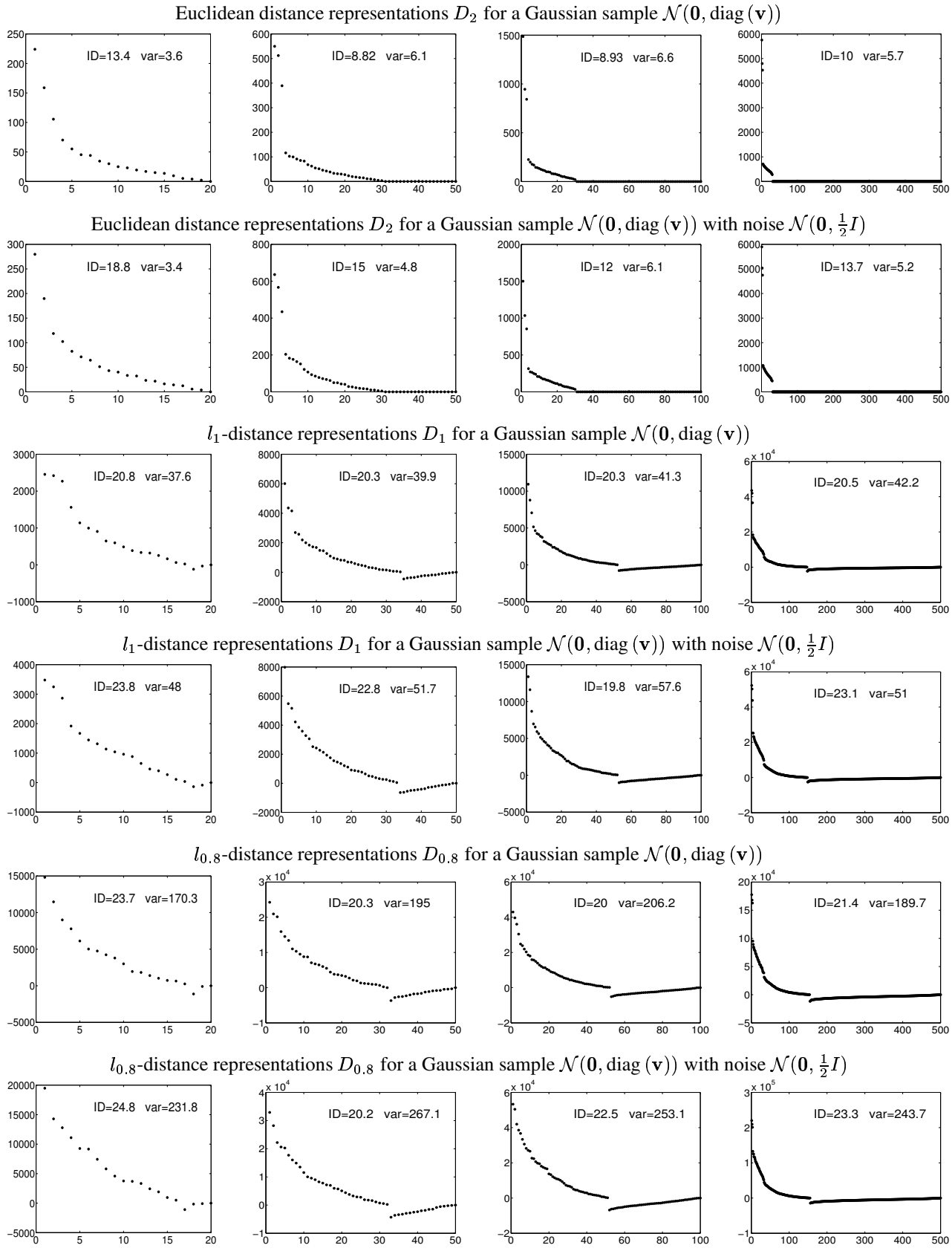


Fig. 7.9: Case 1: Estimation of intrinsic dimensionality for three dissimilarity representations: Euclidean, l_1 and $l_{0.8}$ derived for a Gaussian sample $\mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathbf{v}))$ in a 30-dimensional space, where $\mathbf{v}$ is such that $v_{1,2,3} = 10$ and $v_i = 1$ for $i = 4, \ldots, 30$. Every plot presents the eigenvalues of the pseudo-Euclidean embedding. The cardinality of the Gaussian sample grows from left to right as $20, 50, 100$ and 500.

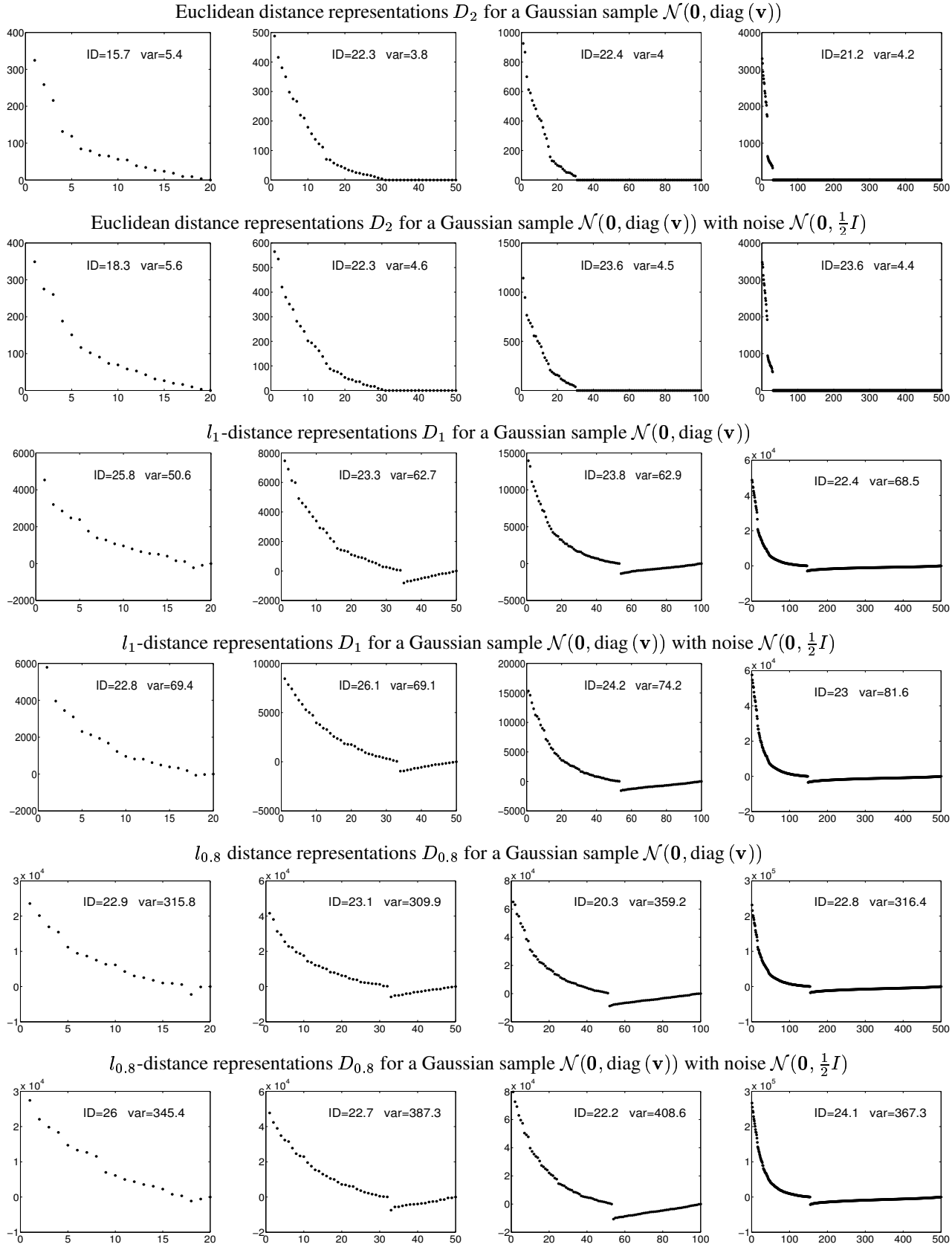


Fig. 7.10: Case 2: Estimation of intrinsic dimensionality for three dissimilarity representations: Euclidean, l_1 and $l_{0.8}$ derived for a Gaussian sample $\mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathbf{v}))$ in a 30-dimensional space, where $\mathbf{v}$ is such that $v_i = 5$ for $i = 1, \ldots, 15$ and $v_i = 1$ for $i = 16, \ldots, 30$. Every plot presents the eigenvalues of the pseudo-Euclidean embedding. The cardinality of the Gaussian sample grows from left to right as $20, 50, 100$ and 500.

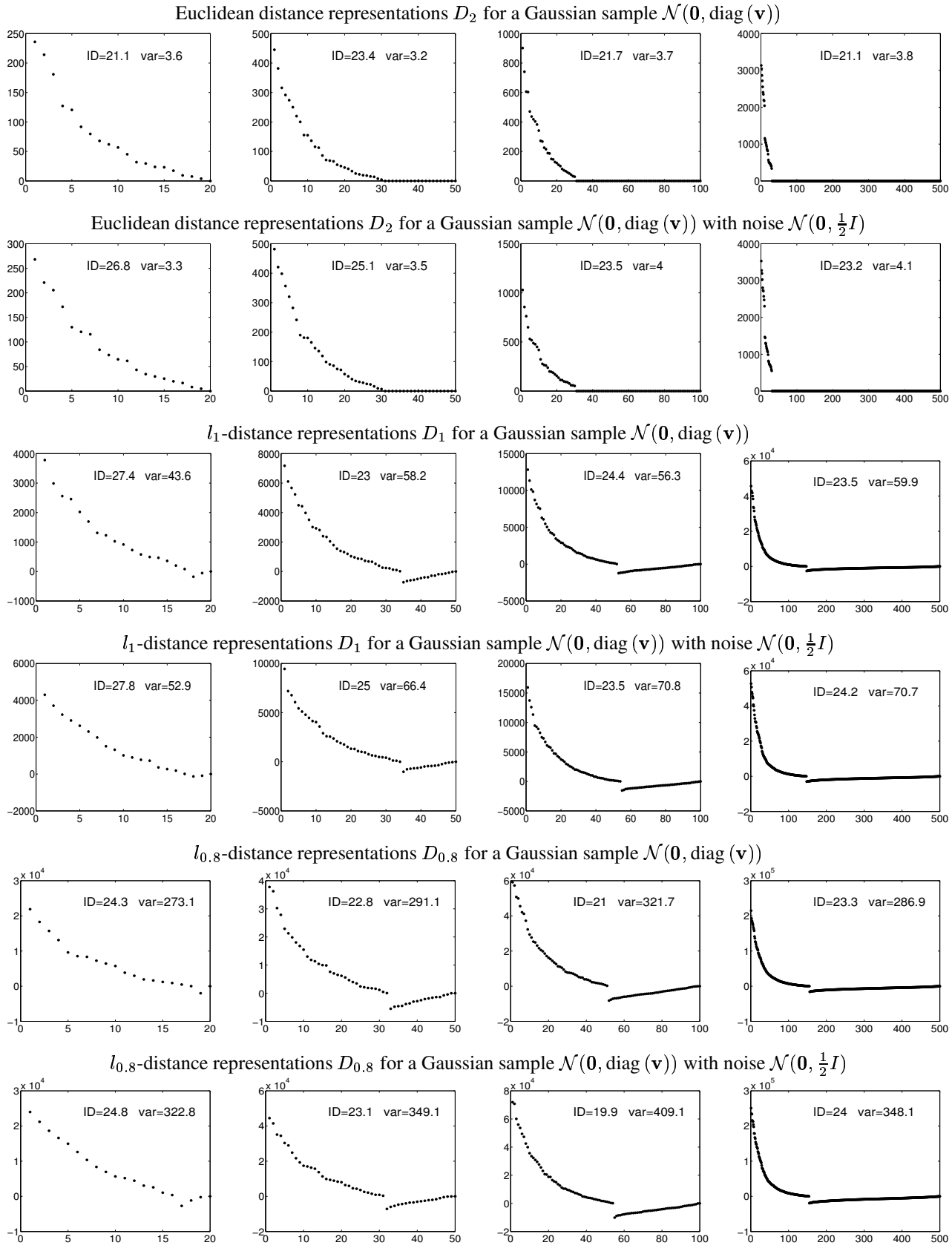


Fig. 7.11: Case 3: Estimation of intrinsic dimensionality for three dissimilarity representations: Euclidean, l_1 and $l_{0.8}$ derived for a Gaussian sample $\mathcal{N}(\mathbf{0}, \mathrm{diag}(\mathbf{v}))$ in a 30-dimensional space, where $\mathbf{v}$ is such that $v_i = 5$ for $i = 1, \ldots, 10$, $v_i = 2$ for $i = 11, \ldots, 20$ and $v_i = 1$ for $i = 21, \ldots, 30$. Every plot presents the eigenvalues of the pseudo-Euclidean embedding. The cardinality of the Gaussian sample grows from left to right as $20, 50, 100$ and 500.

1. **Case 1:** a Gaussian sample $\mathcal{N}(\mathbf{0}, \mathrm{diag}\,(\mathbf{v}))$, where $\mathbf{v}$ is such that $v_{1,2,3} = 10$ and $v_i = 1$ for $i = 4, \ldots, 30$.
2. **Case 2:** a Gaussian sample $\mathcal{N}(\mathbf{0}, \mathrm{diag}\,(\mathbf{v}))$, where $\mathbf{v}$ is such that $v_i = 5$ for $i = 1, \ldots, 15$ and $v_i = 1$ for $i = 16, \ldots, 30$.
3. **Case 3:** a Gaussian sample $\mathcal{N}(\mathbf{0}, \mathrm{diag}\,(\mathbf{v}))$, where $\mathbf{v}$ is such that $v_i = 5$ for $i = 1, \ldots, 10$, $v_i = 2$ for $i = 11, \ldots, 20$, and $v_i = 1$ for $i = 21, \ldots, 30$.

In all cases the 'true' intrinsic dimensionality is 30, since the data is generated by 30 variables. Since the Gaussian samples are not hyper-spherical, the hyper-ellipsoidal data will be treated as such, so they would be indirectly reshaped to a hyper-sphere of a similar volume. The dimensionality estimated from the Euclidean distances by formula (7.3) will be, therefore, smaller than 30. Other dissimilarity measures will also influence the estimation of the intrinsic dimensionality by the amount of 'departure' from the Euclideaness.

One may also be concerned with the effective intrinsic dimensionality, i.e. the number of significant variables, i.e. variables with the largest variance (spread). Such an effective ID can be thought of as $3, 15$ and 10 for the cases 1,2, and 3, respectively. This might be detected by the number of a few the most significant eigenvalues in the pseudo-Euclidean embedding.

Figures 7.9-7.2 show the eigenvalues of the pseudo-Euclidean embeddings for the three cases mentioned above. For each case, three l_p-distance representations are considered for $p = 2, 1, 0.8$, reflecting the proper Euclidean representation, the metric and non-Euclidean city block representation, and the non-metric representation. Also four different sample cardinalities N are taken into account: 20 (undersampled), $50, 10$ (a small sample), and 500 (a large sample). Additionally, the original samples were contaminated with a hypothetical Gaussian noise with the variance of 0.5.

A few general conclusions can be drawn from the analysis of these figures. As expected, the estimated intrinsic dimensionality in all cases is smaller than 30, although the smallest is found for the Euclidean distances in case 1. For non-noisy Gaussian samples in cases 2 and 3, the estimated ID varies between 21 and 24, provided that the sample is larger than 20. Since the added noise is quite large, it disturbs the estimations.

For sufficiently large samples (500 points) and the Euclidean distances, the most informative directions can be revealed, even when noise is added. So, three, 15 and 10 the most significant eigenvalues can be identified, in cases 1–3, respectively. When other distance measures are used (only in case 1), the three eigenvalues can be still clearly detected, while it becomes less obvious in other cases. In general, case 1 seems to be the easiest, such that even for smaller samples, it is possible to distinguish three characteristic eigenvalues. However, it becomes more difficult to judge for cases 2 and 3. Still, if one imagines a curve interpolating the eigenvalues, the change of steepness (hence convexity) suggests the following eigenvalues describe the dimensions of a lesser importance. When determine the number of informative directions, one may, therefore, determine it by the position where the 'eigenvalue curve' changes its steepness. This can be considered as the smallest number of informative dimensions for the problem. The estimated intrinsic dimensionality, formula (7.3), may serve as an upper bound for the determination of the number of significant directions.

Although this is an analysis of a single Gaussian sample, when noise is added, the situation becomes more realistic. When multiple Gaussian samples are considered, the problem becomes much more difficult, since various Gaussian samples might have different numbers of important variables. In such a procedure, all samples are judged as one combined description, so the resultant description might have properties different from any single sample. Yet, the combined description is analyzed from the perspective of significant directions, which in principle allows us to rely on the 'eigenvalue curve' reasoning. However, formula (7.3) cannot be recommended any longer, since the assumption of a single cloud is completely violated. Another type of estimation must be searched for.

7.3 Sampling issues

In this section we will study criteria, which judge from a dissimilarity representation whether a single class is sufficiently sampled. It is partially based on our earlier study [102]. Consider an $n \times n$ dissimilarity matrix $D(R,R)$, where $R := \{p_1, p_2, \ldots, p_n\}$ is a representation set. In general, R may be a subset of a larger training set T. It is assumed here that $R = T$. The entire set R is represented by vectors of dissimilarities $D(p_i, R)$, $i = 1, 2, \ldots, n$. We will address the question whether n, the cardinality of R, is sufficiently large for capturing the variability in the data. Or, in other words, whether it is to be expected that not much can be gained by increasing the number of representation examples. This question is directly related to the complexity of the classification problem as discussed in [104]. To start the analysis, we will restrict ourselves to a more simple issue concerning the sampling of a set of unlabeled objects, possibly forming a single class. Next, the problem will be formulated and some criteria are proposed judging whether the representation set is sufficiently sampled. The usefulness of such criteria is experimentally investigated on two data sets. Although the results are preliminary, they suggest a direction for further study. Some more extensive experiments have been done [104].

The research question refers to the determination of a criterion defined on D judging how well the dissimilarity data are sampled. This can be rephrased as judging whether new objects can be expressed in terms of the ones already present in R or not. Some possible statistics that might be used as such are based on the compactness hypothesis [4, 98, 102]. As it states that similar objects are also close (similar) in their representation, it constrains the dissimilarity measure d in the following way. d has to be such that $d(x,y)$ is small if x and y are very similar, i.e. it should be much smaller for very similar objects than for objects that are very different.

Assume that d is definite, i.e. $d(x,y) = 0$ iff the objects x and y are identical. This implies that they belong to the same class. This can be extended somewhat by assuming that all objects z for which $d(x,z) < \varepsilon$, where $\varepsilon > 0$, are so similar to x (if ε is sufficiently small) that they belong to the same class as x. Consequently, the dissimilarities of x and z to the objects in the representation set R should be close (or in fact positively correlated), i.e. $d(x,p_i) \approx d(z,p_i)$. This implies that their representations $d(x,R)$ and $d(z,R)$ are also close. We conclude that for dissimilarity representations that satisfy the above continuity, a stronger property than formulated by the compactness hypothesis holds, as now also objects that are similar in their representations are also similar in reality. Consequently, they belong to the same class. We will call such representations *true* representations.

A representation set R can be judged to be sufficiently large if an arbitrary new object of the same class is not significantly different from the other objects in the data set. This can be expected if R already contains many objects that are very similar, i.e. if they have a small dissimilarity to at least one other object. All the statistics, studied below are based, in one way or other, on this observation.

We will illustrate the statistics on an artificial example and present later results for some real world data sets. This artificial example will be illustrated on the $l_{0.8}$-distance[6] representation between n normally distributed points in a k-dimensional space. Both, n and k, will be varied between 5 and 500. If $n < k$, then the points lie in an $(n-1)$- dimensional subspace, resulting in an undersampled, difficult problem. If $n \gg k$, then the data set may be judged as sufficiently sampled. Large values of k generate difficult (complex) problems as they demand a large data cardinality n. The results are averaged over 20 experiments, each time based on a new, randomly generated data set. The criteria are presented and discussed below.

In studying the criteria curves, one should remember that the height of the curve or a flattened behavior curve are informative on sampling of a data set. For the PCA, mean relative rank, in-

[6] Remember that the $l_{0.8}$-distance $d_{0.8}(\mathbf{x},\mathbf{y}) = (\sum_{i=1}^{r} |x_i - y_i|^{0.8})^{1/0.8}$ is non-metric.

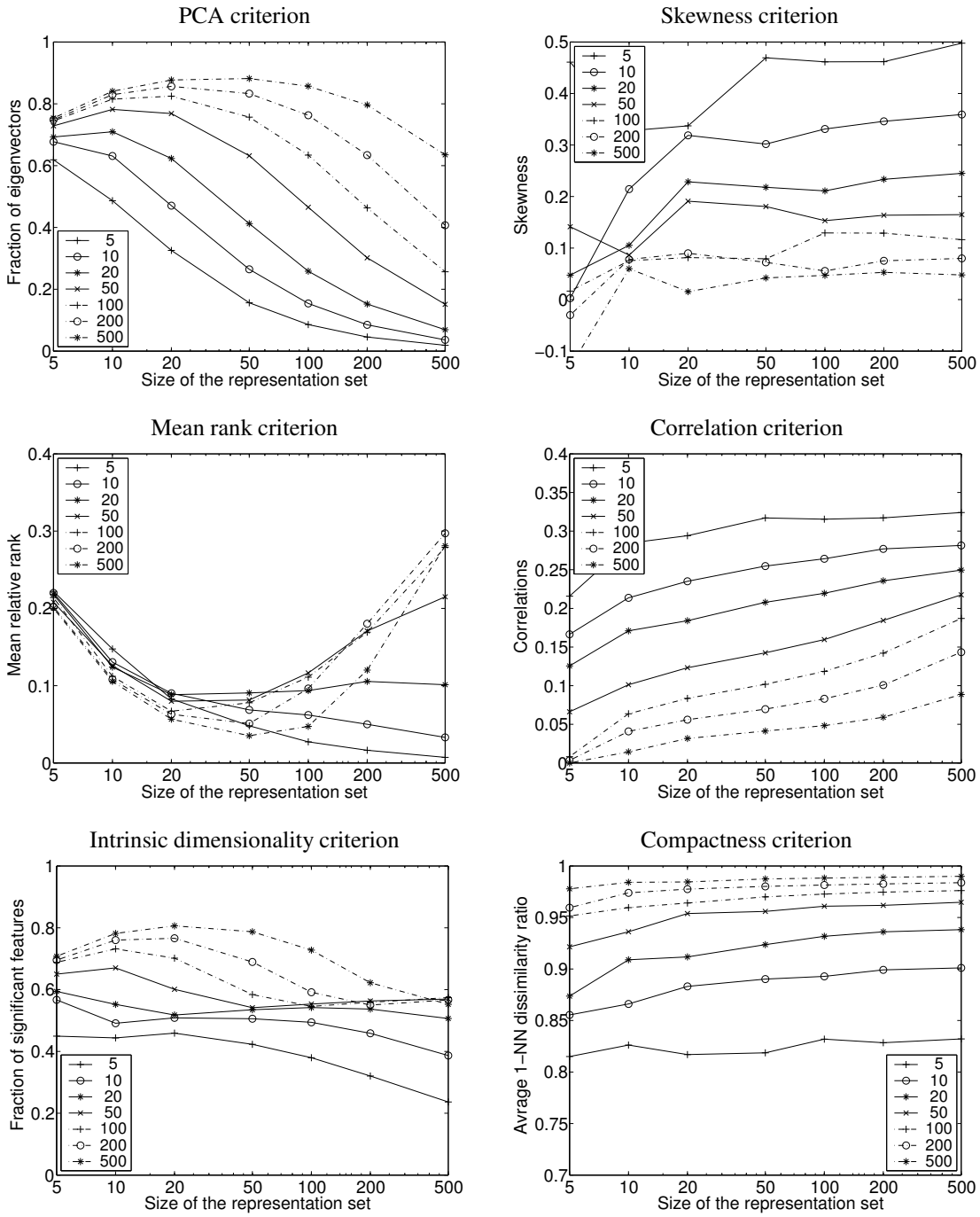


Fig. 7.12: Sampling criteria for the $l_{0.8}$-distance representations $D_{0.8}(R,R)$ computed for artificial Gaussian data sets of a varying dimensionality k (from 5 to 500), as indicated in the legends.

trinsic dimensionality and compactness criteria holds that lower values (of the flattened curves) are indicative either for a sufficient sampling and/or for a compact class description. And the other way around, high flattened values for the curves of skewness and correlation point to a sufficient sampling.

Principal Component Analysis (PCA). A sufficiently large representation set R will contain at least some objects that are very similar to each other, i.e. their representations, the vectors of dissimilarities to all other objects, are very similar. This suggests that the rank of D should be smaller than $|R|$, i.e. $\text{rank}(D) < n$. In practice, this will not be exactly true if objects are not fully alike. A more robust criterion may, therefore, be based on the principal component analysis applied to the dissimilarity matrix D. Basically, the set is sufficiently sampled if n_α, the number of eigenvectors of D for which

the sum of the corresponding eigenvalues equals a fixed fraction α, such as 0.95 of the total sum of eigenvalues (hence α is the explained fraction of the variance) is small in comparison to n. So, for well represented sets, the ratio of n_α/n is expected to be smaller than some small constant(the faster the criterion drops with a growing R, the smaller intrinsic dimensionality of the dissimilarity space representation). So, our criterion is:

$$J_{\text{pca},\alpha} = \frac{n_\alpha}{n}, \tag{7.4}$$

with n_α such that $\alpha = \sum_{i=1}^{n_\alpha} \lambda_i / \sum_{i=1}^{n} \lambda_i$. There is usually no integer n_α for which the above holds exactly, so it would be found by interpolation. In the experiments, in Fig. 7.12, top left, the value of $J_{\text{pca},0.95}$ is shown for the artificial Gaussian example as a function of the cardinality of R. The Gaussian data are studied for various dimensionalities k. It can be concluded that the data sets consisting of more than 100 objects may be sufficiently well sampled for small dimensionalities such as $k = 5$ or 10. On the other hand, the considered number of objects is too small for the Gaussian sets of a larger dimensionality. These generate problems of a too high complexity for the given data size.

Skewness. A new object added to a set of objects that is still not sufficiently well sampled will generate many large dissimilarities and just a few small ones. As a result, for insufficiently sampled data, the distribution of dissimilarities will peak for small values and show a long tail in the direction of large dissimilarities. By adding new objects more and more small dissimilarities will appear. Consequently, the skewness grows with increasing $|R|$. The value to which the skewness grows, however, depends on the problem. After the set becomes 'saturated', however, the skewness curve should flatten. Note also that a negative skewness will indicate a 'tail' of small dissimilarities or a distribution with more than one mode (possible clusters). The skewness of the distribution of the dissimilarities d is given as

$$J_{sk} = E\left[\frac{d - E[d]}{\sqrt{E[d - E[d]]^2}}\right]^3, \tag{7.5}$$

where $E[\cdot]$ denotes the expectation. In practice, the off-diagonal values d_{ij} of the dissimilarity matrix D are used for the estimation. In Fig. 7.12, top right, the skewness of the Gaussian sets are presented. For small representation sets, their cardinalities appear to be insufficient for representing the problem, as it can be concluded from the noisy behavior of the graphs in that area. For large representation sets, the curves corresponding to the Gaussian samples of different dimensionalities 'asymptotically' increase to different values of J_{sk}. These final values may be reached earlier for more simple problems in low dimensions, like $k = 5$ or 10. This is, however, not clearly observable.

Mean relative rank. An element d_{ij} represents the dissimilarity between the objects p_i and p_j. The minimum of d_{ij} over all indices j points to the nearest neighbor of p_i, say, p_z if $z = \text{argmin}_{j \neq i}(d_{ij})$. So, in the representation set R, p_z is the most similar to p_i. We now state that a representation $D(p_i, R)$ of p_i describes the object well if the representation of p_z, i.e. $D(p_z, R)$ is close to $D(p_i, R)$ in the dissimilarity space. This can be measured by ordering the neighbors of the vectors $D(p_i, R)$ and determining the rank number r_i^{NN} of $D(p_z, R)$ in the list of neighbors of $D(p_i, R)$. So we compare the nearest neighbor as found in the original dissimilarities with the nearest neighbors in the dissimilarity space. For a well-described representation we expect that the mean relative rank:

$$J_{mrr} = \frac{1}{n}\sum_{i=1}^{n} r_i^{NN} - 1 \tag{7.6}$$

is close to 0. In Fig. 7.12, middle left, the results for the Gaussian example are shown. Similarly, as for the PCA criterion, it can be concluded that the sizes of the representation set R larger than 100 are sufficient for Gaussian samples in 5 or in 10 dimensions.

Correlation. The correlations between the objects in the dissimilarity space will also be used. Similar objects show similar dissimilarities to other objects and are, thereby, positively correlated. As a consequence, the average of positive correlations $\rho_+(D(p_i,R),D(p_j,R))$ divided by the average of absolute values of negative correlations $\rho_-(D(p_i,R),D(p_j,R))$:

$$J_\rho = \frac{\frac{1}{n^2-n}\sum_{i,j\neq i}^{n}\rho_+(D(p_i,R),D(p_j,R))}{1+\frac{1}{n^2-n}\sum_{i,j\neq i}^{n}|\rho_-(D(p_i,R),D(p_j,R))|} \tag{7.7}$$

will increase for large sample sizes. The constant added in the denominator prevents J_ρ from becoming very large if only small negative correlations appear. For a well-sampled representation set, J_ρ will be large and it will increase only slightly when new objects are added (new objects should not significantly influence the averages of either positive or negative correlations). Fig. 7.12, middle right, shows that this criterion works well for the artificial Gaussian example. For less complex problems J_ρ reaches higher values and exhibits a flatten behavior for sets consisting of at least 100 objects.

Intrinsic embedded dimensionality. Another possibility to judge whether R is sufficiently sampled is to estimate the intrinsic dimensionality of the underlying vector space, where the original dissimilarities are preserved. This can be achieved by a linear embedding (provided that D is symmetric) into a pseudo-Euclidean space; see section 3.3 for details. The representation X, consisting of $m \leq n$ dimensions, is determined such that it has uncorrelated derived features and it is centered at the origin. The dominant variances captured by the eigenvalues determined in the embedding should reveal the intrinsic dimensionality (small variances are expected to show just noise). Since the dimensions corresponding to small variances can be neglected. (Note, however, that when all variances are similar, the intrinsic dimensionality is approximately n.) Let n_α^{emb} be the number of dimensions with significant variances for which the sum of the corresponding magnitudes of variances equals a specified fraction α such as 0.95 of the total sum. Of course, n_α may not be found exactly, so it is interpolated. Since n_α determines the intrinsic dimensionality, so as a criterion we propose the following index:

$$J_{dim,\alpha} = \frac{n_\alpha^{emb}}{n}. \tag{7.8}$$

For low intrinsic dimensionalities, smaller representation sets are needed to describe the data characteristics. Fig. 7.12, bottom left, presents the behavior of our criterion as a function of $|R|$ for the Gaussian data sets. The criterion curves clearly reveal different intrinsic dimensionalities. If R is sufficiently large, then the intrinsic dimensionality remains constant. Since the number of objects is growing, the criterion should then decrease and reach a relatively constant small value in the end (for very large sets). From our plot, we can then conclude that the data sets of more than 100 objects are satisfactorily sampled for original Gaussian data of a low dimensionality, i.e. $k \leq 20$. In other cases, the data are too complex.

Compactness. As mentioned above, a symmetric distance matrix D can be embedded in a pseudo-Euclidean space $\mathcal{E}$. When the representation set is sufficiently large, the intrinsic dimensionality is expected to remain constant during further enlargement. Consequently, the mean of the data should remain approximately the same and the average distance to this mean should decrease (as new objects don't surprise anymore) or be constant. The larger the average distance, the less compact the class is, requiring more samples for its description. Therefore, a compactness criterion can be investigated. It is estimated in the leave-one-out approach as the average square distance to the mean vector in an embedded space:

$$J_{loo} = \frac{1}{n^2-n}\sum_{j=1}^{n}\sum_{i\neq j} d_{\mathcal{E}}^2(\mathbf{x}_i^{-j},\mathbf{m}^j), \tag{7.9}$$

where $\mathbf{x}_i^{-j}$ is a vector representation of the i-th object in the pseudo-Euclidean space determined by all the objects, except object j, and $\mathbf{m}^j$ is the mean of such a configuration. Fig. 7.12, bottom right, shows the behavior of this criterion, clearly indicating a high compactness of the low-dimensional Gaussian data. The case of $k = 500$ is judged as not having a very compact description.

Experiments with the NIST digits. A training set of four classes of handwritten digits shapes of $0, 1, 2$ and 3 from the NIST database [420] constitutes a representation set R. For each class, $n = 200$ objects are considered. The modified Hausdorff distance D_{MH}, Def. 5.6, is computed between the digit contours derived from binary images. Three variants of the distance representations based on the element-wise (Hadamard) power transformation: D_{MH}^{*5}, D_{MH} and $D_{\text{MH}}^{*0.2}$ are studied. These transformations do not change the rank of the dissimilarities, but they influence both dissimilarity and embedded spaces and, thereby, the criterion values in a non-linear way.

Fig. 7.13 - 7.15 present the results of the six criteria introduced in the previous section as a function of a growing representation set. The experiments are repeated 20 times for randomly chosen subsets. The following observations can be made:

1. The four character sets of '0'-'3' show slightly different behavior. In general, the set of '1'-s is the simplest (the most compact one) and the set of '3'-s is the most difficult one.
2. The power transformation influences the criteria significantly. The power of 0.2 has some normalizing effect as it removes the tails of the distribution of dissimilarities.
3. The PCA (7.4) indicates that for the $D_{\text{MH}}^{*0.2}$ representation, the cardinality of R is far from being sufficient. It even shows some yet unexplained peaking phenomenon. For D_{MH} the set of '1'-s is well sampled and for D_{MH}^{*5}, all four character sets are sufficiently large; see Fig. 7.13, left.
4. The skewness criterion (7.5) is noisy and not very informative; see Fig. 7.13, right.
5. The mean relative rank (7.6) shows, Fig. 7.14, left, that the $D_{\text{MH}}^{*0.2}$ set builds a good representation space in which distances correspond well to the original dissimilarities. This can be explained by the linearizing effect of taking a small power $\ll 1$. At the same time, the difference in complexity between the character sets has disappeared. The strongly nonlinear D_{MH}^{*5} dissimilarity representation appears to be difficult according to the mean relative rank (7.6); see Fig. 7.14, left.
6. The correlation criterion (7.7) shows interesting results; see Fig. 7.14. It indicates that with respect to other classes, the set of '1'-s is the best sampled. It is sufficiently sampled for the original representation D_{MH}. For $D_{\text{MH}}^{*0.2}$, the curves are growing fast, which points to a possible large increase of J_ρ, hence an insufficient sampling. On the other hand, the correlation criterion does not seem much increasing for D_{MH}^{*5} with growing R.
7. The embedded intrinsic dimensionality (7.8) of the set of '1'-s is relatively low and much smaller than for other data sets; see Fig. 7.15. The largest intrinsic dimensionality has the class of '0'-s. Remarkably, the dimensionality of the combined set seems to be relatively low, indicating that all classes share some descriptions.
8. The set of '0'-s is the most compact class according to the criterion (7.9); see Fig. 7.15. The sets of '1'-s and '2'-s are much less compact, indicating possible subclasses or elongated distributions. Not surprisingly, this criterion judges the combined set of all characters as more complicated than any single of them.
9. Most data sets may be judged as well sampled in D^{*5} transformation.

A global comparison of Fig. 7.12 to Fig. 7.13 - 7.15 shows that the characteristics of high dimensional Gaussian distributions cannot be found in a real world problem. This observation is confirmed in our recent study (on a number of problems) devoted to complexity and sampling issues [104].

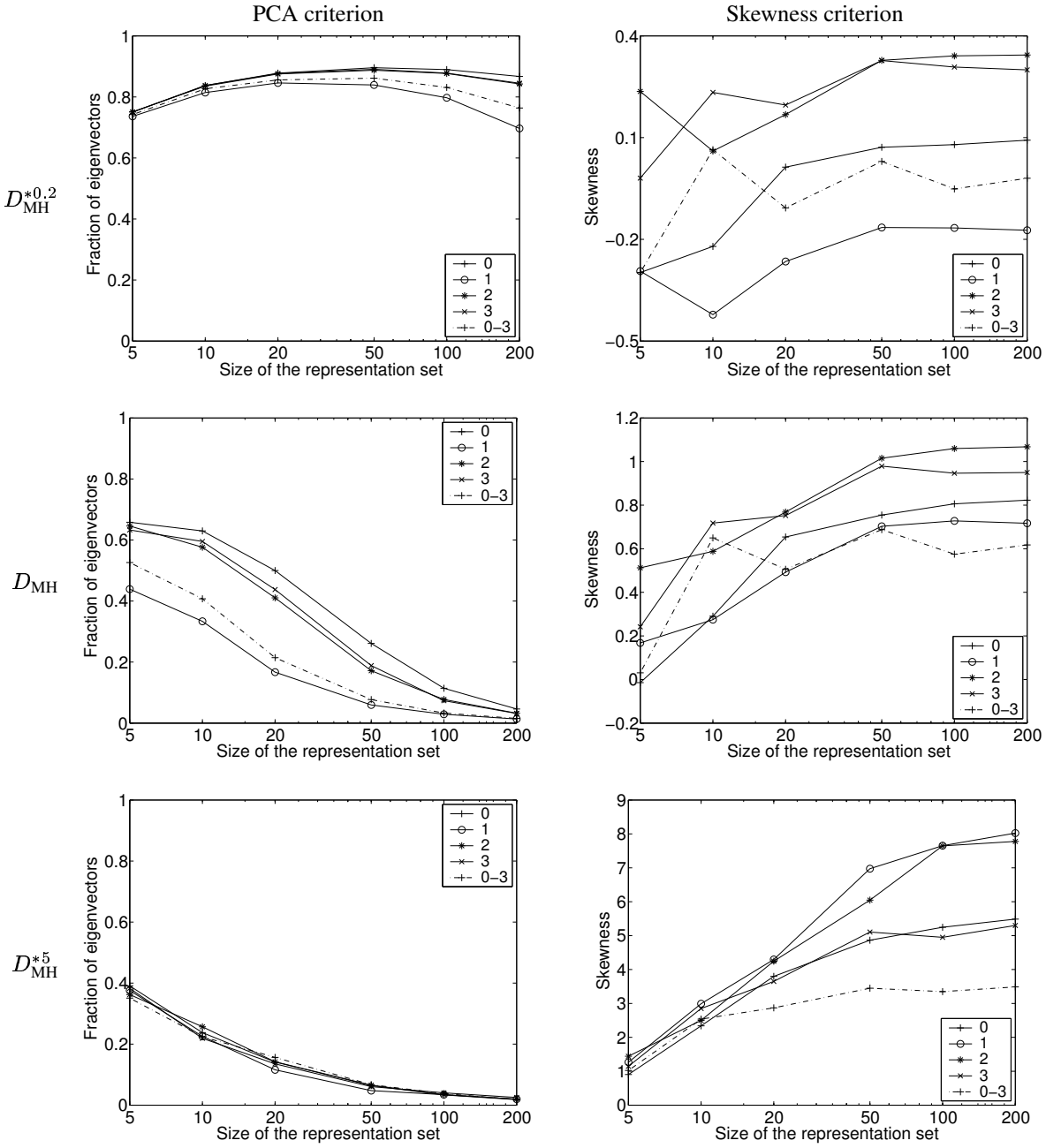


Fig. 7.13: PCA (7.4) and skewness (7.5) criteria as functions of $|R|$ for four sets of the NIST handwritten digits represented by the modified-Hausdorff distances D_{MH}. Three different power transformations are used: $D_{\mathrm{MH}}^{*0.2}$, D_{MH} and D_{MH}^{*5}.

Concerning the criteria used, the following may be concluded. The PCA criterion works well for the $l_{0.8}$ distance representations of artificial data and for the modified Hausdorff distance example on real data. In the latter case, this criterion may seem to indicate some unwanted property of growing criterion curves for $D_{MH}^{*0.2}$. This is however in agreement with the complexity of the class as described by the power transformation. The skewness is noise sensitive. Still, we expect that the distribution of the dissimilarity values should be indicative for the complexity of the problem in one way or another. Indeed, skewness can provide some more insight as studied also in [104]. The nearest neighbor relationships on which the mean relative rank criterion is built appear to be useful in both, the artificial problem, as well as for the real data. Both the estimation of intrinsic embedded dimensionality and compactness of the data description are found to be informative, when treated as complementary information. They give an indication of the problem complexity. The correlation

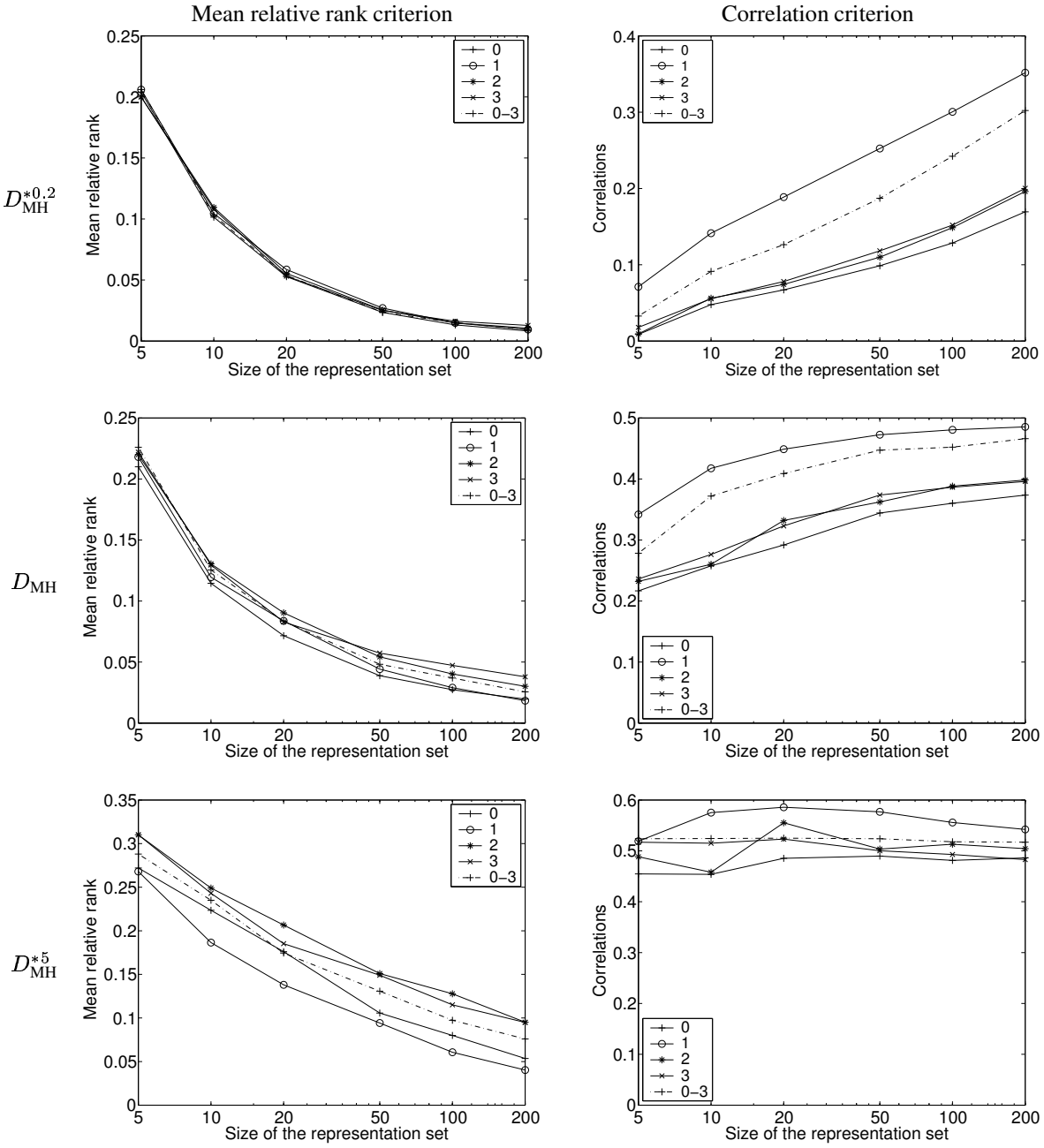


Fig. 7.14: Mean relative rank (7.6) and correlation (7.7) criteria as functions of $|R|$ for four sets of the NIST handwritten digits represented by the modified-Hausdorff distances D_{MH}. Three different power transformations are used: $D_{\mathrm{MH}}^{*0.2}$, D_{MH} and D_{MH}^{*5}.

criterion performed also well. The criteria should be further tested on artificial data sets and in real applications. Other criteria using label information may be considered as well in relation with classification problems. Some further study has also been conducted[104].

7.4 Summary

This chapter pertains to techniques that enable exploration of the dissimilarity data. Many clustering algorithms have been proposed in the neighborhood-based framework (often called proximity-based clustering). Our contribution here is to propose the use of both the embedded and dissimilarity spaces. As such the use of a dissimilarity space for clustering is new. One of our interesting conclusions is that the dissimilarity space approach might be especially useful for problems where

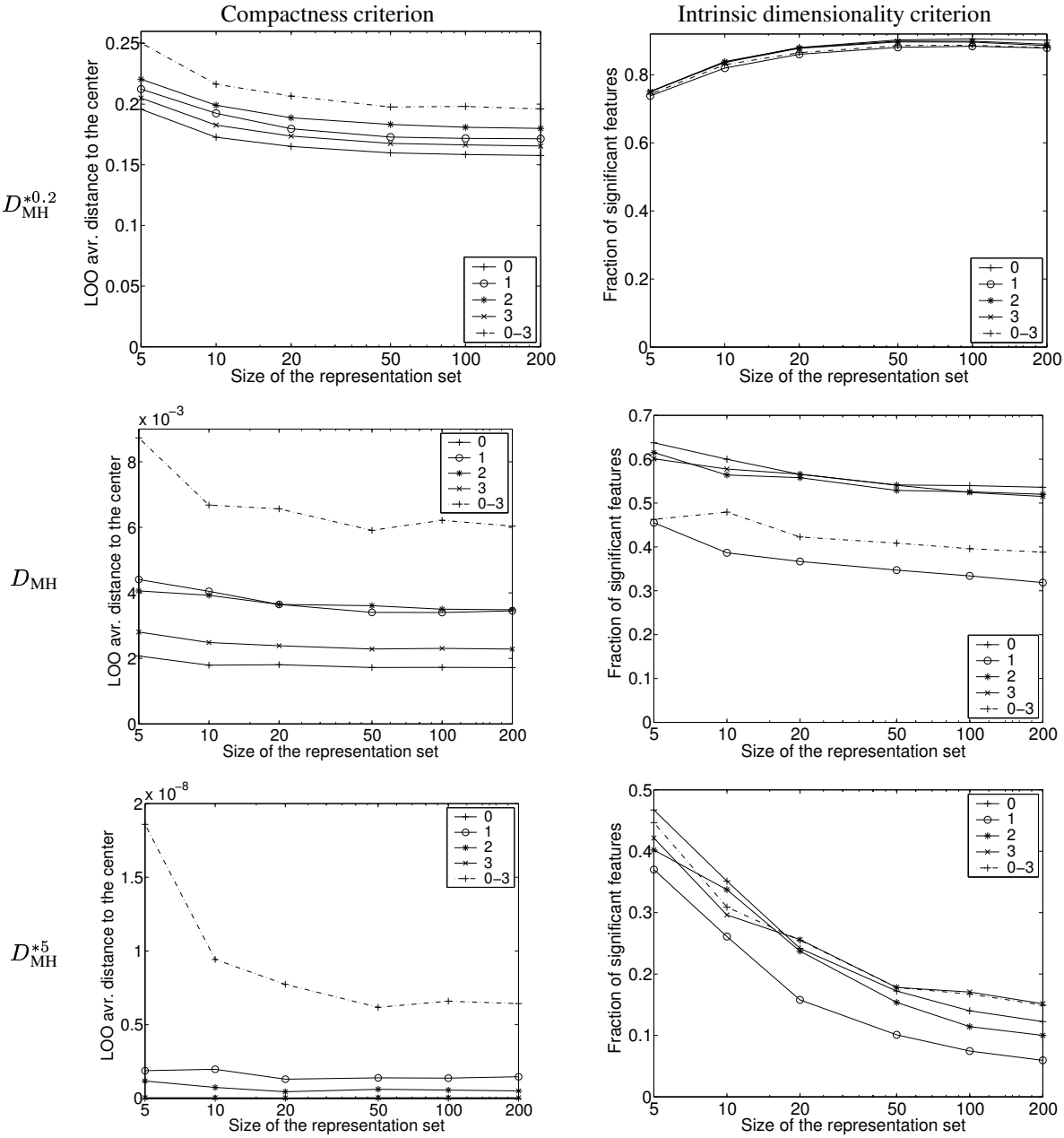


Fig. 7.15: Compactness rank (7.9) and intrinsic dimensionality (7.8) criteria as functions of $|R|$ for four sets of the NIST handwritten digits represented by the modified-Hausdorff distances D_{MH}. Three different power transformations are used: $D_{\mathrm{MH}}^{*0.2}$, D_{MH} and D_{MH}^{*5}.

at least one of the sought groups is compact and some others are widely-spread. In general, the NQC-clustering (mixture-of-Gaussian EM-clustering) seems to work well in the PCA-dissimilarity space. Both the embedding and dissimilarity space grouping techniques give promising results.

Both the embedded and dissimilarity spaces may serve for the estimation of the intrinsic dimensionality of the data. In general, the former methods are based on detecting the satisfactory dimensionality of an embedding, while the latter methods rely on various reduction based techniques. The use of a simple linear pseudo-Euclidean embedding, as well as the principal component analysis in a dissimilarity space gives reasonable indications.

Some statistics have been considered that can be used for examining whether a representation set contains a sufficient number of objects to describe a class. The problem itself is ill-defined as it depends on a specific application as to what 'sufficient' means for a single class. One might imag-

ine that classes are well sampled, but positioned with respect to each other in such a complicated way that the classification problem is difficult for most classifiers. As a consequence, the size of the training set should be judged from an evaluation of the classification result using a test set. In the presented study, an attempt is made to find out whether it is possible to judge from a dissimilarity matrix its sampling density. Some criteria are proposed, which overall, work well. The most indicative are the ones based on the number of the most significant eigenvalues either in the the PCA-dissimilarity space or the pseudo-Euclidean embedding and the mean relative rank criterion.

8. One-class classifiers

The whole is more than the sum of its parts.
"METAPHYSICA 10F-1045A", ARISTOTLE

The problem of describing a single class or a domain has recently gained a lot of attention, since it is identified in many real applications. The area of interest covers all situations, in which the specified targets have to be recognized and non-targets, anomalies or outlier situations have to be detected[1]. These might be examples of any type of fault detection (wearing of a machine) or target detection (e.g. face detection in images), abnormal behavior (intruders attacks on networks, a suspicious behavior in surveillance checks), disease detection, person identification, etc. The methodology for handling such situations can be also useful for imbalanced data, in which one class is represented by a relatively small number of examples, usually due to either an occasional occurrence of such examples or high costs connected to the measurement process. In brief, the problem is characterized by the presence of a target class, which should be well sampled. The goal is to describe this class such that resembling objects are accepted as targets and outliers are rejected. The outliers are, however, badly represented, with unknown priors, or even not provided at all. If available in a training stage, they may have a different distribution than in a testing stage, as may occur in a time-changing process, e.g. wearing-of a machine.

Different methods are developed for that purpose, among others the so-called *one-class classifiers* (OCCs), which are domain or boundary descriptors; see [386, 390]. The OCCs describe the data characteristics such that a proximity function of an object to a target class is defined. In principle, OCCs are concept descriptors. The description of this class should be wide enough to accept most of the new-coming targets, yet sufficiently tight to reject the majority of outliers. The construction of OCCs is, however, an ill-posed problem since the knowledge on a class is deduced from a finite set of target examples, while the outliers are infrequently sampled or not at all.

The basic assumption that an object belongs to a class is that it is similar to other examples within this class. This is expressed by a proximity judgment or a typicality measure. In feature spaces, such a typicality depends on (non-)linear combinations of features. An OCC built in a feature space can, for instance, be found by determining the minimum-volume hypersphere containing (almost) all target points [390, 391] or by finding a hyperplane optimally separating the target points from the origin as well as possible [349, 350]. Then, the proximity function assigns an object to the target class depending whether its feature vector lies inside a hypersphere or on a proper side of a hyperplane. By the use of (conditionally) positive kernels, Def. def:pdfunction, a kernel-based OCC can be designed to offer nonlinear descriptions; see [347, 386]. Alternatively, when objects are represented by dissimilarities, the proximity will become a function of the given dissimilarities. The strength of dissimilarity representations lies in the focus on differences between objects, which may lead to an easier detection of outliers.

In this chapter, some class descriptors built on dissimilarities are proposed. These are constructed based on neighborhoods, in pseudo-Euclidean spaces or in dissimilarity spaces. The results presented here originate from our publications [301, 305, 306].

[1] Outliers and non-targets can be understood differently. Non-targets denote examples of 'opposite' characteristics that the targets posses (e.g. 'ill' versus 'healthy'), while outliers denote examples somewhat different than the targets are (e.g. an advanced versus a mild stage of a disease). For simplicity, we will further refer to outliers only.

8.1 General issues

One-class classifiers are trained to accept the target class and reject the outliers. Such OCCs either fit some density model to the data or directly describe the boundary. In the most simple case, the assignment of a sample to the class depends on its closeness (proximity) to a sort of an average representative. This may not correspond to any actual object, but it will contain a summary of either all or some essential class members. In other case, an instance is considered to be a member of the class if it is somehow judged as jointly close to a set of selected objects, e.g. boundary examples. In all such approaches, an identification procedure is realized by a proximity function $f_{\text{proxm}}(x, \omega_T)$ of an object x to the target class ω_T equipped with a threshold γ, which determines whether an instance belongs to the class or not. Usually, an OCC is expressed as $\mathcal{C}(x) := \mathcal{I}(f_{\text{proxm}}(x, \omega_T) < \gamma)$, where $\mathcal{I}$ is the identification function (i.e. taking value of 1 if the condition is true, and 0, otherwise).

The threshold γ can be determined by using the information on the training objects. A standard way is to supply a fraction r_{tp} (a true positive ratio) of the target objects to be accepted by an OCC [387, 389]. Equivalently, a fraction r_{fn} (a false negative ratio) of the target objects rejected by the OCC can be used. This means that the threshold γ is set up such that $\int \mathcal{I}(f_{\text{proxm}}(x, \omega_T) > \gamma)\, df_{\text{proxm}}(x, \omega_T)(x) = r_{\text{fn}}$, where $\mathcal{I}$ is the indicator function. r_{fn} is set up as a small value, for instance $r_{\text{fn}} = 0.05$, to prevent a high acceptance of outliers as targets (false positive). γ can also be determined as $(1 - r_{\text{thr}})$-percentile of the sorted sequence of the proximity outputs computed for the training (target) examples. r_{thr} is then a user-specified fraction. *Note, however, that γ specified in this way in not related to r_{fn}.* This is important, since in our study, due to the implementation of the one-class classifiers [387], some of them estimate the threshold directly based on r_{fn} and some others based on r_{thr}.

To study the behavior of an OCC, one often uses a ROC (Receiver Operator Characteristics) curve [41, 389], which is a function of the true positive ratio (target acceptance) versus the false positive ratio (outlier acceptance). In order to evaluate that, example outliers are necessary. In principle, an OCC is trained with a fixed target rejection ratio r_{fn} (or the threshold fraction r_{thr}) for which the threshold is determined. This OCC is then optimized for one point on the ROC curve. In order to compare the performance of various classifiers, the AUC measure can be used [40]. It computes the Area Under the Curve (AUC), which is the total OCC's performance integrated over all thresholds. The AUC of 0.5 or less indicates that the OCC is worse than random guessing. The larger AUC, the better the OCC is; for instance in Fig. 8.1, the solid curve indicates a better performance since the AUC becomes larger than for the dashed curve. The black dots indicate points for which the thresholds γ of two OCCs were optimized.

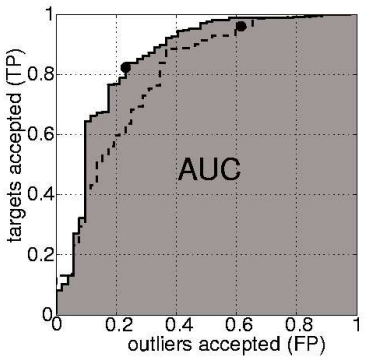


Fig. 8.1: An ROC curve.

8.2 Domain descriptors for dissimilarity representations

Although a dissimilarity measure d provides a flexible way to represent the data, there are some constraints on the measure d itself. Reflectivity and positivity conditions are essential to define a proper measure, however, in conceptual representations 4.2 one can also use negative dissimilarities. Although for our convenience, the symmetry requirement is adopted (it is required for an embedding of the dissimilarity data into a pseudo-Euclidean space), it is not necessary for constructing OCCs in pretopological and dissimilarity spaces. We do not require that d is a metric. Remember that a dissimilarity representation $D(T, R)$ based on the representation set $R = \{p_1, .., p_n\}$ and the training set T can now be interpreted in three ways. In the *pretopological* approach, OCCs will be based

on dissimilarities to neighboring objects. In the *embedding* approach, where necessarily $R \subseteq T$ and the symmetry condition holds, OCCs will be built as in the underlying pseudo-Euclidean space. In the *dissimilarity space* approach, OCCs will be constructed in an n-dimensional dissimilarity space $D(\cdot, R)$. Unless stated otherwise, both R and T consist of the target objects only.

Concave transformations of dissimilarities. Transformations of dissimilarities play a two-fold role. If a measure is unbounded, then some atypical objects of the target class (i.e. with large dissimilarities) may badly influence the solution of an OCC. Therefore, a transformation to a bounded interval might be useful. For instance, a transformation to $[0, 1]$ can be used, such that locally the dissimilarities are scaled linearly and globally, all large dissimilarities become close to 1. Another issue is to impose an extra flexibility of a description by a nonlinear transformation equipped with a parameter to be tuned. The purpose of such a transformation is e.g. to enhance the compactness (expressed by small dissimilarities) of local neighborhoods by setting a proper parameter value. To determine a suitable value is not trivial; usually an additional validation set, possibly containing some outlier examples, should be used. A study on the selection of a proper parameter is left for further research. Some ideas have been recently presented in [388].

Transformations that we have in mind are non-decreasing functions since they preserve the order of original dissimilarities and are concave. The concavity ensures that metric properties are preserved; see Theorem 3.15. Examples of such transformations are the following functions[2] (defined on $\mathbb{R}^0_+$): linear, $f(x) = a\,x$, power, $f(x) = x^p$, where $p \in [0, 1]$, logarithmic, $f(x) = \log(1 + ax)$, or sigmoid $f(x) = 2/(1 + e^{-x/s}) - 1$ or $f(x) = 2/(1 + e^{-x^2/s^2}) - 1$, where s controls the 'slope' of f (the size of the local neighborhoods). Such transformations are applied in an element-wise way to dissimilarity representations such their transformed versions are obtained $D^f(x, R) = f(D(x, R))$.

Below, we will introduce some OCCs constructed in three different frameworks of interpreting the dissimilarities, as discussed above. Their behavior will be illustrated on an artificial example of a theoretical banana target class described by a Euclidean distance representation $D(R, R)$ and its sigmoidal transformations. We will use the following notation: $D^s(x, R) = 2/(1 + e^{-D(x,R)/s}) - 1$ (sigmoidal-I) and $D^{s2}(x, R) = 2/(1 + e^{-D(x,R)^2/s^2}) - 1$ (sigmoidal-II).

8.2.1 Neighborhood-based OCCs

A domain descriptor can be built using the neighborhood relations to some representation objects. Such objects are chosen as the ones which have relatively many close (as judged by dissimilarities) neighbors. For a dissimilarity representation $D(T, R)$, this can be achieved by the k-centers algorithm [426], originally a clustering method. It looks for k center objects, i.e. examples $p_{(1)}, \ldots, p_{(k)}$ that minimize the maximum of the dissimilarities over all the objects to their nearest neighbors. In a forward search strategy, starting from a random initialization, the error $E = \max_{i=1,\ldots,N} \min_{z=1,\ldots,k} D(t_i, p_{(z)})$ is minimized. With M trials, e.g. $M = 50$, the objects corresponding to the minimal value of E are determined; see also section 7.1.2.

Assume that $R_{cent} = \{p_{(1)}, \ldots, p_{(k)}\}$. For N target objects t_i, dissimilarities to their nearest center (among the k chosen centers), $d_{cent}(t_i, R_{cent}) = \min_{z=1,\ldots,k} d(t_i, p_{(z)})$ are computed. The threshold γ is chosen as the $(1-r_{\text{thr}})$-th percentile of the sorted sequence of d_{cent}, where r_{thr} is a chosen fraction, e.g. $r_{\text{thr}} = 0.1$. Also, a suitable γ can be sought such that a specified false negative ratio, e.g. $r_{\text{fn}} = 0.05$, is reached. The k-centers data description [386, 387], $\mathcal{C}_{\text{k-CDD}}$, relies on the dissimilarities to k objects only. It becomes then:

$$\mathcal{C}_{\text{k-CDD}}(D(x, R_{cent})) = \mathcal{I}(\min_z d(x, p_{(z)}) \leq \gamma). \tag{8.1}$$

[2] It is straightforward to check their monotonicity and concavity. The latter is guaranteed by non-positive second derivative [125].

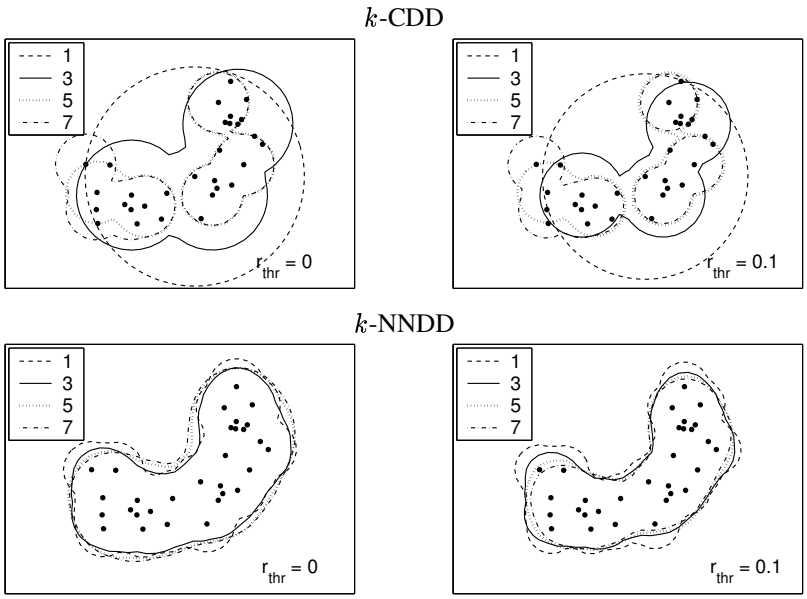


Fig. 8.2: Neighborhood-based OCCs for a Euclidean distance representation D of a theoretical banana class. The plots on the left and right sides show the OCCs with the thresholds $r_{\text{thr}} = 0$ and $r_{\text{thr}} = 0.1$, respectively. Remember that r_{thr} is a threshold on the derived conceptual proximity values and *not* the false negative ratio. That is why for $r_{\text{thr}} = 0.1$, one cannot expect 10% of points to be outside the class boundary. The legends refer to various choices of k.

Another OCC, the k-nearest neighbor data description (k-NNDD) is realized by $\mathcal{C}_{\text{k-NNDD}}$, where the proximity function relies on the nearest neighbor dissimilarities [386, 387]. For N target training objects t_i, the averaged k nearest neighbor dissimilarities $d_{nn}(t_i, R) = \frac{1}{k}\sum_{j=1}^{k} d(t_i, p_i^j)$, where $p_i^j \in R$ is the j-th nearest neighbor of t_i, are computed. Then, a threshold γ is determined as the $(1-r_{\text{thr}})$-th percentile of the sorted sequence of d_{nn}. Similarly, as above, γ can be first found to ensure the that the fraction of rejected target examples is r_{fn}. The classifier becomes then:

$$\mathcal{C}_{\text{k-NNDD}}(D(x,R)) = \mathcal{I}(d_{nn}(x,R) \leq \gamma) = \mathcal{I}(\frac{1}{k}\sum_{j=1}^{k} d(x, p_x^j)) \leq \gamma), \tag{8.2}$$

where $p_x^j \in R$ is the j-th nearest neighbor of x. So, $\mathcal{C}_{\text{k-NNDD}}$ relies on the dissimilarities to all objects from R. Note that for such OCCs, non-decreasing and concave transformations preserve the order of dissimilarities, hence they will hardly change the OCCs built on the original dissimilarities.

In both cases, the proximity function can also be used to define not one, but many thresholds, e.g. thresholds either for each center (the k-CDD) or for each object (the k-NNDD). This, however, requires a lot of data for a good estimation. As such, the OCCs are built using the target information only. It is not clear to us how potential outliers can be used to define the boundary. In general, if $|R| < |T|$, then such OCCs will be denoted as the reduced versions, i.e. the k-nearest neighbor and k-centers reduced data descriptions, the k-NNRDD and k-CRDD, respectively.

As an example, the OCCs are trained on a Euclidean distance representation D of the banana class. Since the theoretical data are 2-dimensional the boundaries of the OCCs can be drawn for such a case. The results are presented in Fig. 8.2. The 'bubble'-like character of the k-CDD is caused by Euclidean balls (containing neighboring objects) around the centers, hence different values of k influence the boundary a lot. On the other hand, the boundary of the k-NNDD relies on averaged distances to each object of R, hence it becomes smoother.

8.2.2 Generalized mean class descriptor

One of the simplest way to describe a class relies on the proximity to the 'average' representative. If objects are described as vectors in a feature space, then the mean vector plays such a role. In [301], we discussed that a proximity to the average representative can be formulated in the case, where only a dissimilarity representation is given. The details are also presented in section 4.5, where the generalized nearest mean classifier is discussed.

Assume R represents the target class ω_T. Any symmetric dissimilarity matrix $D(R,R)$ can be seen as a description of an underlying, lower-dimensional pseudo-Euclidean space $\mathcal{E}$ such that the pseudo-Euclidean distances are preserved. Assume that $\mathbf{x} \in \mathcal{E}$ results from the projection of $D(x,R)$. From section 4.5, it is known that the proximity function $f_{\text{proxm}}(D(x,R),\omega_T) = ||\mathbf{x} - \overline{\mathbf{x}}_{\mathcal{E}}||^2_{\mathcal{E}}$ can equivalently be computed by the use of dissimilarities as $f_{\text{proxm}}(D(x,R),\omega_T) = \frac{1}{n}\sum_{i=1}^{n} d^2(x,p_i) - \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(p_i,p_j)$. For the construction of an OCC, the threshold γ can be chosen as the $(1-r_{\text{thr}})$-th percentile of the sorted sequence of $f_{\text{proxm}}(D(t_i,R),\omega_T)$ which express the square distances γ_d. The generalized mean-class data description (GMDD) becomes then:

$$\mathcal{C}_{\text{GMDD}}(D(x,R)) = \mathcal{I}\,(\frac{1}{n}\sum_{i=1}^{n} d^2(x,p_i) - \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d^2(p_i,p_j) \leq \gamma). \tag{8.3}$$

Note that $\mathcal{C}_{\text{GMDD}}(D(x,R)) = \mathcal{I}\,(\frac{1}{n}\,\mathbf{1}^T D^{*2}(x,R) - \frac{1}{2n^2}\,\mathbf{1}^T D^{*2}\mathbf{1} \leq \gamma)$. Since this inequality implicitly corresponds to $||\mathbf{x}_i - \overline{\mathbf{x}}_{\mathcal{E}}||^2_{\mathcal{E}} \leq \gamma$, this OCC basically accepts objects as targets if they lie in a pseudo-Euclidean hypersphere with the radius of $\sqrt{\gamma}$.

In general, $R = T$, however R might consist of a fixed, small fraction of the objects from T. The objects in R should reflect the information on original objects such that the pseudo-Euclidean mean defined by R lies close to the original mean. Our proposal for the selection of R relies on formula (4.23) discussed in section 4.5. There, we showed that given two classes, the difference between the average between-class square dissimilarities and the average within-class square dissimilarities approximates the square pseudo-distance between the two class means in an embedded space. This knowledge can be used as follows. Assume R is randomly chosen out of T. Consider two classes: one defined on R and the other on T. Now, the square pseudo-distance between the two class means can be approximated using formula (4.23). Hence, we can proceed with random choices of R and finally choosing the one which offerers the smallest difference to the mean defined on the complete set. This is computed fast, so e.g. $N = 100$ of possible sets R can be considered. We will refer to this selection as to the *mean-resemblance*.

Some flexibility can be gained by nonlinear transformations of the dissimilarities. As an example, the GMDD is trained on a Euclidean distance representation D of the 2D banana class, as well as its sigmoidal transformations D^s and D^{s^2}. The OCC's boundaries are presented in Fig. 8.3. Since the original representation is Euclidean, then the GMDD on D yields a spherical description (the boundary is defined by the square distance to the mean of the target class), as observed in the figure. The parameters of sigmoidal transformations are not optimized: they were chosen in relation to local neighborhoods. Depending on the parameter of the transformation, the boundary can become either tighter or wider.

Generalized weighted mean class descriptor. Since the GMDD relies on all objects of the set R, a natural extension is to define a similar classifier, but based on some objects only. This leads to the concept of a weighted mean in a pseudo-Euclidean space $\mathcal{E}$. So, $\overline{\mathbf{x}}^{\beta} := \sum_{i=1}^{n} \beta_i \mathbf{x}_i$, where all β_i are nonnegative and $\sum_{i=1}^{n} \beta_i = 1$. Ideally, the β_i should be selected such that many of them are zero and only some of them are positive. This would imply a sparse formulation based on the dissimilarities to the non-zero objects only.

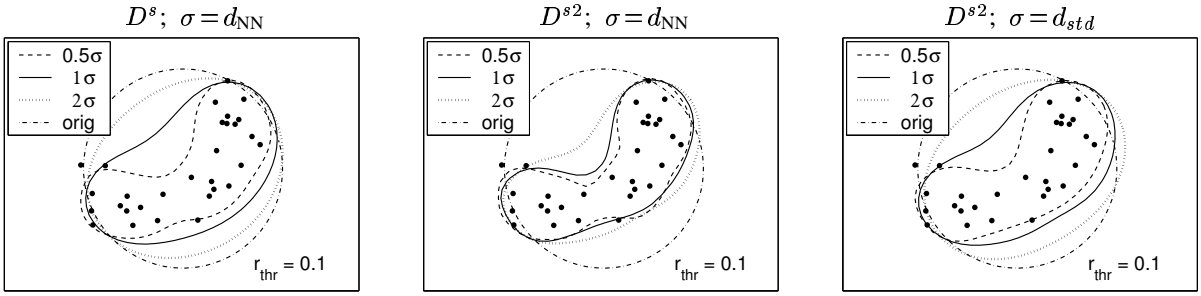


Fig. 8.3: Generalized mean class descriptor (GMDD) for dissimilarity representation D (orig) and its sigmoidal transformations: D^s and D^{s2} of a theoretical banana class. s is a parameter of such transformations. The legend describes various choices of s, i.e. $s = 0.5\sigma, \sigma, 2\sigma$, where σ is defined for the original distances D either as the averaged $\sqrt{N}$-nearest neighbor distance (d_{NN}) or the standard deviation (d_{std}). The threshold $r_{\mathrm{thr}} = 0.1$ has been used.

Remember that for the pseudo-Euclidean considerations, $R \subseteq T$ and $D(R,R)$ should be symmetric. An OCC in the embedded $\mathcal{E}$ can be now designed based on the square distance to the weighted mean of the target class. Similarly as above, let $X := \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ be a vector representation in $\mathcal{E}$ resulting from the embedding of $D(R,R)$. The remaining $T \setminus R$ objects are then projected to $\mathcal{E}$. Let $\overline{\mathbf{x}}^{\beta}_{\mathcal{E}}$ be the weighted mean vector of all objects. The classifier can be now described as a pseudo-Euclidean hypersphere with the center being the weighted mean and some radius $\sqrt{\gamma}$. This leads to the proximity function $f_{\mathrm{proxm}}(D(x,R),\omega_T) = ||\mathbf{x} - \overline{\mathbf{x}}^{\beta}_{\mathcal{E}}||^2_{\mathcal{E}}$. Similarly as above, such a proximity can be equivalently expressed by the dissimilarities only as $f_{\mathrm{proxm}}(D(x,R),\omega_T) = \sum_{i=1}^{n} \beta_i d^2(x,p_i) - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\beta_i\beta_j d^2(p_i,p_j) = \boldsymbol{\beta}^T D^{*2}(x,R) - \frac{1}{2}\boldsymbol{\beta}^T D^{*2}(R,R)\boldsymbol{\beta}$. This can be derived analogously as in section 4.5 by using $\overline{\mathbf{x}}^{\beta}_{\mathcal{E}}$ instead of $\overline{\mathbf{x}}$. Note such a formulation is similar to the support vector data description as proposed by Tax [386, 390]. The difference lies in the fact that Tax reformulates it for kernels, hence positive definite similarity representations, while we focus on general dissimilarities.

The question now arises how the weights β_i should be found. A logical approach relies on determining β_i such that the pseudo-hypersphere has a minimal (positive) radius, hence the square pseudo-Euclidean distances to the weighted mean are minimized (in the pseudo-Euclidean sense). All training objects p_i ($T = R$) may be forced to lie inside the pseudo-Euclidean hypersphere. It may be, however, useful to allow for the rejection of some target examples to obtain a tighter boundary. This can be realized by introducing the nonnegative slack variables ξ_i accounting for possible errors:

$$\begin{aligned} \min\ & \gamma + \tfrac{1}{\nu N}\sum_{i=1}^{N}\xi_i \\ \text{s.t.}\ & \boldsymbol{\beta}^T D^{*2}(t_i,R) - \tfrac{1}{2}\boldsymbol{\beta}^T D^{*2}(R,R)\boldsymbol{\beta} \le \gamma + \xi_i, \quad i = 1,2,..,N \\ & \boldsymbol{\beta}^T\mathbf{1} = 1,\ \beta_i \ge 0,\ \gamma \ge 0,\ \xi_i \ge 0, \end{aligned} \tag{8.4}$$

where $\nu \in (0,1]$ is a user-specified parameter. The idea of using the form of $\frac{1}{\nu N}\sum_{i=1}^{N}\xi_i$ comes from the support-vector research e.g. [349, 350, 390], where it can be proved that ν is an upper bound for the error on the target class. By a proper reformulation, the above optimization is a quadratic programming (QP) problem. By solving it, one hopes to find a sparse solution, i.e. based on a relatively few objects only. The problem, however, lies in solving a *non-convex* QP problem, since the dissimilarity matrix $D^{*2}(R,R)$ is not positive definite (although if D is Euclidean, then $-D^{*2}$ is conditionally positive definite; see a part of the SVC in section 4.5 and section 4.6 on relations between conditionally positive definite and Euclidean matrices). From the Euclidean point of view, the difficulty is caused by the fact that a local optimum can only be found, in contrast to the convex formulation, where the global optimum is guaranteed. So, a suitable (non-standard) software is necessary to solve the problem. We plan to investigate methods of solving this optimization task in our future research. The generalized weighted mean class data description GWMDD would be then

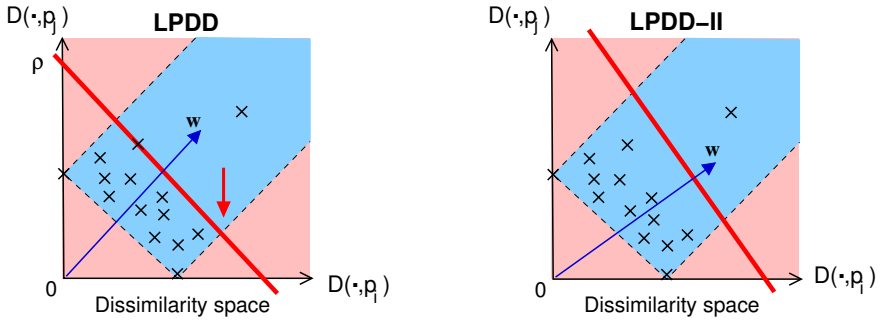


Fig. 8.4: Illustrations of the LPDD (left) and the LPDD-II (right). The dashed lines indicate the boundary of the area which contains the genuine objects if the measure is metric. The LPDD tries to minimize the max-norm distance from the bounding hyperplane to the origin, while the LPDD-II tries to attract the hyperplane towards the average of the distribution. The LPDD-II is defined below.

defined as

$$\mathcal{C}_{\text{WGMDD}}(D(x,R)) = \mathcal{I}\,(\sum_{\beta_k \neq 0} \beta_k D^{*2}(x,p_k) - d_\beta \leq \gamma), \quad d_\beta = \frac{1}{2}\boldsymbol{\beta}^T D^{*2}(R,R)\boldsymbol{\beta}. \tag{8.5}$$

As before, concave transformations can be applied to the dissimilarities to add an extra flexibility.

8.2.3 Linear programming dissimilarity data description

A one-class classifier designed in a dissimilarity space was proposed by us in [306]. When the set R contains objects from the class of interest, then objects x with large $D(x,R)$ are considered as outliers and should be remote from the origin in the dissimilarity space. This characteristic is used in our OCC. If the dissimilarity measure D is a metric, then all vectors $D(x,R)$, lie in a prism, bounded from below by a hyperplane on which the objects from R lie and bounded from above by the largest dissimilarities (we assume that d is bounded if not it can be scaled to be such).

Consider a dissimilarity representation $D(T,R)$, where $R=\{p_1,..,p_n\}$ is the representation set and $T := \{t_1,\ldots,t_N\}$ is a set of objects. Let H be a hyperplane in $\mathbb{R}^n$, i.e. $H := \{\mathbf{x}\in\mathbb{R}^n : \mathbf{w}^T\mathbf{x} = \rho,\ \mathbf{w}\neq\mathbf{0}\in\mathbb{R}^n,\ \rho\in\mathbb{R}\}$ and let $\mathbf{x}\in\mathbb{R}^n$ be any point outside H. $\mathbf{x}$ can be projected onto H by using an arbitrary norm l_p, $p\geq 1$ [257]. Then, the distance between $\mathbf{x}$ and the hyperplane H or, in fact, the distance between $\mathbf{x}$ and its projection $\mathbf{x}_H$ onto H, is measured by the dual norm l_q such that q satisfies $1/p+1/q=1$. It is given by $d_q(\mathbf{x},H)=||\mathbf{x}-\mathbf{x}_H||_q=|\mathbf{w}^T\mathbf{x}-\rho|/||\mathbf{w}||_p$.

To describe a class in a non-negative dissimilarity space, one could minimize the volume of the prism, cut by a hyperplane $H:\mathbf{w}^T D(x,R)=\rho$ suppressing the data from above (in general H is not expected to be parallel to the prism's bottom hyperplane); see Fig. 8.4. In this case, non-negative dissimilarities impose both $\rho\geq 0$ and $w_i\geq 0$. However, this task might not be feasible. A natural extension is to minimize the volume of a simplex with the main vertex coinciding with the origin of a dissimilarity space and the other vertices, say $\mathbf{v}_j$, resulting from the intersection of H and the axes of a dissimilarity space. Note that $\mathbf{v}_j$ is a vector of all zero elements except for $v_{ji}=\rho/w_i$, provided that $w_i\neq 0$. Assume now that there are $r<n$ non-zero weights of the hyperplane H, so effectively, H is constructed in a $\mathbb{R}^r$. From geometry, we know that the volume V of such a simplex can be expressed as $(V_{\text{base}}/r!)\cdot(\rho/||\mathbf{w}||_2)$, where V_{base} is the volume of the base, defined by the vertices $\mathbf{v}_j$. The minimization of $h=\rho/||\mathbf{w}||_2$, i.e. the Euclidean distance from the origin to H, is then related to the minimization of V.

Let $D(T,R)$ be a dissimilarity representation bounded by the hyperplane H, i.e. $\mathbf{w}^T D(t_i,R)\leq\rho$ for $i=1,\ldots,n$, such that the l_q distance to the origin $d_q(\underline{0},H)=\rho/||\mathbf{w}||_p$ is minimal (remember that q satisfies $1/p+1/q=1$ for $p\geq 1$, since l_q and l_p are the dual norms) [257]. This means that H can be

determined by minimizing $\rho - ||\mathbf{w}||_p$. However, in order to avoid any arbitrary scaling of $\mathbf{w}$, we may require that $||\mathbf{w}||_p = 1$. Then, the construction of H can be solved by the minimization of ρ only. The mathematical programming formulation of such a problem is [20, 257]:

$$\begin{aligned} &\min \ \rho \\ &\text{s.t.} \quad \mathbf{w}^T D(t_i, R) \leq \rho, \quad i = 1, 2, .., N, \\ &\qquad ||\mathbf{w}||_p = 1, \quad \rho \geq 0. \end{aligned} \tag{8.6}$$

(Note that from the algorithmic point of view, the assumption of $||\mathbf{w}||_p = 1$ requires that the dissimilarities are bounded by not too large values, e.g. 1 or 10, otherwise ρ should become very large to fulfill the constraints, which might cause the problem be unbounded.)

If $p = 2$, then the hyperplane H is found when h - the Euclidean distance from the origin to H is minimized. However, the problem is then formulated by a quadratic optimization. A simpler, linear programming (LP) formulation is of interest to us. This can be realized for $p = 1$. Knowing that $||\mathbf{w}||_2 \leq ||\mathbf{w}||_1 \leq \sqrt{n}||\mathbf{w}||_2$ and by the assumption of $||\mathbf{w}||_1 = 1$, after simple calculations we find out that $\rho < h = \rho/||\mathbf{w}||_2 < \sqrt{n}\,\rho$. Therefore, by minimizing $d_\infty(\mathbf{0}, H) = \rho$ (and requiring that $||\mathbf{w}||_1 = 1$), h will be bounded, (for a fixed and small R, the minimization of ρ bounds h) and, therefore, the volume of the considered simplex, as well.

By the above reasoning and formula (8.6), a class represented by dissimilarities can be characterized by a linear proximity function with the weights w_j and the threshold ρ. Such a hyperplane simply 'pushes' the objects in the direction of the origin in the dissimilarity space. Our one-class classifier, the Linear Programming Dissimilarity-data Description $\mathcal{C}_{\text{LPDD}}$ is then defined as:

$$\mathcal{C}_{\text{LPDD}}(D(x, R)) = \mathcal{I}\Big(\sum_{w_j \neq 0} w_j D(x, p_j) \leq \rho\Big). \tag{8.7}$$

The proximity function is found as the solution to a soft-margin formulation[3], which is a straightforward extension of the hard-margin case (by neglecting the slack variables), as:

$$\begin{aligned} &\min \ \rho + \tfrac{1}{\nu N} \textstyle\sum_{i=1}^{N} \xi_i \\ &\text{s.t.} \quad \mathbf{w}^T D(t_i, R) \leq \rho + \xi_i, \quad i = 1, 2, .., N \\ &\qquad \mathbf{w}^T \mathbf{1} = 1, \ w_i \geq 0, \ \rho \geq 0, \ \xi_i \geq 0, \end{aligned} \tag{8.8}$$

where ξ_i are the slack variables, allowing objects to lie above the hyperplane, i.e. accommodating some targets as outliers. In the LP formulations, sparse solutions are obtained, meaning that only some weights w_j are positive. Objects corresponding to such non-zero weights, will be called *support objects* (SO). (Note that they cannot be called support vectors, since they directly refer to the objects and not to their vector representations.) These support objects construct the effective representation set R_e, $|R_e| = r$ and, as a result, test objects need to be evaluated by computing dissimilarities to objects from R_e only.

The left plot of Fig. 8.4 shows a two-dimensional pictorial illustration of the LPDD. The data are represented in a metric dissimilarity space, and by the triangle inequality, the dissimilarities can only lie inside the prism indicated by the dashed lines. The LPDD boundary is given by the hyperplane, as close to the origin as possible in terms of the l_∞ distance (determined by the minimization of ρ), while still accepting (most) target objects[4]. The outliers should be remote from the origin.

[3] We abuse here somewhat the soft-margin and hard-margin formulation from the support vector research e.g. [349, 350, 390], where it can be proved that ν is an upper bound for the error on the target class.

[4] This picture might be misleading. In a $\mathbb{R}^n$ space, all representation objects lie in $(n-1)$-dimensional subspaces determined by all but one basis axes. E.g. in a 3D, the objects from R are placed on the $xy-$, $xz-$ and yz-planes.

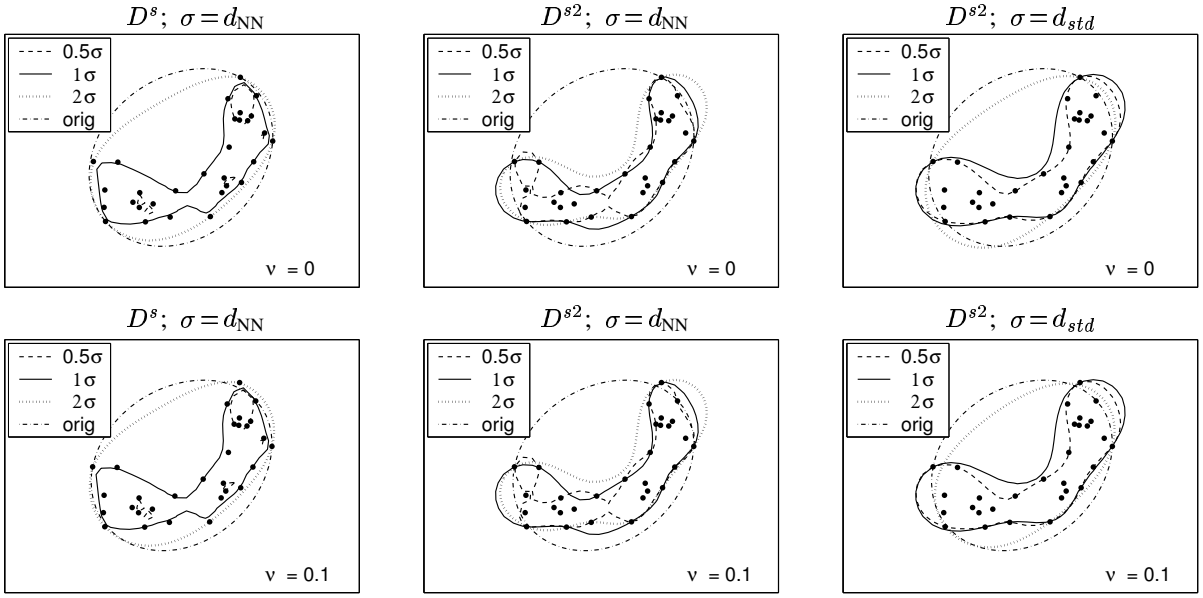


Fig. 8.5: Linear programming data description for a dissimilarity representation D (orig) and its sigmoidal transformations D^s and D^{s2} for a theoretical banana class. Various boundaries are shown depending on the choice of s; $s = 0.5\sigma, \sigma, 2\sigma$, where σ is defined for the original distances D either as the averaged $\sqrt{N}$-nearest neighbor distance (d_{NN}) or the standard deviation (d_{std}). The number of support objects varies from $2, 3$ for the LPDD based on original distances (marked by a dash-dotted line), 6 for $s = 2\sigma$ and up to 18 for $s = 0.5v$. The upper plots refer to $\nu = 0$ and the bottom plots refer to $\nu = 0.1$.

Proposition* 8.1 Consider formula (8.8) for $D(T,T)$. Then, $\nu \in (0,1]$ is the upper bound on the outlier fraction for the target class, i.e. the fraction of objects that lie outside the boundary; see also [349, 350]. This means that $\frac{1}{n}\sum_{i=1}^{N}(1 - \mathcal{C}_{\mathrm{LPDD}}(D(t_i,T)) \leq \nu$.

Sketch of proof. The proof goes analogously to the proofs given in [349, 350]. Intuitively, these proofs follow the reasoning given as: assume we have found a solution of (8.8). If ρ is increased slightly, the term $\sum_i \xi_i$ in the objective function will change proportionally to the *number* of points that have non-zero ξ_i (i.e. the rejected target objects). At the optimum of (8.8), therefore, it has to hold $N\nu \geq$ number of outliers.

As before, nonlinear transformations of dissimilarities can be used. The LPDD is trained on a Euclidean distance representation D of the 2D banana class, as well as on sigmoidal transformations D^s and D^{s2}. The OCC's boundaries are presented in Fig. 8.5. As it can be observed, such a LP formulation offers flexible descriptions of the data boundary depending on the transformation.

Using outlier information. The LPDD can straightforwardly be extended to handle example outliers. This means that T contains some outliers. The representation set R can also contain some outliers. If the problem to be solved describes the targets against 'pure' non-targets (healthy versus diseased people), we think that the instances of R should belong to the target class, otherwise the outliers from R may become support objects, hence objects which determine the decision. This point, however, needs to be investigated further.

In the LPDD, the hyperplane in a dissimilarity space is attracted towards the origin and the objects are placed in the half-space below this hyperplane. In fact, they lie in a simplex with the main vertex coinciding with the origin and the other vertices resulting from the intersection of the hyperplane and the axes of this dissimilarity space. This is described by the constraint $\mathbf{w}^T D(t_i, R) \leq \rho + \xi_i$, where $\xi_i \geq 0$ account for possible errors, such that the targets can be found above the hyperplane. This is the place, where the outliers should lie. If we allow some outliers to be accepted as targets, this would lead to $\mathbf{w}^T D(t_i, R) \geq \rho - \xi_i$, assuming that t_i are now outlier examples. A variable $y_i \in \{+1, -1\}$ can

be used to denote the targets as 1 and outliers as –1. The formulation (8.8) remains then the same except that the constraint there changes to $y_i\,(\mathbf{w}^T D(t_i,R)) \leq y_i\rho + \xi_i$. This constraint simply forces the known outliers ($y_i = -1$) to be placed in the right half-space of the hyperplane.

Linear programming dissimilarity data description II. A linear programming formulation for novelty detection has also been proposed in [53]. The reasoning there starts from a feature space in the spirit of positive definite kernels $K(S,S)$ based on the vector set $S = \{\mathbf{x}_1, .., \mathbf{x}_N\}$. The authors restricted themselves to (modified) RBF kernels, i.e. for $K(\mathbf{x}_i, \mathbf{x}_j) = e^{-D(\mathbf{x}_i,\mathbf{x}_j)^2/2s^2}$, where D is either Euclidean or l_1 distance. In principle, we will refer to RBFp, as to the 'Gaussian' kernel based on the l_p distance. Here, to be consistent with our LPDD method, we will rewrite their soft-margin LP formulation (a hard-margin formulation is then obvious by neglecting the slack variables), to include a trade-off parameter ν as follows:

$$\begin{array}{ll} \min & \frac{1}{N}\sum_{i=1}^N (\mathbf{w}^T K(\mathbf{x}_i, S) + \rho) + \frac{1}{\nu N}\sum_{i=1}^N \xi_i \\ \text{s.t.} & \mathbf{w}^T K(\mathbf{x}_i, S) + \rho \geq -\xi_i, \quad i = 1, 2, .., N \\ & \mathbf{w}^T \mathbf{1} = 1,\ w_i \geq 0,\ \xi_i \geq 0. \end{array} \tag{8.9}$$

Unfortunately, μ now lacks the interpretation as given in the LPDD case. $\frac{1}{\nu N}$ is a trade-off parameter, relating different quantities, i.e. weighting the error contributions and the average classifier output. From our point of view, K can be any similarity representation, moreover, not necessarily square (in the same way as the LPDD is defined for general dissimilarity representations). So, for simplicity, we can denote this method as Linear Programming Similarity-data Description (LPSD).

Following the principles as described above, one can consider an equivalent formulation for the LPDD. Including also the information on possible outliers ($y_i = -1$), the soft-margin LPDD-II (a hard-margin LPDD-II is then obvious) becomes then:

$$\begin{array}{ll} \min & \frac{1}{N}\sum_{i=1}^N (\rho - \mathbf{w}^T D(t_i, R)) + \frac{1}{\nu N}\sum_{i=1}^N \xi_i \\ \text{s.t.} & y_i\,\mathbf{w}^T D(t_i, R) \leq y_i\rho + \xi_i, \quad i = 1, 2, .., N \\ & \mathbf{w}^T \mathbf{1} = 1,\ w_i \geq 0,\ \rho \geq 0,\ \xi_i \geq 0. \end{array} \tag{8.10}$$

Similarly to the LPDD, a sparse solution is obtained. Hence, also the objects corresponding to non-zero weights are the support objects. The $\mathcal{C}_{\text{LPDD-II}}$ is defined identically as the $\mathcal{C}_{\text{LPDD}}$ in (8.7). The difference lies in the way how the weights $\mathbf{w}$ are found during the training process.

If only target objects are given, the hyperplane is determined such that its averaged output is attracted towards origin. Hence, such a formulation may lead either to a narrow description of the target class (narrower than the LPDD) for a compact class or to a wide description of the target class, when there are examples lying further away than the main bulk of the data. See Fig. 8.4 for an illustration of such a case. However, when outlier examples are present, this might be advantageous, since the outliers influence the average output and, as a result, a hyperplane 'balanced' in-between the targets and outliers can be determined.

Here, to be consistent with our dissimilarity approaches, we will focus on the dissimilarity-based OCCs. For some remarks concerning the LPSD, see our paper [306].

8.2.4 More issues on class descriptors

There are additional points to be discussed concerning the class descriptors, especially the LP classifiers. An important point refers to the support objects of the LPDD and the LPDD-II. There is an essential difference between such support object and to a similar concept of support vectors in the SVC terminology. Since we operate on dissimilarities, the chosen support objects are related

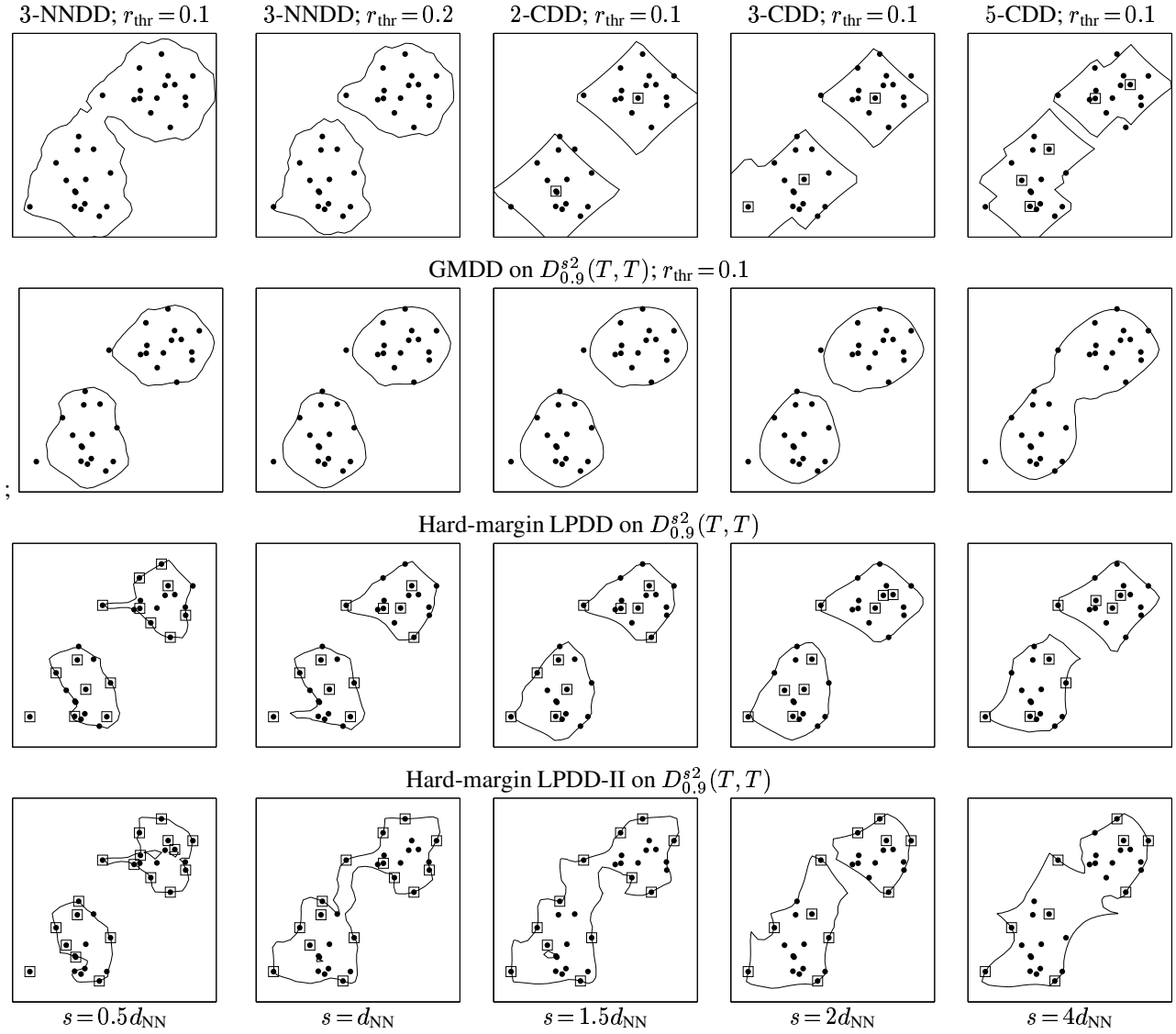


Fig. 8.6: One-class classifiers: neighborhood-based OCCs (the k-NNDD and the k-CDD in the top row), the GMDD (second row from the top) and hard-margin LP classifiers for the two cluster data represented by the non-metric $l_{0.9}$ distances $D_{0.9}(T,T)$ and their sigmoidal transformation $D^{s2}_{0.9}$. The neighborhood-based OCCs are designed on the original dissimilarities (since they are not influenced by sigmoidal transformations), while other OCCs are built on their transformed versions. In the latter case, the slope parameter s is fixed, such as $s = 0.5 d_{\text{NN}}, 0.8 d_{\text{NN}}, d_{\text{NN}}, 1.5 d_{\text{NN}}, 2 d_{\text{NN}}$, where d_{NN} is the averaged $\sqrt{N}$-nearest neighbor distance. The OCCs boundaries for the LP classifiers are originally determined by a hyperplane in a dissimilarity space, which correspond to nonlinear boundaries in the input space. Support objects are marked by squares. Note that in case of the LPDD-II, the support objects tend to lie on the boundary, which might not be a case for the LPDD. Since the boundary of the k-CDD is determined by k objects, these are also marked by squares (see top row).

to the dimensions of a dissimilarity space. The boundary of the OCCs is determined by a suitable hyperplane on which some objects are likely to lie. In general, such boundary objects are different than the support objects, although they may coincide. The boundary objects are in principle objects which are far away from the origin, hence they influence the hyperplane weights, hence the support objects. This means that by removing support objects from the target class, the OCC's boundary may remain nearly unchanged (if there are other objects close the removed support objects), however, by a removal of a far-away boundary object, other support objects can be chosen. This is in agreement with the support-vector formulations, where support objects are boundary objects.

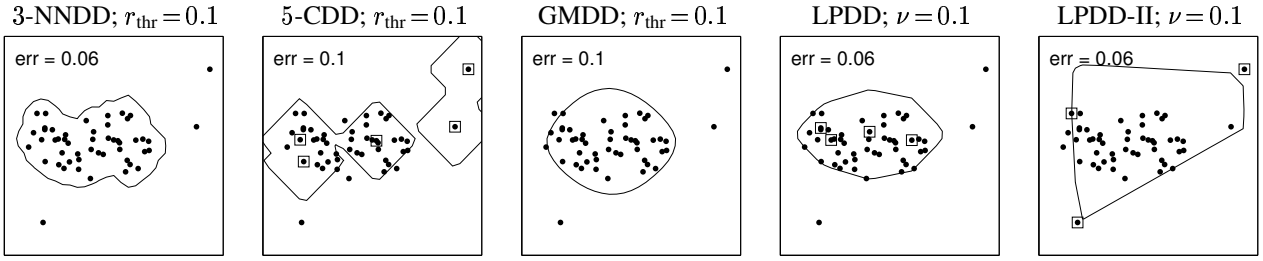


Fig. 8.7: The OCCs designed for the uniform cloud with outliers. The data are represented by the city block distances $D_1(T,T)$. Support objects are marked by squares. Note that the k-CDD might not be able to disregard outliers in the target class. Note also that the LPDD defined by four support objects defines a reasonably tight boundary around the cloud.

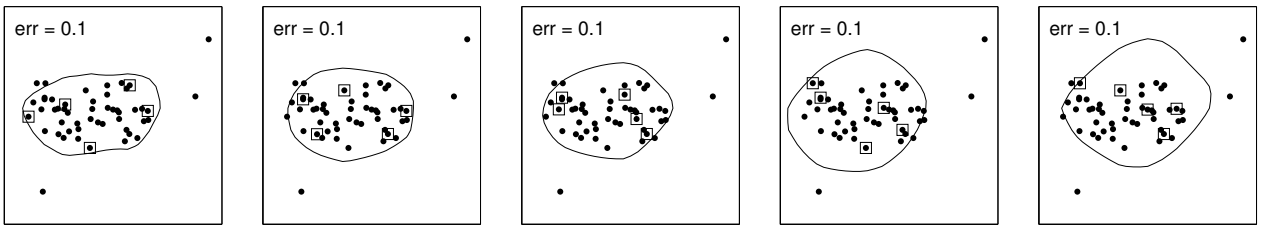


Fig. 8.8: The GMDD built on a sigmoidal-I transformations of the l_1 distance representation D_1 designed for the uniform cloud with outliers. Five objects are selected from T based on the mean-resemblance criterion (section 8.2.2), for the representation set $R \subset T$. They are marked by squares. All training data points are assumed to be targets. The GMDD is trained with $r_{\text{thr}} = 0.1$. The plots presents the boundaries in the input space, which are originally considered in the embedded pseudo-Euclidean spaces defined by $D_1^s(T,R)$ and $D_1^{s2}(T,R)$. From left to right, s takes the values of $0.5d_{\text{me}}, d_{\text{me}}, 1.5d_{\text{me}}, 2d_{\text{me}}$ and $4d_{\text{me}}$, where d_{me} is the average distance. *err* in the plots refers to the effective error on the target set.

In general, all class descriptors mentioned here are able to uncover clusters in the data, as well as, 'outliers' present in the target class, provided that proper parameters are specified. Consider two 2-dimensional artificial data sets. The first set consists of two clusters of 15 points each. It will be denoted as the *two-cluster* data. These data are represented by a non-metric $l_{0.9}$ distance. The second set contains one uniform, rectangular cluster, contaminated with three outlier points. It will be denoted as the *uniform cloud with outliers* data. In total, 50 points are given. For a dissimilarity representation, the l_1 (city block) distance is used. (We have explicitly chosen non-Euclidean distances to show that any distance can work.) Now, the one-class classifiers are trained. Since the artificial data are 2-dimensional, it is possible to draw the decision boundaries in the 2D input space, even if the OCCs are trained in (high-dimensional) dissimilarity spaces. Figures 8.6 - 8.10 show the boundaries of various dissimilarity-based class descriptors trained on square dissimilarity matrices $D(T,T)$ or their sigmoidally transformed versions.

In Fig. 8.6, the two-cluster data set and decision boundaries of the trained OCCs are shown. The neighborhood-based OCCs are designed on the original dissimilarities $D_{0.9}$ since they are hardly influenced by concave non-decreasing transformations. Other OCCs are built on their sigmoidally transformed versions $D_{0.9}^s$ and $D_{0.9}^{s2}$. Although the OCCs are trained on square dissimilarity representations, the LP classifiers offer sparse solutions by choosing a number of support objects, i.e. objects which determine the boundary and to which the dissimilarities should be computed in a testing stage. Basically, the number of support objects is related to the complexity of the boundary, which can be observed while comparing the leftmost and rightmost plots of the LPDD and the LPDD-II in Fig. 8.6. The k-CDD is also determined by a small number of objects, namely k objects, where k is specified beforehand. (This is the difference with the LP classifiers, where the support objects are specified by ν and are determined in the response to the mathematical programming

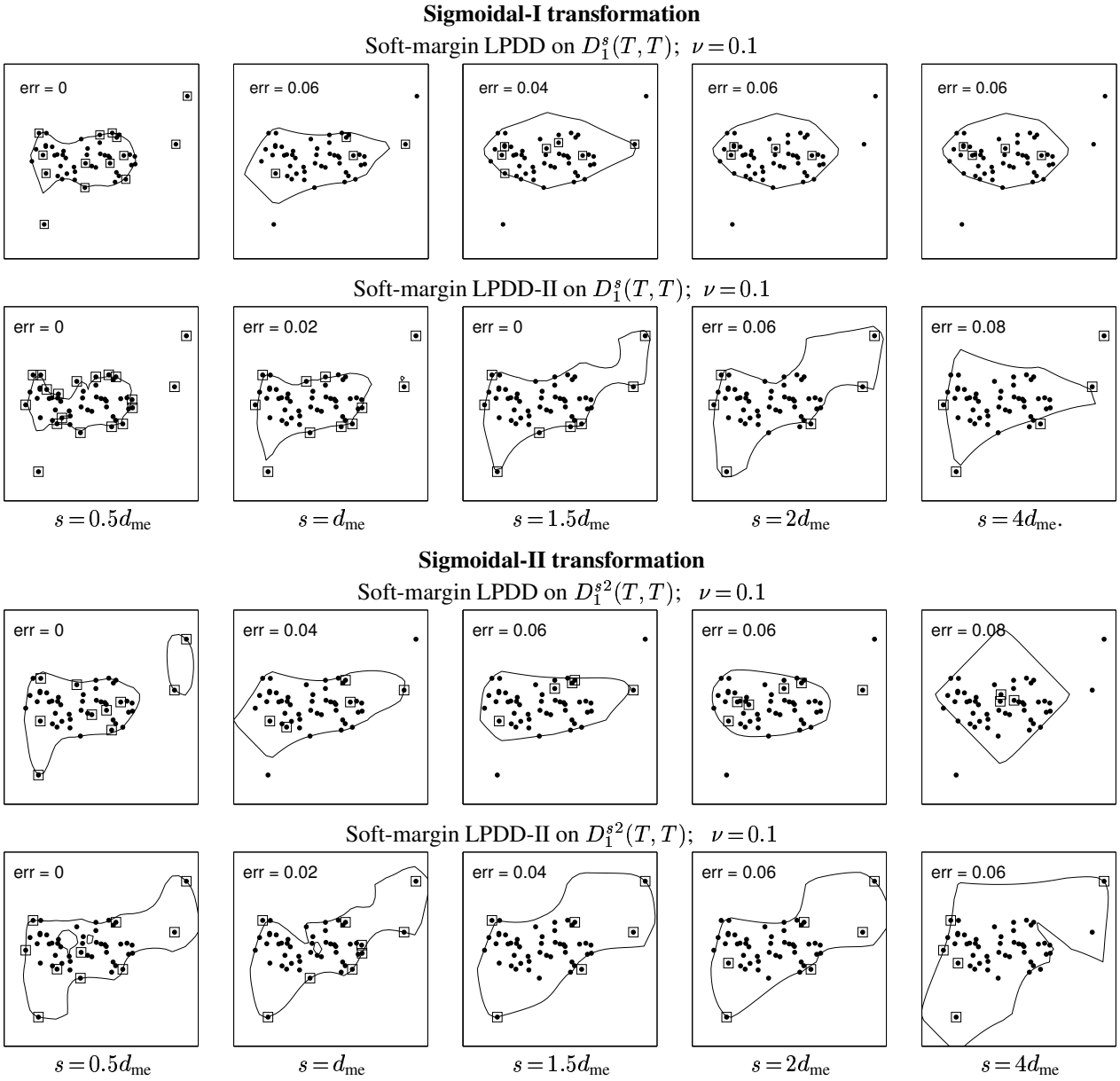


Fig. 8.9: One-class soft-margin LP classifiers, trained with $\nu = 0.1$, designed for the uniform cloud with outliers. All training data points are assumed to be targets. The data are described by the l_1 distance representation D_1. However, here, its sigmoidal transformations are used. Ideally, the OCCs should disregard at maximum 10% of the points. The plots presents the boundaries in the input space, which are originally found in dissimilarity spaces D_1^s and D_1^{s2}. From left to right, s takes the values of $0.5d_{\text{me}}, d_{\text{me}}, 1.5d_{\text{me}}, 2d_{\text{me}}$ and $4d_{\text{me}}$, where d_{me} is the average distance. err in the plots refers to the effective error on the target set. Support objects are marked by squares. They belong to the OCCs.

formulation.) On the contrary, the k-NNDD and the GMDD require all objects for the boundary construction.

Two cases are considered for the artificial data with three outliers. First, all the data points are assumed to be targets and soft-margin LP classifiers with $\nu=0.1$ are trained. Then, these three outliers should possibly be ignored. This can be observed in some plots of Fig. 8.7 and Fig. 8.9, provided that a proper scaling parameter s of the sigmoidal transformations is used. Another possibility is to label the outliers appropriately and use them in the training of the LP classifiers (other classifiers, the k-CDD, the k-NNDD and the GMDD cannot directly incorporate such label information). So, in the training set, 47 points are labeled as targets and three points as outliers. Given that, it is sufficient

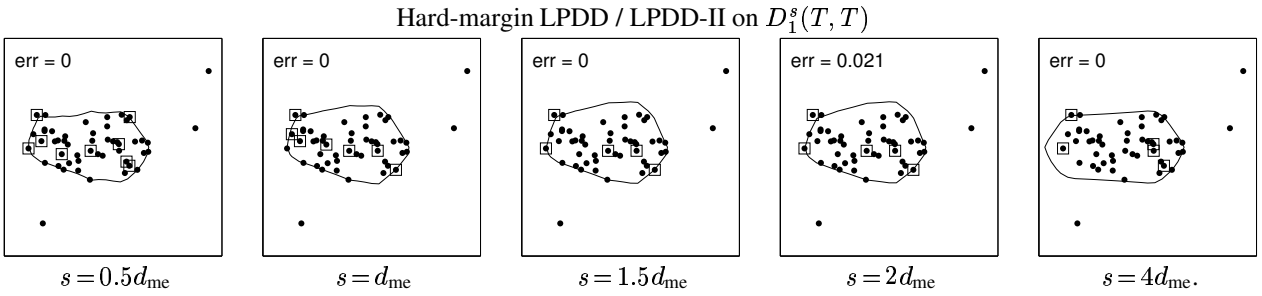


Fig. 8.10: One-class hard-margin LP classifiers designed for the uniform cloud with three outliers. In a training stage, these three outliers are labeled so. Hence, they should be disregarded by the OCCs (except for the outliers, the data points make are a relatively compact cloud, so a hard-margin OCC should describe it well). The data are described by the l_1 distance representation D_1 and its sigmoidal transformation. The plots show the boundaries in the input space, which are originally found in dissimilarity spaces D_1^s. From left to right, s takes the values of $0.5d_{\text{me}}, d_{\text{me}}, 1.5d_{\text{me}}, 2d_{\text{me}}$ and $4d_{\text{me}}$, where d_{me} is the average distance. err in the plots refers to the effective error on the target set. Support objects are marked by squares. The results are the same for both the LPDD and the LPDD-II. Since the cloud is compact and the outliers are disregarded, both LP classifiers return the same support objects, hence the same boundaries.

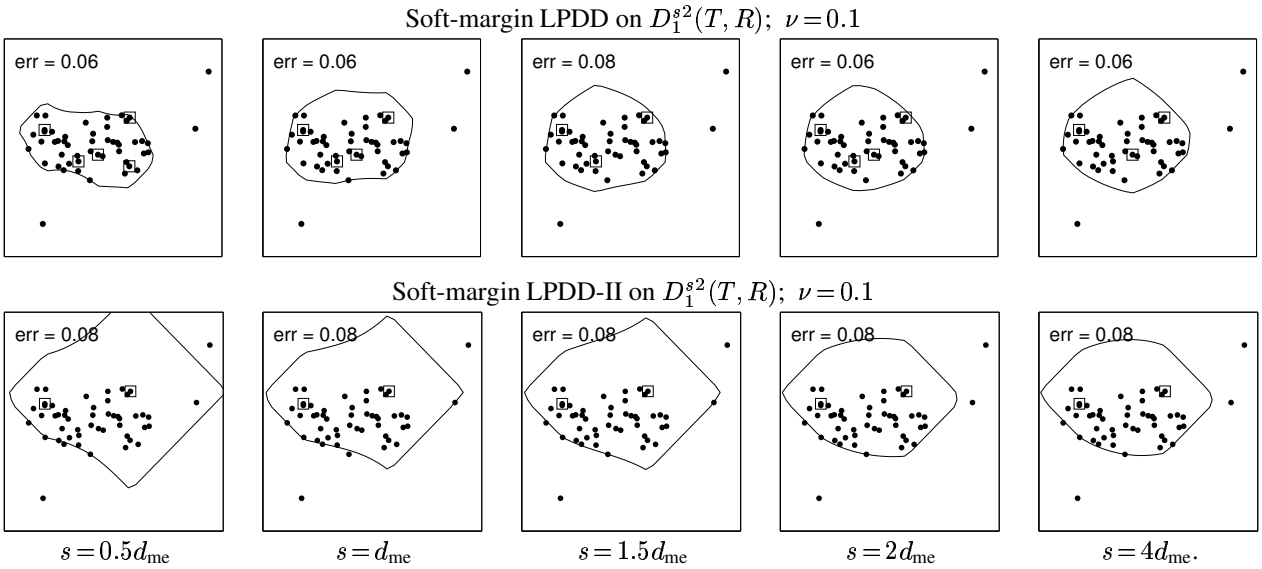


Fig. 8.11: One-class soft-margin LP classifiers designed for the uniform cloud with outliers. T contains 50 training points. Five randomly chosen points from T are assigned to R. The data are described by the l_1 distance representation $D_1(T,R)$. In the LP classifiers, its sigmoidal transformation $D_1^{s2}(T,R)$ is used. The rows show the results in the input space of the boundaries originally found by the sigmoidally transformed l_1 distances in a dissimilarity space. From left to right, s takes the values of $0.5d_{\text{me}}, d_{\text{me}}, 1.5d_{\text{me}}, 2d_{\text{me}}$ and $4d_{\text{me}}$, where d_{me} is the average distance. err refers to the effective error on the target set. Support objects are marked by squares. Note that the support object come from R, so there might be maximum five support objects.

to train hard-margin LP classifiers, since the remaining points make a compact cloud. The results are presented in Fig. 8.10. While soft-margin LP OCCs trained on the targets seem to be highly influenced by the slope parameter s (Fig. 8.9), the hard-margin classifiers, trained by using also the outlier information, seem to be much less.

When the LP classifiers are designed by treating all the points as targets, it is much harder for the LPDD-II to disregard the three outliers than for the LPDD; compare the boundaries of the LPDD and the LPDD-II in Fig. 8.7 and Fig. 8.9. This is not surprising, since the boundary of the LPDD-II is determined by taking into account the averaged dissimilarity output to which outliers significantly

contribute. In such cases (where only the target data are provided, yet, possibly containing some 'outlier' examples), logically, it seems more reasonable to use the LPDD. On the other hand, when outlier information is used for training, the LPDD-II might work better. In our case, however, Fig. 8.10, both the LPDD and the LPDD-II determine the same support objects, hence find the same boundary. They both provide a tight description around the uniform cloud and they seem not to depend much on the parameter of the sigmoidal transformation.

Additionally, Fig. 8.11 shows some results for a rectangular dissimilarity representation $D(T, R)$, in which just five points are randomly chosen from T for the set R. The results are obtained assuming that all the points constitute the target class. In such a case, the support objects come from R, so there can be maximally five of them. Since the boundary relies on the dissimilarities to a few objects only, its flexibility is limited. The boundary changes only somewhat with growing parameter s of sigmoidal transformations; compare Fig. 8.11 with two bottom rows of Fig. 8.9. It might be therefore useful to pre-select a representation set R smaller than the original training set T.

8.3 Experiments

In this section, we will now present how the introduced one-class classifiers work in practice.

8.3.1 Experiment I: Condition monitoring

Fault detection is an important problem in the machine diagnostics: failure to detect faults can lead to machine damage, while false alarms can lead to unnecessary expenses. As an example, we will consider a detection of four types of fault in ball-bearing cages, a data set from the Structural Integrity and Damage Assessment Network [124] considered in [53]. Each data instance consists of 2048 samples of acceleration. After pre-processing with a discrete Fast Fourier Transform, each signal is characterized by 32 attributes, which is a sparse sampling. The data set consists of five categories: normal behavior NB, corresponding to the measurements made on new ball-bearings and four types of anomalies, A_1 - A_4. See appendix A.2 for further description. Here, we performed experiments in the same way, as described in [53], making use of the same training set, and independent validation and test sets as defined in Fig. 8.12.

	Train	Validation	Test
NB	913	913	913
A_1		747	747
A_2		996	996
A_3			996
A_4			996

Fig. 8.12: Class cardinalities in fault detection data.

Since there is no prior information available on suitable dissimilarity measures, three simple measures are used: Euclidean (l_2 distance), city block (l_1 distance) and non-metric $l_{0.8}$ distance. Hence, we analyze three different dissimilarity representations: $D_{0.8}$, D_1 and D_2. All the dissimilarity representations are scaled by $1/100$, which basically corresponds to a change of 'unit' and it is performed to avoid too large dissimilarities (our linear programming implementation get stuck if the dissimilarities become too large). For each DR, three different concave transformations are studied: a power transformation with the parameter p and sigmoidal-I and sigmoidal-II transformations described by the parameter s; see section 8.2. Since neighborhood-based OCCs are hardly influenced by such transformations, they are not applied in this case.

The OCCs are trained on (transformed) representations defined for the NB class, i.e. they are based on $D(T, R)$, where D now stands for a chosen dissimilarity representation, T is a training set consisting of 913 examples of the NB class and $R \subseteq T$ is a representation set. Two cases are here considered: either R is equivalent to T (which means that an OCC is trained on a square dissimilarity matrix) or R consists of 20% of the target examples selected by the k-centers algorithm (which means that an OCC is trained on a rectangular dissimilarity matrix). The optimal values of either p (power transformation), s (sigmoidal transformation) or k (parameter for the k-CDD and k-NNDD)

are found using the validation set consisting of examples coming from the NB class and two outlier subclasses: A_1 and A_2. There is no unique way of determining a good parameter. In our case, we train the OCCs to reject not more than 1% of the targets[5], so we automatically choose the parameter s (p or k) such corresponds to the smallest mean error averaged over the NB class and two outlier subclasses on the validation set. If there is a sequence of such parameters for which the same error is reached, as a final parameter we choose its median. Note that this is a rough approach, since in practice one may weight the error contributions depending on what is more costly: a false alarm or missing the machine fault. One may also wish to keep e.g. the percentage of the target rejection under a chosen value, which would lead to some other way of establishing the parameter value. Since we cannot decide what are the factors to be taken into account in such a decision, an automatic procedure based on the averaged error (on targets and outliers) seems appropriate to follow.

To select a suitable s for sigmoidal-I and sigmoidal-II transformations, the range of $[0.1d_{avr}, 5d_{avr}]$, where d_{avr} is the average distance within the target class, was considered (for smaller values, our optimization procedure for the LP classifiers did not terminate). k was selected as the best integer between 1 and 50 on the validation set.

The errors of the first kind (classifying targets as outliers, i.e. false alarms) and the second kind (classifying outliers as targets, missing fault detection) kind for the OCCs built on the $D_{0.8}$ and D_1 representations are shown in Tables 8.1-8.3. As expected, sigmoidal transformations offer more flexibility. However, a more important observation is that the $l_{0.8}$ distance measure seems to be more advantageous than both metrics l_1 or l_2. The results for the Euclidean dissimilarity representations and their transformed versions are very bad (much worse than for the city block representation D_1), so, thereby, we have limited ourselves to present only the results for $D_{0.8}$ and D_1.

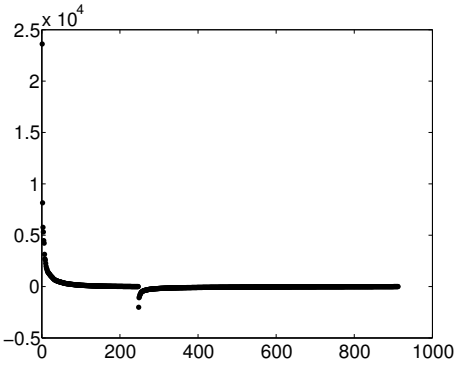


Fig. 8.13: Eigenvalues in the embedding of $D_{0.8}(T, T)$.

To judge the results, two factors are especially taken into account: the error on the target class which should be kept small, possibly round 1% and the error on the two outliers subclasses A_3 and A_4, which are novel to the classifiers. A number of conclusions can be drawn from Tables 8.1-8.3:

(1) In general, the performance of the OCCs is significantly better for $D_{0.8}$ than for D_1 (which in turn is much better than by using D_2). Also, sigmoidal transformations offer more flexibility and better results for the GMDD and the LP classifiers than the power transformations.

(2) The best overall performance is reached for the 1-NNDD on $D_{0.8}$ trained with $r_{\text{thr}} = 0.05$; see Table 8.2. The 1-NNDD yields the errors of 9.9% and 8.3% on the A_3 and A_4 outlier subclasses, respectively, while maintaining the zero errors for the other outlier subclasses and an error of 1.4% for the normal class. The errors on D_1 are increased to 2.1% and 9.8% for the subclasses A_3 and A_4 and to 1.6% for the normal class. In both cases, such a performance is achieved based on dissimilarities to all 913 training examples. When 183 objects are used, then the performance deteriorates, becoming worse than the one reached by the LP classifiers defined on less than 20 support objects.

On the other hand, since the boundary made by the 1-NNDD is very wide around the data (see e.g. Fig. 8.2), a larger threshold (i.e. 0.05) on the dissimilarities should be used.

(3) The best LP performance is reached for the LPDD-II with $\nu = 0.01$ trained on a sigmoidal-I transformation of $D_{0.8}$; see Table 8.2. The errors on the A_3 and A_4 outlier subclasses are

[5] This statement is only approximately true. If $\nu = 0.01$ for the LPDD, then 1% is the maximum error on the target class. This is, however not guaranteed for the LPDD-II. In case of other OCCs, a threshold r_{thr} is set up to e.g. 0.01.

Table 8.1: The errors of the first and second kind (in %) of the OCCs trained on the $l_{0.8}$ distance representation $D_{0.8}(T,R)$ and l_1 distance representation $D_1(T,R)$ for the ball-bearing data. T is the target class (normal behavior) consisting of 913 samples. R is a subset of 183 (20%) examples from T chosen by the k-centers algorithm with $k=183$. R_e is the effective set of objects on which the constructed OCCs rely. The optimal $p \in \{1, 0.8, 0.5, 0.3\}$ of a power transformation is chosen based on the performance on the validation set.

OCC	ν or r_{thr}	Optimal p	$\lvert R_e \rvert$	Validation errors [%]		Test errors [%]				
				NB	A_1+A_2	NB	A_1	A_2	A_3	A_4
LPDD on $\mathbf{D^P_{0.8}}$										
$D(T,T)$	$\nu=0.00$	$p=0.5$	12	0.8	1.1	1.2	0.0	1.2	19.6	19.2
$D(T,R)$		$p=0.3$	8	0.7	1.8	0.9	0.0	3.1	27.0	29.2
$D(T,T)$	$\nu=0.01$	$p=0.5$	12	1.0	0.9	1.3	0.0	1.2	**19.7**	**18.4**
$D(T,R)$		$p=0.3$	11	0.8	1.6	1.2	0.0	2.1	24.0	24.3
LPDD on $\mathbf{D^P_1}$										
$D(T,T)$	$\nu=0.00$	$p=0.5$	11	0.7	2.1	1.3	0.0	3.6	27.5	27.6
$D(T,R)$		$p=0.5$	8	0.8	1.8	1.0	0.0	2.8	28.3	29.5
$D(T,T)$	$\nu=0.01$	$p=0.3$	17	1.5	2.1	2.2	0.0	3.5	29.5	30.1
$D(T,R)$		$p=0.3$	8	1.9	2.6	1.3	0.0	4.3	29.3	29.7
LPDD-II on $\mathbf{D^P_{0.8}(T,R)}$										
$D(T,T)$	$\nu=0.00$	$p=0.8$	15	1.4	0.5	1.8	0.0	0.8	**19.2**	**16.2**
$D(T,R)$		$p=1$	1	0.0	53.2	0.0	1.5	91.4	96.0	97.6
$D(T,T)$	$\nu=0.01$	$p=1$	11	1.6	0.7	1.5	0.0	1.1	23.6	21.4
$D(T,R)$		$p=1$	1	1.1	48.1	1.1	0.4	83.4	93.2	94.4
LPDD-II on $\mathbf{D^P_1}$										
$D(T,T)$	$\nu=0.00$	$p=0.8$	15	1.5	0.6	1.8	0.0	1.0	18.9	17.2
$D(T,R)$		$p=1$	1	0.0	57.1	0.0	4.1	97.1	99.1	98.9
$D(T,T)$	$\nu=0.01$	$p=0.8$	13	1.5	0.6	1.9	0.0	1.0	18.6	17.2
$D(T,R)$		$p=1$	1	1.1	53.9	1.3	1.7	92.2	96.8	97.6
GMDD on $\mathbf{D^P_{0.8}}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$p=0.3$	913	0.1	50.2	0.0	2.7	85.0	95.7	96.3
$D(T,R)$		$p=0.3$	183	0.1	50.2	0.0	2.7	85.0	95.7	96.3
$D(T,T)$	$r_{\text{thr}}=0.01$	$p=0.3$	913	0.7	16.1	1.0	0.0	27.2	62.6	64.3
$D(T,R)$		$p=0.3$	183	0.7	16.1	1.0	0.0	27.2	62.6	64.3
GMDD on $\mathbf{D^P_1}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$p=0.3$	913	0.1	56.5	0.0	6.4	93.3	98.3	98.5
$D(T,R)$		$p=0.3$	183	0.1	56.5	0.0	6.4	93.3	98.3	98.5
$D(T,T)$	$r_{\text{thr}}=0.01$	$p=0.3$	913	0.7	28.2	0.8	0.1	48.6	75.5	79.2
$D(T,R)$		$p=0.3$	183	0.7	28.2	0.8	0.1	48.6	75.5	79.2

11.7% and 9.3%, respectively. The error on the normal class is 1.5%. Such results are obtained by using 17 support objects only. The best LPDD (keeping target error small), based on 14 support objects yields an error of 1.3% for the target class and errors of 15.8% and 13.6% for the above mentioned outlier subclasses. Such performances are only somewhat worse than the results for the best 1-NNDD, while they are based on dissimilarities to less than 2% of all training objects.

(4) Since our experiments are done in the same way as in [53], our results can be compared. In [53], a sparse linear programming formulation has been proposed (from which our the LPDD-II method is derived) for Gaussian kernels. The results on the test set reported there (and also recreated by us [306]) are 1.3% for NB, 0% for A_1, 46.7% for A_2, 71.7% for A_3 and 74.5% for A_4, which are very bad in comparison to our LPDD results on a sigmoidal-I transformation of $D_{0.8}$ of D_1. We think that this is mainly caused be the use of the Euclidean distance (the Gaussian kernel is based on it). This is supported by the facts that our LPDDs perform also badly on D_2 and when a radial basis function is defined on the city block distance (d_1), better results are obtained for the method in [53]; see [306].

Table 8.2: The errors of the first and second kind (in %) of the OCCs trained on the $l_{0.8}$ distance representation $D_{0.8}(T,R)$ for the ball-bearing data. T is the target class (normal behavior) consisting of 913 samples. R is a subset of T (reduced set) consisting of 183 (20%) examples from T chosen by the k-centers algorithm with $k=183$. R_e is the effective set of objects on which the constructed OCCs rely. The optimal parameter s of the sigmoidal transformations or the optimal k for the k-CDD and k-NNDD are selected based on the performance on the validation set.

OCC built on	ν or r_{thr}	Optimal s or k	$\lvert R_e\rvert$	Validation errors [%]		Test errors [%]				
				NB	A_1+A_2	NB	A_1	A_2	A_3	A_4
LPDD on a sigmoidal-I transformation of $\mathbf{D_{0.8}}$										
$D(T,T)$	$\nu=0.00$	$s=7.89$	17	0.9	0.6	1.3	0.0	0.9	18.2	17.0
$D(T,R)$		$s=8.86$	8	0.7	0.7	1.2	0.0	1.0	16.1	14.4
$D(T,T)$	$\nu=0.01$	$s=9.02$	12	0.9	0.7	1.5	0.0	1.1	19.5	17.6
$D(T,R)$		$s=10.86$	10	0.8	0.6	1.2	0.0	1.2	19.5	16.9
LPDD on a sigmoidal-II transformation of $\mathbf{D_{0.8}}$										
$D(T,T)$	$\nu=0.00$	$s=11.27$	16	0.8	0.8	1.2	0.0	0.9	19.0	17.6
$D(T,R)$		$s=12.86$	11	0.5	0.6	1.3	0.0	1.0	17.6	16.3
$D(T,T)$	$\nu=0.01$	$s=12.40$	13	0.8	0.9	1.2	0.0	1.2	20.2	18.9
$D(T,R)$		$s=10.00$	14	0.9	0.8	1.3	0.0	1.0	**15.8**	**13.6**
LPDD-II on a sigmoidal-I transformation of $\mathbf{D_{0.8}}$										
$D(T,T)$	$\nu=0.00$	$s=22.26$	14	1.1	0.3	1.6	0.0	0.7	17.1	14.0
$D(T,R)$		$s=19.44$	1	0.0	53.2	0.0	1.5	91.4	96.0	97.6
$D(T,T)$	$\nu=0.01$	$s=13.52$	17	1.4	0.3	1.5	0.0	0.5	**11.7**	**9.3**
$D(T,R)$		$s=19.44$	1	1.1	48.1	1.1	0.4	83.4	93.2	94.4
LPDD-II on a sigmoidal-II transformation of $\mathbf{D_{0.8}}$										
$D(T,T)$	$\nu=0.00$	$s=39.45$	6	0.5	1.7	0.2	0.0	2.2	27.1	26.8
$D(T,R)$		$s=20.35$	1	0.0	53.2	0.0	1.5	91.4	96.0	97.6
$D(T,T)$	$\nu=0.01$	$s=38.32$	8	1.2	1.1	1.4	0.0	2.0	23.6	23.3
$D(T,R)$		$s=20.35$	1	1.1	48.1	1.0	0.4	83.8	93.3	94.4
GMDD on a sigmoidal-I transformation of $\mathbf{D_{0.8}}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$s=2.25$	913	0.7	11.9	0.8	0.0	21.4	55.7	58.9
$D(T,R)$		$s=2.29$	183	0.7	13.3	0.8	0.0	23.0	57.5	61.2
$D(T,T)$	$r_{\text{thr}}=0.01$	$s=2.25$	913	1.4	4.4	1.4	0.0	7.6	40.7	42.7
$D(T,R)$		$s=2.29$	183	1.4	4.5	1.4	0.0	7.8	41.5	43.3
GMDD on a sigmoidal-II transformation of $\mathbf{D_{0.8}}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$s=4.51$	913	2.3	0.8	2.2	0.0	0.9	**19.3**	**18.6**
$D(T,R)$		$s=4.86$	183	1.4	1.9	1.4	0.0	2.6	27.2	28.6
$D(T,T)$	$r_{\text{thr}}=0.01$	$s=5.07$	913	1.5	2.0	1.8	0.0	3.2	28.8	29.7
$D(T,R)$		$s=4.57$	183	2.4	0.8	2.2	0.0	0.9	19.4	18.7
k-CDD on $\mathbf{D_{0.8}}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$k=38$	38	0.4	5.1	0.5	0.0	9.5	41.9	41.8
$D(T,R)$		$k=6$	6	0.1	4.6	0.2	0.0	8.8	39.7	39.3
$D(T,T)$	$r_{\text{thr}}=0.01$	$k=47$	47	1.3	0.7	1.5	0.0	1.5	**22.9**	**22.1**
$D(T,R)$		$k=42$	42	1.1	1.7	1.4	0.0	3.3	26.8	27.9
k-NNDD on $\mathbf{D_{0.8}}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$k=6$	913	0.0	42.3	0.0	0.9	71.4	90.8	91.7
$D(T,R)$		$k=1$	183	0.0	42.6	0.0	0.7	72.9	90.5	91.9
$D(T,T)$	$r_{\text{thr}}=0.01$	$k=1$	913	0.2	2.3	0.1	0.0	3.4	29.8	31.5
$D(T,R)$		$k=2$	183	0.4	6.7	0.8	0.0	11.3	46.1	46.7
$D(T,T)$	$r_{\text{thr}}=0.05$	$k=1$	913	1.2	0.3	1.4	0.0	0.0	**9.9**	**8.3**
$D(T,R)$		$k=1$	183	2.8	0.5	2.6	0.0	0.5	15.0	13.9

Table 8.3: The errors of the first and second kind (in %) of the OCCs trained on the l_1 distance representation $D_1(T,R)$ for the ball-bearing data. T is the target class (normal behavior) consisting of 913 samples. R is a subset of T (reduced set) consisting of 183 (20%) examples from T chosen by the k-centers algorithm with $k=183$. R_e is the effective set of objects on which the constructed OCCs rely. The optimal parameter s of the sigmoidal transformations or the optimal k for the k-CDD and k-NNDD are selected based on the performance on the validation set.

OCC built on	ν or r_{thr}	Optimal s or k	$\lvert R_e\rvert$	Validation errors [%]		Test errors [%]				
				NB	A_1+A_2	NB	A_1	A_2	A_3	A_4
LPDD on a sigmoidal-I transformation of $\mathbf{D_1}$										
$D(T,T)$	$\nu=0.00$	$s=5.53$	13	1.1	1.1	1.3	0.0	1.6	20.9	20.0
$D(T,R)$		$s=6.28$	9	0.7	0.9	1.1	0.0	1.5	21.0	20.5
$D(T,T)$	$\nu=0.01$	$s=7.37$	10	1.1	1.2	1.5	0.0	2.5	22.5	22.5
$D(T,R)$		$s=1.57$	20	2.0	0.5	2.1	0.0	0.7	**15.4**	**13.2**
LPDD on a sigmoidal-II transformation of $\mathbf{D_1}$										
$D(T,T)$	$\nu=0.00$	$s=6.24$	12	0.9	1.3	1.0	0.0	1.8	23.3	22.2
$D(T,R)$		$s=6.57$	10	0.8	1.1	1.0	0.0	1.6	21.3	20.2
$D(T,T)$	$\nu=0.01$	$s=6.09$	11	1.2	1.2	1.5	0.0	1.7	21.8	21.3
$D(T,R)$		$s=6.86$	7	1.1	1.8	1.3	0.0	2.3	26.7	27.2
LPDD-II on a sigmoidal-I transformation of $\mathbf{D_1}$										
$D(T,T)$	$\nu=0.00$	$s=13.61$	14	1.2	0.6	1.8	0.0	0.9	18.6	17.3
$D(T,R)$		$s=9.71$	1	0.0	57.1	0.0	4.1	97.1	99.1	98.9
$D(T,T)$	$\nu=0.01$	$s=10.35$	15	1.6	0.2	1.6	0.0	0.6	**15.2**	**13.5**
$D(T,R)$		$s=9.71$	1	1.1	53.9	1.3	1.7	92.2	96.8	97.6
LPDD-II on a sigmoidal-II transformation of $\mathbf{D_1}$										
$D(T,T)$	$\nu=0.00$	$s=10.09$	8	0.8	1.5	1.0	0.0	2.4	28.7	27.5
$D(T,R)$		$s=10.28$	1	0.0	57.1	0.0	4.1	97.1	99.1	98.9
$D(T,T)$	$\nu=0.01$	$s=9.07$	6	0.5	1.8	1.0	0.0	3.2	21.3	18.5
$D(T,R)$		$s=5.71$	1	1.1	53.8	1.4	1.7	91.9	96.7	97.2
GMDD on a sigmoidal-I transformation of $\mathbf{D_1}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$s=1.13$	913	0.7	20.4	0.9	0.0	36.3	67.5	71.6
$D(T,R)$		$s=1.14$	183	0.7	21.0	0.9	0.0	37.1	68.3	72.5
$D(T,T)$	$r_{\text{thr}}=0.01$	$s=1.13$	913	1.1	12.9	1.3	0.0	22.2	56.2	58.7
$D(T,R)$		$s=1.14$	183	1.1	13.1	1.2	0.0	23.0	56.7	59.4
GMDD on a sigmoidal-II transformation of $\mathbf{D_1}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$s=2.13$	913	2.8	1.3	2.8	0.0	1.5	23.0	22.7
$D(T,R)$		$s=2.14$	183	2.8	1.4	2.6	0.0	1.8	23.5	23.7
$D(T,T)$	$r_{\text{thr}}=0.01$	$s=2.13$	913	3.2	1.3	3.3	0.0	1.2	**22.1**	**21.6**
$D(T,R)$		$s=2.14$	183	2.8	1.3	3.0	0.0	1.5	23.0	22.8
k-CDD on $\mathbf{D_1}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$k=42$	42	0.8	6.7	0.4	0.0	11.2	48.0	49.2
$D(T,R)$		$k=45$	45	0.1	16.5	0.2	0.0	28.3	61.8	65.2
$D(T,T)$	$r_{\text{thr}}=0.01$	$k=47$	47	1.3	2.8	1.1	0.0	4.8	**32.6**	**34.6**
$D(T,R)$		$k=42$	42	0.7	4.2	1.6	0.0	7.5	38.2	39.9
k-NNDD on $\mathbf{D_1}$										
$D(T,T)$	$r_{\text{thr}}=0.00$	$k=1$	913	0.0	47.8	0.0	1.3	81.2	94.8	94.7
$D(T,R)$		$k=2$	183	0.0	48.3	0.0	1.6	81.8	94.9	95.4
$D(T,T)$	$r_{\text{thr}}=0.01$	$k=2$	913	0.7	4.6	0.5	0.0	6.6	39.4	42.4
$D(T,R)$		$k=1$	183	0.3	11.0	0.3	0.0	19.7	54.4	56.5
$D(T,T)$	$r_{\text{thr}}=0.05$	$k=1$	913	1.9	0.4	1.6	0.0	0.0	**12.1**	**9.8**
$D(T,R)$		$k=1$	183	2.6	0.9	2.7	0.0	1.5	20.9	20.1

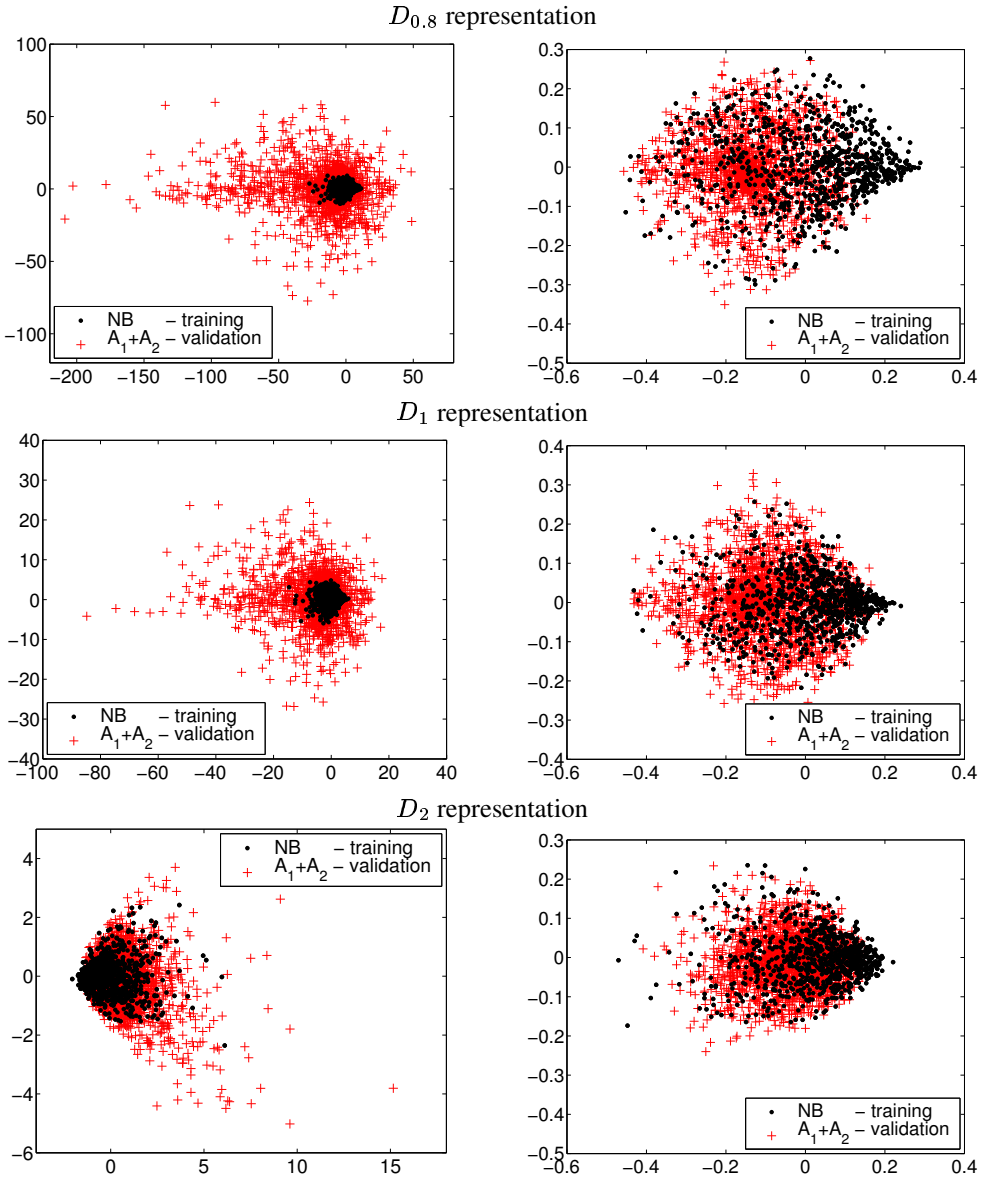


Fig. 8.14: 2D linear approximate embeddings of the dissimilarity representations (left) or their sigmoidal-I transformations (right). The embedding spaces are defined on the training target objects $D(R,R)$. The outliers from the validation sets are then projected to such spaces. Note the scale differences.

(5) The LPDD may benefit from a representation set R smaller than the training set T. This can especially be observed for sigmoidal transformations of $D_{0.8}$, upper rows in Table 8.2, where the test errors about the same or smaller than in case of a complete dissimilarity representations. On the contrary, the LPDD-II determines only one support object and its performance deteriorates to about 90% error on the outlier class. A smaller R seems to be disadvantageous for other OCCs, as well.

(6) The GMDD and k-CDD do not perform well.

To better explain differences in the OCCs, we will discuss the data characteristics. As explained in chapter 6, to understand the data, one may visualize their dissimilarity relations. Here, we simply apply the pseudo-Euclidean embedding to each of the dissimilarity representations $D_{0.8}(T,T)$, $D_1(T,T)$ and $D_2(T,T)$ (T is the training target class) as well as their best (in terms of the performance) sigmoidal-I transformations. The mappings are defined on the NB class and then the remaining outliers from the validation set are added after that. The projections to 2D are shown in Fig. 8.14. Remember that these linear projections preserve the variance as much as possible as

revealed in two dimensions. The preserved variances are equal to 19.3%, 26.1% and 70.6% for the $D_{0.8}(T,T)$, $D_1(T,T)$ and $D_2(T,T)$, respectively. For the sigmoidal transformations they are somewhat smaller. Although the preserved variance is only large in case of the Euclidean representation (70.6%), the first two eigenvalues of the embedding (which correspond to the variances) are significantly the larges for all the cases. See e.g. Fig. 8.13, where the eigenvalues for $D_{0.8}(T,T)$ are shown. Note that in case of $D_2(T,T)$, the projection is equivalent to the PCA projection applied to data instances represented by their pre-processed 32 attributes in $\mathbb{R}^{32}$; see section 3.3.1.

Although the 2D projections of the dissimilarity data only roughly approximate the actual relations, still, they allow to build some intuition. Analyzing the left plots in Fig. 8.14, one can immediately see that for the $l_{0.8}$ and l_1 distances, the target data (NB class) is a rather compact cloud. The outliers are widely spread in-between and around the target class. So, the target class seems to lie among the outliers (and the overlap for the target class is very high). Judging visually, the ratio of the area of the target cloud to the outlier cloud is smaller for the $D_{0.8}$ than for D_1. This simply suggests that the $l_{0.8}$ distance offers better discrimination between the targets and outliers. On the contrary, in case of the Euclidean representation, the target cloud is very large in comparison to the outlier cloud. Hence, many outliers will be incorrectly assigned.

Analyzing the right plots in Fig. 8.14, one can observe that they change the sizes of the target and outlier clouds and they also shift their positions with respect to each other when comparing the left plots. As a result, some parts are non-overlapping. Hence, possibly better OCCs can be built. Also bad performances of the GMDD and the k-CDD can now be somewhat understood. The NB class seems to be a relatively compact cloud. Since the k-CDD builds a bubble-like description around the k centers (see Fig. 8.2), it will not be beneficial for a single bulk. Since the GMDD still builds a relatively wide boundary around the data points and hence its flexibility is limited (due to the fact that it relies on the mean of the target class in a pseudo-Euclidean space; see Fig. 8.3), it will not be advantageous in case of a high overlap between the targets and outliers. So, the only flexible OCCs which allow for building tights boundaries are of use. Hence, the good performance of the LP classifiers and the k-NNDD.

8.3.2 Experiment II: Diseased mucosa in the oral cavity

In this experiment, we will analyze the autofluorescence spectra acquired from healthy and diseased mucosa in the oral cavity; see section A for the data description. The measurements were taken at 11 different anatomical locations using six different excitation wavelengths. We will focus on a single excitation wavelength of 365 nm. After preprocessing [406], each spectrum consists of 199 bins. In total, 856 and 132 spectra representing healthy and diseased tissue, respectively, were obtained for each excitation wavelength. The spectra are normalized so that they yield a unit area.

In our study, all the spectra are 50 times randomly split into the training set T and the test set T_{te} in the ratio of $60:40$, respectively. The training set consist of both target and outlier examples, while the representation set $R \subset T$ contains only targets. Hence, $|R| = 514$, $|T| = 594$ and $|T_{te}| = 394$ ($337/57$ healthy/diseased patients). OCCs relying on the target information only are trained on $D(R,R)$. If the outlier information can be incorporated, then the OCCs are trained in $D(T,R)$. In a testing stage, $D(T_{te},R)$ is used in both cases.

Eight dissimilarity representations have been considered for the normalized spectra. The first three dissimilarity representations (DRs) are based on the l_1 distances computed between the Gaussian-smoothed spectra ($\sigma = 3$ samples) themselves (D_1) and their first and the second order Gaussian-smoothed ($\sigma = 3$ samples) derivatives (D_1^{der} and D_1^{2der}, respectively). The zero-crossings of the derivatives indicate the peaks and valleys of the spectra, so they are informative. The differences between the spectra focus on the overlap, the differences in first derivatives emphasize the loca-

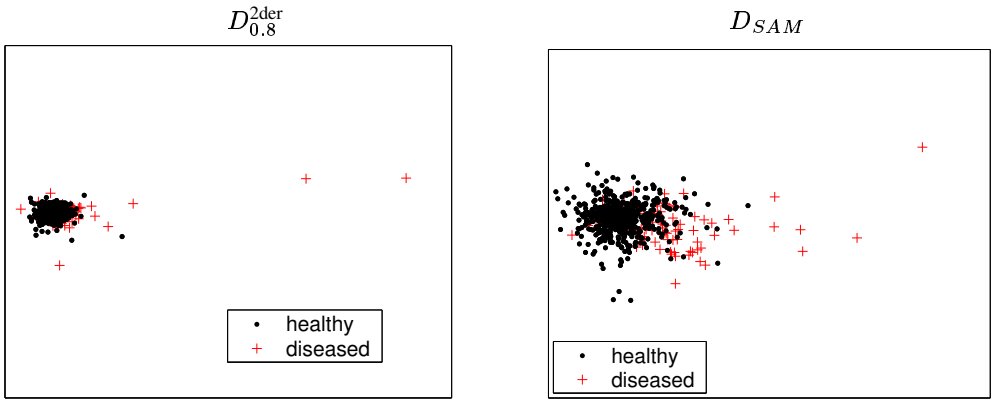


Fig. 8.15: The 2D approximate embeddings of two dissimilarity representations $D_{0.8}^{2der}$ (left) and D_{SAM} (right) between the autofluoresence spectra. The embedding spaces are defined on the (training) target objects $D(R,R)$. The outliers $T \backslash R$ are then projected to such spaces.

tions of peaks and valleys, while the differences in second derivatives indicate the tempo of changes in spectra. Also $l_{0.8}$ non-metric distances have been used, again between the spectra and their Gaussian-smoothed derivatives, yielding the representations $D_{0.8}, D_{0.8}^{der}$ and $D_{0.8}^{2der}$, correspondingly. Since spectra posses a natural connectivity given by the order of the sampled wavelengths, the dissimilarity measures which make use of that fact might be beneficial. Derivative-based dissimilarity measures take such information into account. Next representation D_{SAM} is based on the spherical geodesic distance $d_{SAM}(\mathbf{x},\mathbf{y}) = 1 \arccos(\mathbf{x}^T\mathbf{y})/r^2$ (here $r=1$), which is actually the spectral angular mapper distance [239]; see also section 3.3.8. The remaining representation relies on a Bhattacharyya distance, a divergence measure between two probability distributions; see section 5.2. This measure is applicable, since the normalized spectra, say, s_i can be considered as unidimensional histogram-like distributions. They are constant on disjoint intervals $I_1, \ldots, I_N$, such that $s_i(x) = \sum_{z=1}^{N} h_z^i \mathcal{I}(x \in I_z)$, where h_z^i are nonnegative and $\mu(I_z)$ is the length of I_z. The Bhattacharyya distance [118] is then: $d_{BH}(s_i,s_j) = -\log\left(\sum_{z=1}^{N}(h_z^i h_z^j)^{1/2}\right)\mu(I_z)$. So, all the dissimilarity representations emphasize different aspects of the spectra.

The AUC performances for the LPDD, the LPDD-II, the GMDD, the k-CDD and the k-NNDD for six DRs are presented in Table 8.4. Since in the one-class classification, there is always a trade-off between the false positive and false negative ratios and we just want to compare the methods, the AUC measures seem to be appropriate. Otherwise, we need to fix a point of comparison, which is subjective. The AUC performance gives us an overall measure; see also section 8.1.

The LPDD, the LPDD-II and the GMDD rely also on power and sigmoidal transformations of the DRs while the k-NNDD and k-CDD are built on the original dissimilarities directly. We do not present the results of the LPDD-II for sigmoidal-II transformations, since the corresponding linear programming problems did not terminated successfully (they were infeasible).

Since the $l_{0.8}$ distance representations are somewhat more discriminative than the l_1 distance representations, for the derivative-based measures, the results for the later are omitted. The following conclusions can be made by analyzing Table 8.4:

1. Concerning various dissimilarity measures, the most discriminative is the $l_{0.8}$ distance between the smooth second order derivatives of the spectra ($D_{0.8}^{2der}$) and the less discriminative one is the geodesic spherical distance (D_{geo}). The probabilistic Bhattacharyya distance is also good.
2. The LPDD yields better results than the LPDD-II.
3. The use of outlier information is beneficial for the LP classifiers. Both the LPDD and the LPDD-II perform significantly better than when trained on the target examples only. Yet,

Table 8.4: AUC measure ($\cdot 100$) of the OCCs trained on various dissimilarity representations (and their transformations) for the oral cavity problem. The OCCs are trained either on $D(R,R)$ or on $D(T,R)$ (in case of the LP classifiers), where T is a training set of both target and outlier examples and $R \subset T$ consists of the targets only; $|R| = 514$ and $|T| = 594$. In parenthesis, either the average number of support objects for the LP classifiers is shown or $|R_e|$, the effective number of objects from R determining the boundary of the OCCs. (The GMDD is based on 10 objects determined by the mean-resemblance approach as explained in section 8.2.2.) The results are averaged over 50 runs. The standard deviations of the AUC$\cdot$100 means are less than 1.4, (for the LPDD trained on $D(T,R)$), while in the majority of cases, they are less than 0.6. The best results are printed in bold for each classifier and each dissimilarity representation.

Transformation	$\mathbf{D_1}$	$\mathbf{D_{0.8}}$	$\mathbf{D_{0.8}^{der}}$	$\mathbf{D_{0.8}^{2der}}$	$\mathbf{D_{SAM}}$	$\mathbf{D_{BH}}$
LPDD, $\nu = 0.05$, trained on $\mathbf{D(R,R)}$						
Original	72.3 (2.4)	73.5 (3.2)	72.5 (2.9)	79.4 (3.2)	68.1 (2.8)	74.4 (2.1)
Power; $p = 0.5$	72.7 (7.3)	73.1 (7.7)	72.6 (3.2)	79.7 (3.4)	68.5 (5.0)	74.7 (2.6)
Sigm-I; $s = d_{avr}$	73.7 (6.8)	73.8 (7.9)	73.0 (4.1)	79.9 (3.6)	72.5 (6.2)	**79.3** (5.0)
Sigm-II; $s = d_{avr}$	**75.9** (7.6)	**76.4** (8.0)	**74.4** (7.0)	**80.4** (4.8)	**74.5** (7.5)	77.8 (6.9)
LPDD, $\nu = 0.05$, trained on $\mathbf{D(T,R)}$; outliers used						
Original	79.9 (5.5)	80.1 (5.8)	82.8 (6.1)	84.9 (5.0)	80.0 (6.0)	79.5 (2.7)
Power; $p = 0.5$	82.5 (20.9)	82.6 (21.7)	84.5 (12.6)	86.1 (9.5)	80.1 (25.3)	83.4 (5.1)
Sigm-I; $s = d_{avr}$	**82.6**(15.8)	**82.7**(16.5)	84.7 (11.2)	**86.4** (8.7)	**82.0**(15.9)	**84.5** (7.7)
Sigm-II; $s = d_{avr}$	82.2 (13.3)	82.3 (15.0)	**86.0**(15.8)	86.4 (10.8)	81.7 (11.6)	82.9 (12.0)
LPDD-II, $\nu = 0.05$, trained on $\mathbf{D(R,R)}$						
Original	**66.4** (3.5)	**55.5** (3.2)	63.4 (5.9)	**76.5** (6.8)	58.8 (4.9)	**74.2** (3.8)
Power; $p = 0.5$	53.0 (11.0)	48.0 (13.1)	**68.6**(12.1)	76.2 (13.2)	**61.9**(12.3)	70.0 (4.8)
Sigm-I; $s = d_{avr}$	52.0 (15.9)	51.6 (15.9)	59.1 (9.4)	56.5 (10.0)	55.5 (13.2)	51.4 (10.1)
LPDD-II, $\nu = 0.05$, trained on $\mathbf{D(T,R)}$; outliers used						
Original	76.0 (3.4)	66.0 (3.6)	78.8 (6.0)	**81.7** (5.2)	74.6 (4.0)	81.2 (3.2)
Power; $p = 0.5$	**78.3**(11.1)	**78.2**(14.9)	**79.8** (8.5)	81.6 (8.2)	76.5 (7.8)	**81.6** (4.1)
Sigm-I; $s = d_{avr}$	76.8 (14.0)	76.8 (15.2)	78.1 (8.6)	78.1 (8.7)	75.5 (11.9)	77.3 (9.9)
GMDD, $\mathbf{r_{thr}} = 0.05$, trained on $\mathbf{D(R,R)}$						
Original	77.0 (10.0)	77.0 (10.0)	78.8 (10.0)	79.5 (10.0)	76.7 (10.0)	77.2 (10.0)
Power; $p = 0.5$	78.0 (10.0)	78.2 (10.0)	**79.5**(10.0)	79.7 (10.0)	77.6 (10.0)	78.9 (10.0)
Sigm-I; $s = d_{avr}$	**78.1**(10.0)	78.4 (10.0)	79.4 (10.0)	**79.8**(10.0)	**77.8**(10.0)	**79.2**(10.0)
Sigm-II; $s = d_{avr}$	77.9 (10.0)	**78.7**(10.0)	78.7 (10.0)	79.2 (10.0)	**77.8**(10.0)	79.0 (10.0)
k-CDD, $\mathbf{r_{thr}} = 0.05$, trained on $\mathbf{D(R,R)}$						
$k = 1$	61.6 (1.0)	58.7 (1.0)	48.0 (1.0)	53.7 (1.0)	50.9 (1.0)	72.5 (1.0)
$k = 5$	65.5 (5.0)	66.0 (5.0)	75.2 (5.0)	76.7 (5.0)	65.7 (5.0)	73.2 (5.0)
$k = 11$	74.4 (11.0)	74.6 (11.0)	76.1 (11.0)	78.7 (11.0)	73.6 (11.0)	76.9 (11.0)
$k = 21$	76.2 (21.0)	76.8 (21.0)	78.7 (21.0)	81.1 (21.0)	76.3 (21.0)	81.0 (2.0)
$k = 41$	77.5 (41.0)	78.1 (41.0)	82.0 (41.0)	83.2 (41.0)	**78.0**(41.0)	82.9 (41.0)
$k = 61$	**78.7**(61.0)	**78.7**(61.0)	**83.0**(61.0)	**84.3**(61.0)	77.9 (61.0)	**83.1**(61.0)
k-NNDD, $\mathbf{r_{thr}} = 0.05$, trained on $\mathbf{D(R,R)}$						
$k = 1$	73.4 (10.0)	74.5 (10.0)	76.4 (10.0)	78.6 (10.0)	73.2 (10.0)	77.0 (10.0)
$k = 3$	77.9 (10.0)	78.6 (10.0)	79.4 (10.0)	79.4 (10.0)	79.0 (10.0)	80.1 (10.0)
$k = 5$	78.3 (10.0)	79.4 (10.0)	79.9 (10.0)	79.7 (10.0)	78.5 (10.0)	79.7 (10.0)
$k = 1$	76.6 (26.0)	77.6 (26.0)	79.7 (26.0)	**81.9**(26.0)	77.5 (26.0)	81.7 (26.0)
$k = 3$	**79.7**(26.0)	**80.3**(26.0)	**80.4**(26.0)	**81.9**(26.0)	**80.0**(26.0)	**82.6**(26.0)
$k = 5$	**79.7**(26.0)	80.2 (26.0)	**80.4**(26.0)	81.4 (26.0)	79.9 (26.0)	82.1 (26.0)
k-NNDD, $\mathbf{r_{thr}} = 0.05$, trained on $\mathbf{D(R,R)}$; $\mathbf{\lvert R_e \rvert = 514}$						
$k = 1$	80.0	80.4	**86.7**	**88.2**	80.4	**85.7**
$k = 5$	**81.0**	**81.4**	85.7	86.2	**81.1**	84.9
$k = 11$	80.9	81.3	84.8	85.2	80.8	84.3
$k = 21$	80.7	81.1	84.0	84.4	80.4	83.6
$k = 41$	80.1	80.6	83.2	83.6	80.0	82.8
$k = 61$	79.9	80.3	82.8	83.0	79.8	82.3

Table 8.5: AUC measure (in $\cdot 100$) of the OCCs for the Gower distance representation and its transformations describing the heart disease problem. The OCCs are trained either on $D(R,R)$ or on $D(T,R)$ (in case of the LP classifiers), where T is a training set of both target and outlier examples and $R \subset T$ consists of the targets only. $|R| = 84$ and $|T| = 183$. In parenthesis, either the average number of support objects for the LP classifiers or the effective number of objects from R determining the boundary of the OCCs are shown. The results are averaged over 50 runs. The standard deviations of the AUC$\cdot 100$ means are less than 0.5.

LPDD, $\nu = 0.05$, trained on				
	$\mathbf{D(R,R)}$		$\mathbf{D(T,R)}$	
Original	82.3	(49.5)	85.3	(25.3)
Power; $p = 0.5$	83.9	(62.6)	**85.6**	(44.6)
Power; $p = 0.3$	**85.1**	(70.4)	85.2	(69.2)
Sigm-I; $s = 0.5 d_{avr}$	83.6	(61.3)	85.5	(42.7)
Sigm-I; $s = d_{avr}$	82.6	(53.0)	85.4	(29.2)
Sigm-II; $s = 0.5 d_{avr}$	84.4	(70.4)	84.6	(68.3)
Sigm-II; $s = d_{avr}$	81.7	(37.5)	84.9	(16.9)
LPDD-II, $\nu = 0.05$, trained on				
	$\mathbf{D(R,R)}$		$\mathbf{D(T,R)}$	
Original	86.7	(34.4)	81.4	(53.6)
Power; $p = 0.5$	87.6	(45.6)	82.4	(72.8)
Power; $p = 0.3$	**87.8**	(57.1)	83.4	(80.9)
Sigm-I; $s = 0.5 d_{avr}$	87.5	(45.4)	82.2	(71.1)
Sigm-I; $s = d_{avr}$	87.0	(37.3)	81.5	(58.5)
Sigm-II; $s = 0.5 d_{avr}$	84.4	(70.4)	84.6	(68.3)
Sigm-II; $s = d_{avr}$	81.7	(37.5)	**84.9**	(16.9)
GMDD, $\mathbf{r_{thr}} = 0.05$, trained on $\mathbf{D(R,R)}$				
Original	85.7	(8.0)	85.9	(13.0)
Power; $p = 0.5$	85.7	(8.0)	**86.2**	(13.0)
Power; $p = 0.3$	85.4	(8.0)	85.8	(13.0)
Sigm-I; $s = 0.5 d_{avr}$	85.6	(8.0)	85.9	(13.0)
Sigm-I; $s = d_{avr}$	**86.0**	(8.0)	85.8	(13.0)
Sigm-II; $s = 0.5 d_{avr}$	84.8	(8.0)	85.5	(13.0)
Sigm-II; $s = d_{avr}$	85.5	(8.0)	85.8	(13.0)
k-NNDD, $\mathbf{r_{thr}} = 0.1$, trained on $\mathbf{D(R,R)}$				
$k = 1$	74.8	(8.0)	75.0	(13.0)
$k = 3$	80.5	(8.0)	80.1	(13.0)
$k = 5$	82.3	(8.0)	82.2	(13.0)
$k = 7$	**82.8**	(8.0)	**83.5**	(13.0)
$\mathbf{r_{thr}} = 0.1$	**k-CDD**		**k-NNDD**	
$k = 1$	**82.2**	(1.0)	76.1	(84.0)
$k = 5$	75.4	(5.0)	81.0	(84.0)
$k = 8$	74.5	(8.0)	82.3	(84.0)
$k = 13$	74.2	(13.0)	83.6	(84.0)
$k = 21$	74.7	(21.0)	84.8	(84.0)
$k = 31$	74.7	(31.0)	85.6	(84.0)
$k = 51$	75.2	(51.0)	**86.3**	(84.0)

they need more support objects for this.

4. Here, we only present the GMDD based on 10 objects determined by the mean-resemblance (see section 8.2.2), since they are nearly the same as the results obtained by the GMDD trained on the complete set R (514 objects). This suggests that the target class is rather compact. Although the GMDD is not a flexible classifier, its performance is better or the same as of the LP classifiers trained on the target class. When compared to 11-CDD (hence based on a similar number of objects), it reaches a better performance. However, when a larger k is used, the k-CDD gives better results.
5. The best result for OCCs trained on the target class is obtained for the 1-NNDD on $D_{0.8}^{2\text{der}}$

(AUC is 88.2). When outlier information is used, then the LPDD behaves similarly well, however, additionally, it allows for a significant reduction in computation. In the best case, the LPDD selects at most 44 support objects (out of 514 in the set R) for the sigmoidal-I transformation of the $D^{2\text{der}}_{0.8}$ reaching the AUC of 88.9 or 13 support objects for the power transformation of the same DR, reaching the AUC of 88.4. The 1-NNDD relies on dissimilarities to all 514 objects.

8.3.3 Experiment III: Heart disease data

In this experiment, we will analyze the heart disease data, which provide information on ill and healthy patients. The goal is to detect the presence of a heart disease. There are 303 instances, where 139 correspond to healthy patients. Since the data consist of mixed types: continuous, dichotomous and categorical variables, a Gower's dissimilarity, as defined in section 5.1, can be computed.

In our study, all the data are 50 times randomly split into the training set T and the test set T_{te} in the ratio of $60:40$, respectively. The training set consists of both target and outlier examples, while the representation set $R \subset T$ contains only the target examples. $|R| = 84$, $|T| = 183$ and $|T_{te}| = 120$ (55 healthy patients and 64 diseased patients). The OCCs relying on the target information only are trained on $D(R, R)$. If the information on outliers can be incorporated to the classifier in the learning stage, the OCCs are trained on $D(T, R)$. In a testing stage, $D(T_{te}, R)$ is used in both cases. The results are shown in Table 8.5. Since there are more diseased patients than the healthy ones, we have also tried to design an OCC by assuming that the ill patients form the target class. However, the results have become worse, so they are not presented here.

The problem is difficult, since the target class cannot be easily distinguished from the outliers, which suggests that the measurements, based on which the Gower dissimilarity is derived, do not have enough discriminative power. Such a conclusion can be drawn because of the following facts:

- The LP classifiers need many support objects, on average around 60% of the target objects.
- The LPDD-II decreases its performance when it is trained using the outlier information as well. This suggests a high overlap of the target and outlier classes.
- The k-NNDD improves its performance with growing k, while k-CDD not.
- The GMDD outperforms the k-NNDD (which is often a very good classifier).

This all suggests that the target class can be described by one cloud (1-CDD is the best), but the outliers lie 'in-between' the targets. It seems that in such a case, the GMDD may perform relatively well. When defined on a reduced set R_e of eight or 13 objects, it performs comparably to the best LPDD-II.

8.4 Conclusions

This chapter is devoted to one-class classifications problems. Such problems are identified in many real applications, such as health diagnostics, machine condition monitoring or industrial inspection. Given training examples, the goal is to describe the target class such that resembling objects are accepted as targets and outliers (non-targets) are rejected. Such a detection has to be performed in an unknown or ill-defined context of alternative phenomena. The target class is assumed to be well sampled and well defined. The alternative outlier set is usually ill-defined: it is badly sampled (even not present at all) with unknown and hard to predict priors. Since the non-target class is ill-defined, in complex problems, an effective set of features discriminating between targets and outliers cannot be easily found. Hence, it seems appropriate to build a representation on the raw data. The dissimilarity representation, describing objects by their dissimilarities to the target

examples, may be effective for such problems since it naturally protects the target class against unseen novel examples.

For such types of problems one-class classifiers (OCCs) designed as boundary descriptors might be suitable. Here, examples of three types of OCCs built on dissimilarity representations are considered: neighborhood-based, in an embedded space and in a dissimilarity space. We proposed two classifiers: the GMDD, a simple OCC in an embedded space defined by the distance to the mean and the LPDD and the LPDD-II, OCCs defined as hyperplanes in a dissimilarity space. In both cases, sparse solutions can be obtained, meaning that the OCCs are ultimately based on a relatively small number of target objects. For the LPDD, they are detected automatically by solving the linear programming formulation, while for the GMDD they can be forced by choosing a specified fraction of objects to represent the target class. These OCCs are compared to the neighborhood-based OCCs: the k-CDD and k-NNDD.

Three different problems are analyzed here: machine monitoring, lesion diagnostics and heart disease diagnostics. The following conclusions can be made. When the outliers do not heavily overlap with the target objects, and some outliers are used for training, the LPDDs provide the best solutions as a trade-off between the performance and the computational aspect (the effective number of target objects which define the boundary). The k-NNDD, based on the average dissimilarity to the k-nearest neighbors, is a good classifier and may outperform the LPDDs, yet, it requires many more target objects for the good definition of its boundary. When there is a high overlap between the targets and outliers, the GMDD, as a weak classifier, may become good, since it relies on global information, i.e. the distance to the mean (in an embedded space), instead of the local information, as e.g. the k-NNDD does.

Concerning dissimilarity measures, the best measures in machine monitoring and lesion diagnostics are non-metric $l_{0.8}$ distances between some intermediate representations. This is an interesting point, since it supports our idea that non-metric dissimilarities can be beneficial for learning, which is currently partly neglected to think in this direction.

9. Classification

Inanimate objects are always correct and cannot, unfortunately, be reproached with anything. I have never observed a chair shift from one foot to another, or a bed rear on its hind legs. And tables, even when they are tired, will not dare to bend their knees. I suspect that objects do this from pedagogical considerations, to reprove us constantly for our instability.

"OBJECTS", ZBIGNIEW HERBERT

The challenge of automatic classification is to develop computer methods which learn to distinguish among a number of classes. Each class is represented by a set of example objects. When an appropriate mathematical representation of objects is found, here based on dissimilarity representations, a decision rule is constructed. Usually, standard two-class classification problems are studied first, since multi-class problems are often solved by combining two-class discrimination functions.

To construct a dissimilarity-based classifier, in general, a training set T of cardinality N and a representation set R of cardinality n will be used. R is a collection of prototypes objects from T. In a learning process, a classifier is constructed on the $N \times n$ dissimilarity representation $D(T, R)$, relating all training objects to all prototypes. The information on a set T_{te} of t new objects is provided by their dissimilarities to the examples from R, i.e. as a $t \times n$ dissimilarity matrix $D(T_{te}, R)$. Similarly as in chapter 8, dissimilarity representations are interpreted in three different ways, in pretopological spaces, in embedded spaces and in dissimilarity spaces. All these approaches together with particular discrimination functions are introduced and described in chapter 4.

Many interesting questions can be formulated for dissimilarity-based classification, so we can only discuss some of the most intriguing problems. Basically, we will demonstrate that the k-nearest neighbor (k-NN) method can be outperformed by alternative classifiers built on dissimilarity representations, especially for small representation sets. When the dissimilarity measure is discriminative and the classes are densely sampled, then in a close neighborhood of an object (measured by the given dissimilarity), there will be many objects of the same class. This is the reason why the k-NN rule is expected to perform well[1] for sufficiently large sample sizes. Thereby, it becomes our reference method.

Other essential questions refer to the selection of an informative representation set from a given training set, the use of non-metric dissimilarity measures and their possible corrections to make them metric and the usefulness of monotonic transformations. The results presented here come from our experiments conducted on various data sets. They are supported by articles [108, 109] and our earlier publications [103, 106, 290, 291, 293–296, 300, 301, 315].

9.1 Proof of principle

In this section, simple decision rules in dissimilarity spaces and in embedded spaces are considered to provide the 'proof of principle' that alternative dissimilarity-based classifiers are beneficial. This section aims at explaining our way of thinking and a general set-up of experiments. It plays an introductory role to the subsequent sections, where a more advanced study has been conducted.

[1] Under the assumption of sampling from the same probability distribution, the k-NN rule in a Euclidean space (with metric distances computed there) reaches asymptotically the error of at most twice the Bayes error.

Our experiments will demonstrate that the tradeoff between the recognition accuracy and the computational effort can be significantly improved by using a linear (or quadratic) classifier built in dissimilarity or embedded spaces, instead of the k-NN rule. Such a linear classifier is constructed from a training set described by the dissimilarities to the representation set. If this set is small, it has the advantage that only a small set of dissimilarities has to be computed for its evaluation, while it may still profit from the accuracy offered by a large training set.

9.1.1 Nearest neighbor rule and alternative dissimilarity-based classifiers

The k-NN rule [71], assigning an object to the class most frequently represented among its k nearest neighbors, is a simple and intuitive approach. Hence, it is commonly practiced in pattern recognition. It does not require any training, except for the choice of k. Conventionally, in a feature space, the k-NN rule relies either on the (appropriately weighted) Euclidean or city block distance, derived from the feature-based representations. For metric distances, it is known to be asymptotically optimal in the Bayes sense [87, 187]. It can learn complex boundaries and generalize well, provided that an increasing set of training objects T is available and the volumes of k-neighborhoods become close to zero. However, for a given finite training set (e.g. when the data points are sparsely sampled or have variable characteristics over the space), the classification performance of the k-NN method may significantly differ from its asymptotic behavior. To handle such situations, many variants of the NN rule as well as many distance measures have been invented or adopted for the feature based representations. They often take the local structure into account or weight the neighbor contributions appropriately; see e.g. [6, 89, 189, 251, 288, 331, 421]. So, such approaches are designed to optimize either the parameters of the distance measure over the feature space or the number of nearest neighbors k.

In this dissertation we study dissimilarity representations derived either from sensor measurements or from some other intermediate representations (e.g. string-descriptions, shapes or feature spaces). Consequently, we cannot always refer to the accompanying feature-based representation for the analysis of the NN rule, since such a space may not exist or might not be given. The dissimilarity measure is designed for a proper comparison of objects and, when derived, it serves for the construction of dissimilarity representations. In this sense, it is not optimized any longer.

For a test set T_{te}, the k-NN rule makes a decision by ranking the dissimilarities $D(T_{te}, R)$, where $R{:=}T$, and applying the voting mechanism. Although based on local k-neighborhoods, this method is still computationally expensive, since dissimilarities to all training examples have to be found. Another disadvantage is that it potentially decreases its performance when the training set is small. Also, the classification performance may be affected by the presence of noisy prototypes. Such limitations can be overcome by classifiers constructed either in dissimilarity spaces or in embedded spaces, which become 'more global' by making use of all representation objects. The advantage of such an approach is that the dissimilarity information is captured in some appropriate vector space, where many traditional classifiers can be adopted. Moreover, such a decision rule can be optimized by using a training set larger than the given representation set.

Many dissimilarity measures are based on sums of differences between (pre-processed) measurements. If such differences have approximately the same distributions (which may be a case for the standardized feature-based data or for the normalized image or spectra representations), their sum is approximately normally distributed. Hence, Bayesian classifiers assuming normal distributions, the (R)NLC or (R)NQC, (Regularized) Normal density based Linear or Quadratic Classifiers, as described in section 4.4.1, should perform well in such dissimilarity spaces. In practice, even if the assumption on normality is violated, such classifiers tend to work well. They may perform much better than the k-NN method, especially when the number of representation objects is small,

since they are less local in their decisions. By using weighted combinations of dissimilarities, they suppress the influence of noisy examples as well.

Since the training can be done off-line, here we are only concerned with the computational effort needed for the evaluation of a new object. Given n examples in the representation set and the computed n dissimilarities, the additional complexity of the RNLC is $\mathcal{O}(n)$ (products and sums), while the complexity of the RNQC is $\mathcal{O}(n^2)$[2]. The 1-NN rule requires $\mathcal{O}(n)$ comparisons and the k-NN rule needs at least $\mathcal{O}(n)$ and at most $\mathcal{O}(n \log(n))$ comparisons. Thereby, the k-NN rule might seem to be preferable[3]. However, our point is that the k-NN method requires a larger R than the RNLC/RNQC to reach the same accuracy. If the cost of computing dissimilarities is very high (dissimilarities are computed for data with a large amount of measurements such as images or spectra or applied in the template matching process), the cardinality of R is crucial for judging the computational complexity. Therefore, we claim, that the RNLC can improve the k-NN rule with respect to the recognition accuracy and computational effort. The same holds for the RNQC if R is small.

The other approach to dissimilarities relies on a linear embedding[4] into a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$, where $p+q=m$. Hence, the objects are represented as points in this space such that the pseudo-Euclidean distances between them reflect the original dissimilarities. Traditional discrimination functions operating in vector spaces can be adopted to make use of indefinite inner products. The details of such a construction can be found in section 3.3.3. The projection of a test object $D(t,R)$ onto $\mathbb{R}^{(p,q)}$ requires $\mathcal{O}(n)$ operations and the evaluation of a linear classifier needs $\mathcal{O}(m)$ operations (since we have an m-dimensional space), $m<n$, so the total complexity is $\mathcal{O}(nm)$; see also section 3.3.5. Consequently, this approach might be more computationally expensive than the use of dissimilarity spaces if m is large. The projection is unsupervised, i.e. no class information is used in the embedding (how to use it is an interesting point for a further study). The embedded space simply spatially reflects the dissimilarity information. By using linear or quadratic classifiers there, a better performance may be reached than by the k-NN applied to the original dissimilarities.

In summary, the k-NN rule operates on the dissimilarities directly, so by the use of local neighborhoods, it works in a pretopological space, section 4.3. The discrimination functions in dissimilarity spaces treat $D(T,R)$ as 'input features', hence build their decision based on (non-)linearly weighted dissimilarities $D(\cdot,R)$. The embedded spaces allow us to represent objects as points such that the distances are preserved. In this way (if the compactness hypothesis holds), we expect that the classes become relatively compact clouds of points. If the assumption on the true representation holds, then, additionally, there would be *no* overlap between the classes. Since an embedded space is a vector space, vector-based classifiers can be constructed there.

In the coming sections, we will present the results of two experiments. The first one shows the behavior of some classifiers in the three frameworks discussed above, as applied to square dissimilarity representations. The second experiment further explores the dissimilarity space approach.

[2] We assume that the number of classes c is very small with respect to n, cardinality of R, and that a c-class problem is solved by combining the result of c classifiers trained one-against-all. Hence $\mathcal{O}(c\,n)=\mathcal{O}(n)$ and $\mathcal{O}(c\,n^2)=\mathcal{O}(n^2)$.

[3] Since the k-NN method is often applied to metric distances, to avoid the expensive computation time, there has also been interest in approximate and fast NN search. Many algorithms have been proposed, usually making use of the triangle inequality. Examples can be found in [21, 179, 270, 271, 273, 313]. In our study, assuming general, possibly non-metric dissimilarities, we focus on the exact NN methods.

[4] There exist a number of nonlinear embeddings (usually with some distortion) into a Euclidean or l_1-space. Some of them were used in chapter 6 to visualize the dissimilarity data as 2D spatial configurations. Here, we focus on the linear embedding, mainly due to the computational aspect. Nonlinear mappings often require more operations than the linear ones. The study of the use of nonlinear mappings is left for future research.

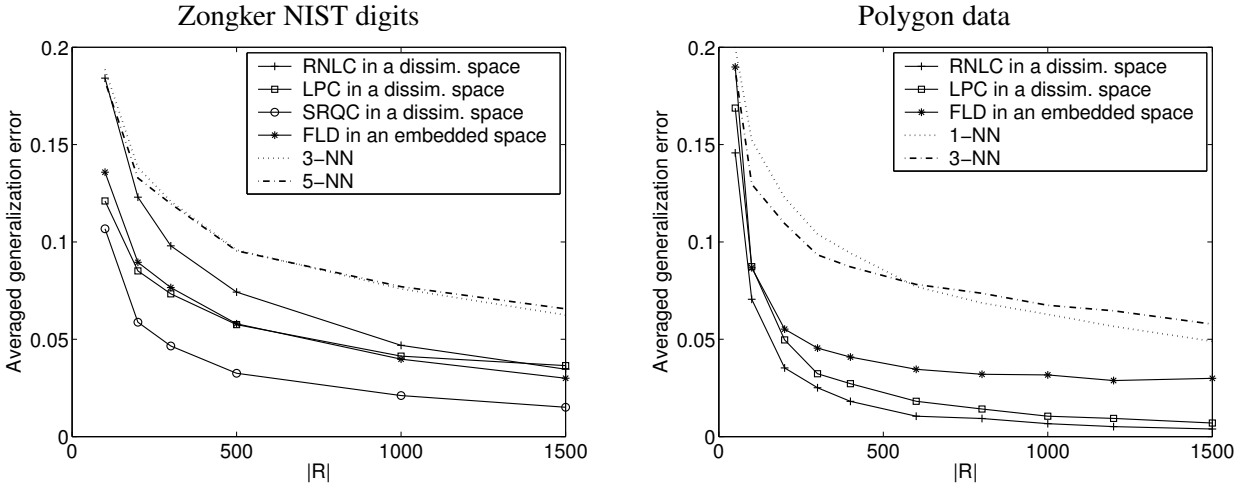


Fig. 9.1: Generalization error (averaged over 25 runs) of the decision functions in the dissimilarity and embedded spaces and the k-NN rule directly applied. The classifiers are trained on two dissimilarity representations: the Zongker NIST digits based on deformable template matching distance (left) and the Hausdorff distance between the polygon corners (right). In general, the standard deviations of the means are less than 0.007 and in the majority of cases, less than 0.003.

9.1.2 Experiment I: learning from square dissimilarity representations

The following example illustrates the use and benefits of dissimilarity-based classifiers over the direct use of the k-NN rule. Two data sets are chosen for this purpose; see section A.2 for details. The first data are the NIST handwritten digits [420], consisting of 2000 images of ten evenly probable classes. The similarity measure based on deformable template matching, as defined in [207] serves for building the non-metric dissimilarity representation. The second data refer to randomly generated polygons, consisting of 2000 polygons, evenly distributed over two classes of convex quadrilaterals and irregular heptagons. The polygons are compared by computing the Hausdorff distance, Def. 5.3, between their corners.

In both cases, the entire data set is randomly split into the design set L of 1500 examples and the test set T_{te} of 500 examples. Growing representation sets R (such that R ultimately becomes L) are randomly chosen from the design set L. Hence, for a growing set R, the following classifiers are built on $D(R, R)$ and tested on $D(T_{te}, R)$; see section 4.4.1 for the classifier descriptions:

1. The k-NN rule is applied to $D(T_{te}, R)$ directly.
2. Two linear classifiers in a dissimilarity space $D(R, R)$: the linear programming classifier (LPC), formulation (4.15), with the trade-off parameter of $\Lambda = 1$ and the RNLC with a fixed regularization parameter of $\lambda = 0.01$. The regularization is necessary, since otherwise the estimated covariance matrix becomes singular. Additionally, also the SQRC (strongly regularized quadratic classifier) is used for the NIST digits.
3. The Fisher linear classifier (FLD) in an embedded pseudo-Euclidean space. For the NIST digits, the dimensionality of the pseudo-Euclidean space is related to a fixed fraction of the preserved generalized variance, section 3.3.4, hence it will grow with a growing R. For the polygon data, the dimensionality is fixed to 45. These are two different approaches, since the number of eigenvalues significantly different from zero, determined in the embedding process (hence estimating the intrinsic dimensionality) seems to be fixed for the polygon data, while not for the deformable template matching distance on the NIST digits.

The results are shown in Fig. 9.1. For comparison, the test results of the best k-NN ($k = 1, 3, \ldots, 15$) rule are presented as well. These figures make clear that the alternative dissimilarity-based decision functions may perform well, much better than the best k-NN rule.

```
split the entire set into the design set L and the test set T_te
define a vector of the cardinalities r_R for the representation set R
for i = 1 to |r_R| do
    randomly select R ⊆ L of the cardinality r_R(i)
    error_{k-NN}(i) = test (k-NN, D(T_te, R))
    for z = i to |r_R| do
        choose the training set T of the cardinality r_R(z) such that
        T = R + objects randomly selected (per class) from L\R
        train (RNLC/RNQC, D(T, R))
        error_{RNLC/RNQC}(i, z) = test (RNLC/RNQC, D(T_te, R))
    end
end
```

Fig. 9.2: Pseudo-code for a single experiment in section 9.1.3.

9.1.3 Experiment II: the dissimilarity space approach

The experiments are conducted to compare the results of the k-NN rule and the RNLC and the RNQC built on dissimilarity representations[5]. They are designed to observe and analyze the behavior of these classifiers in relation to different sizes of both representation and training sets. We are concerned with possible gains of using small representation sets R and large training sets. A small R is of interest, because of both storage and computational aspects (the evaluation for new objects should be cheap).

Two different dissimilarity measures are studied for the NIST digit sets [420], represented by 2000 binary images, 200 images per class. The measures are: the Euclidean distance between Gaussian smoothed images (images are blurred to make the measure be somewhat robust against tilting and thickness) computed in a pixel-wise way and the modified Hausdorff distance, Def. 5.6, between the shape contours. The experiments are performed 25 times for randomly chosen training and test sets for each R under investigation. In a single experiment, each data set is randomly split into two equal-sized sets consisting of 1000 objects: the design set L and the test set T_{te}. L serves for obtaining both the representation set R and the training set T. After R is chosen, a number of training sets of different sizes are then considered. First, T is identical to R and then it is gradually enlarged by adding random objects until it becomes L.

There are many ways of selecting the representation set R out of the design set L; some of them will be discussed in the subsequent sections. Here, we do not study the best possible set R for the given problem, instead, we focus on illustrating our approach. Therefore, the representation objects are chosen randomly. Additionally, the condensed nearest neighbor (CNN) is used for the selection. In a single experiment, initially, a subset of the design set L is used for representation. Then, it is increased gradually by randomly adding new objects until it is equivalent to the complete set L. In this way a number of representation sets of different sizes can be studied.

The CNN criterion is based on the condensed nearest neighbor method [86, 187] developed to reduce the computational effort of the 1-NN rule. The CNN method finds a subset of the training set so that the 1-NN rule gives a zero error when tested on the remaining objects. Here, the representation set R becomes the condensed set found on the design set L. In contrast to the random selection, cardinality of R is automatically determined by the CNN method and it is fixed in a single experiment. However, since the training sets differ in all experiments, the number of representation objects may vary. Therefore, the size of R is averaged over all runs when reported in Table 9.1.

[5] The results presented here come from [293].

Both the RNLC and the RNQC, assuming normal distributions with equal or different class covariance matrices respectively, are built for different training sets. The regularized versions are used to prevent the estimated covariance matrices from being singular (e.g. in the case of the RNLC, when $|T|$ approaches $|R|$). Regularization takes care that the inverse operation is possible by emphasizing the variances with respect to the covariances; see also section 4.4.1. When $|T| \approx |R|$, then the estimation of the covariance matrices is poor. In such cases, different regularizations may significantly influence the performance of the RNLC/RNQC. For sufficiently large training sets, these matrices are well defined and no regularization is needed. In our experiments, the regularization parameters are fixed values of at most 0.01 for training sets such that $|T| \approx |R|$. Since they are not optimized, the results presented here might not be the best possible.

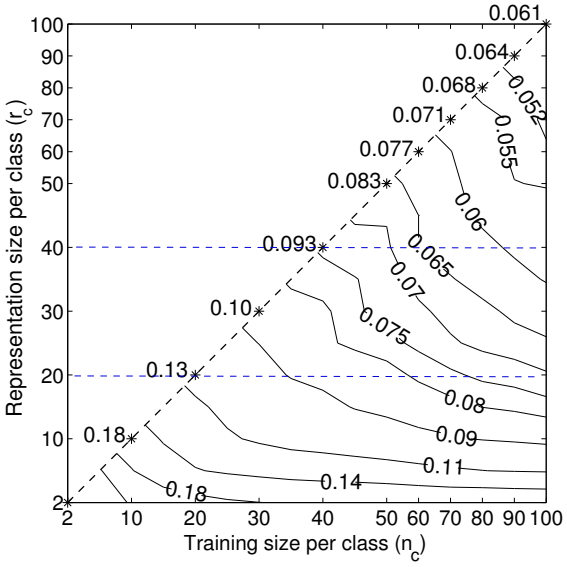
The pseudo-code for a single experiment is schematically shown in Fig. 9.2. In case of the k-NN rule, the following fixed choices of $k = 1, 3, 5, 7$ and 9 have been studied. Additionally, we have tried to optimize k via the leave-one-out procedure on $D(T, T)$. However, the k determined in such a way was always found to be one of the fixed, odd k mentioned above. For both digit sets, the best k-NN test results are found either for $k = 1$ or $k = 3$. In the experiments below we will report only the best test results for the studied values of k.

Since for the CNN criterion the cardinality of R is automatically determined by the method itself, the outer loop in the pseudo-code 9.2 is superfluous. The training sets are chosen differently than in case of a random selection. Here, the classes are likely to be unequally present in the determined set R, therefore the training set is constructed from R by adding objects, randomly selected from *all* the remaining examples in L. The generalization errors are averaged over the experiments and serve for making the plots.

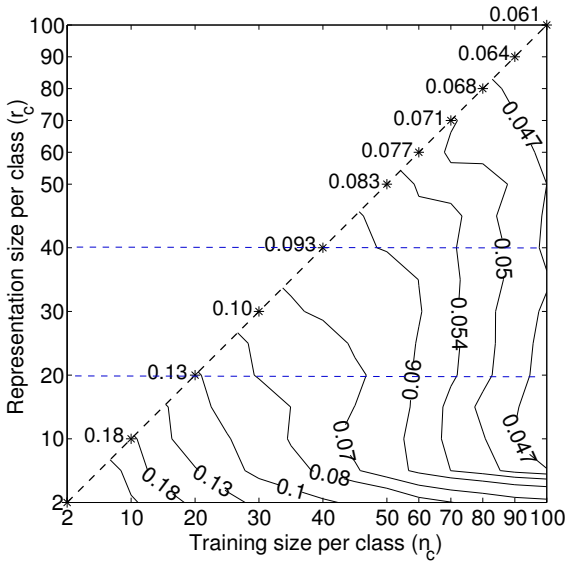
Results. The generalization error rates of the k-NN rule and the RNLC/RNQC are presented in Fig. 9.3. The k-NN results, marked by '*', are presented on the $r_c = n_c$ line. The RNLC's (RNQC's) curves are lines of the constant classification error (on independent test sets) relating the sizes of the representation and training sets. Additionally, Table 9.1 summarizes the results of the study. Given the fixed cardinality of R, the worst and the best results, depending on the training set size, are reported for the RNLC/RNQC. The CNN selection provides only a single set R of a fixed size.

The k-NN rule versus the RNLC. When T and R are identical, the RNLC (with error curves starting on the $r_c = n_c$ line in Fig. 9.3, left plots), generally yields a better performance than the equivalent k-NN rule based on the same R (compare also the k-NN results with the worst cases of the RNLC in Table 9.1). When r_c is fixed (i.e. in the horizontal directions of Fig. 9.3), the classifiers yield the same computational complexity for an evaluation of new objects. However, larger training sets reduce the error rate of the RNLC by a factor of 2 in comparison to the k-NN error (based on the same R). For instance, in Fig. 9.3(a), we observe that the classification error of 0.18 is reached by the k-NN rule based on $r_c = 10$ prototypes for which the RNLC offers a higher accuracy of ≈ 0.16 if trained also with $n_c = 10$ objects, reaching 0.09 when n_c increases to 100. In other words, for a chosen representation set R (hence a fixed computational complexity for an evaluation of a new object) the RNLC error, with the increase of training size, decreases significantly to the values that can only be obtained by the k-NN method if it is based on a much larger R. For instance, in Fig. 9.3(c), the RNLC built on $r_c = 10$ prototypes (and the training set of $n_c = 100$ objects) reaches an accuracy (an error of 0.12) for which the k-NN rule needs 40 objects in its representation set. The computational load with respect to the number of computed dissimilarities of the RNLC for the same classification accuracy is thereby reduced to 25%.

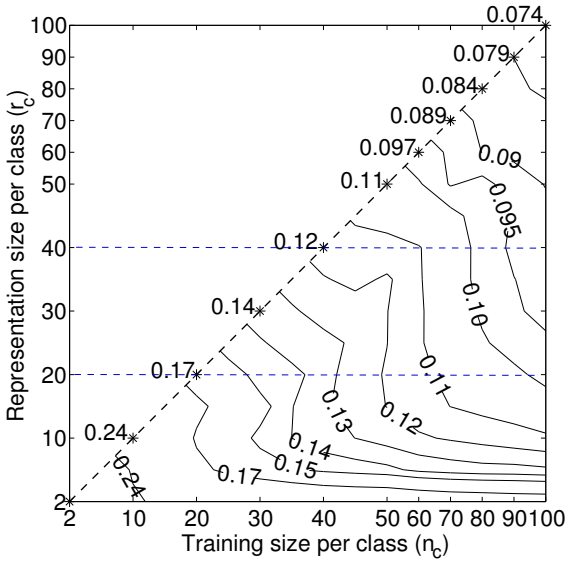
Following the RNLC's curves of constant error, it can be observed that for large training sets much small representations sets are needed for the same performance. The RNLC may sometimes demand only half the computational effort for the evaluation of new objects when compared to the k-NN



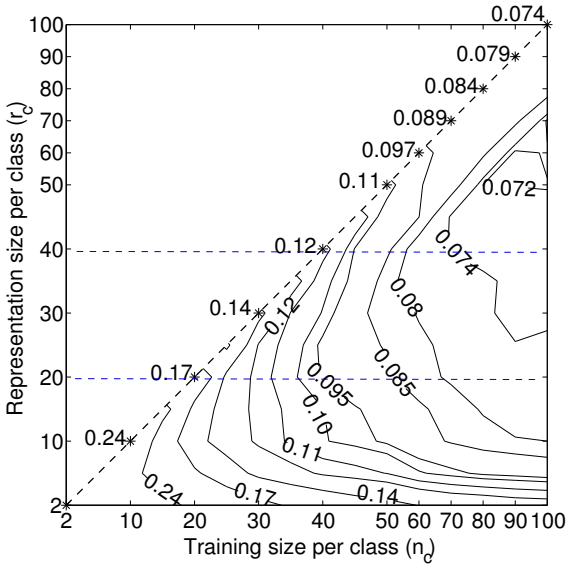
(a) The RNLC on the Euclidean DR



(b) The RNQC on the Euclidean DR



(c) The RNLC on the modified-Hausdorff DR



(d) The RNQC on the modified-Hausdorff DR

Fig. 9.3: Generalization errors (averaged over 25 runs) for the blurred Euclidean (top) and modified-Hausdorff (bottom) dissimilarity representations derived on the pixel-based NIST digit set. The lines correspond to the averaged generalization error lines of the RNLC (left) and the RNQC (right) in dissimilarity spaces. The k-NN results are indicated by '*'. All the representation sets are chosen randomly. If a horizontal line is drawn at the fixed r_c, then its crossing points with the error lines determinate the number of training objects n_c needed for reaching a specific performance. For instance, in subplot (c), for $r_c = 20$, the RNLC needs $n_c \approx 30$ training objects to reach the error of 0.15 and $n_c = 95$ objects to reach the error of 0.1. The k-NN error equals 0.17 for $n_c := r_c = 20$.

method. Also, for the fixed, possibly large training set (i.e. in the vertical directions of the considered figures), the RNLC constructed on a small R, might gain a similar or higher accuracy than the k-NN rule, but now based on the complete $D(T,T)$. This is observed, e.g. in Fig. 9.3(a) for $n_c = 40$. The k-NN method yields an error of 0.093 and the RNLC reaches a smaller error when trained on $D(T,R)$ with R consisting of $r_c \geq 20$.

Since the best k-NN results for both digit data sets are found for $k = 1$ or 3 [293], the results of the 1-NN rule based on the CNN criterion can be compared to the results of the k-NN rule

Table 9.1: Averaged generalization error (in %) with its standard deviation for the k-NN rule and the RNLC/RNQC in dissimilarity spaces for the blurred NIST digit data set and a random selection of the representation set R with r_c objects per class. Additionally, R is also selected by the CNN criterion. Here, the presented RNLC/RNQC errors refer either to the worst (left column) or to the best (right column) results achieved for a fixed r_c.

Euclidean dissimilarity representation					
Random selection					
r_c	k-NN	RNLC		RNQC	
10	17.5 (0.4)	15.6 (0.3)	8.6 (0.1)	19.0 (0.5)	4.4 (0.1)
20	12.5 (0.3)	10.2 (0.1)	7.1 (0.1)	10.3 (0.2)	4.6 (0.1)
50	8.3 (0.2)	6.6 (0.2)	5.5 (0.1)	5.6 (0.2)	4.7 (0.1)
70	7.1 (0.2)	5.8 (0.1)	5.1 (0.1)	5.0 (0.1)	4.6 (0.1)
90	6.4 (0.1)	5.1 (0.1)	5.0 (0.1)	4.6 (0.1)	4.6 (0.1)
CNN selection					
r_c	1-NN	RNLC		RNQC	
20	10.6 (0.2)	8.5 (0.2)	5.7 (0.1)	8.7 (0.4)	4.6 (0.1)
Modified-Hausdorff dissimilarity representation					
Random selection					
r_c	k-NN	RNLC		RNQC	
10	24.4 (0.4)	21.3 (0.3)	11.1 (0.2)	34.9 (0.8)	8.0 (0.2)
20	17.1 (0.2)	15.6 (0.3)	9.8 (0.2)	21.2 (0.5)	7.4 (0.2)
50	10.6 (0.2)	10.3 (0.2)	9.0 (0.2)	9.9 (0.2)	7.2 (0.2)
70	8.9 (0.1)	9.2 (0.2)	8.7 (0.2)	8.2 (0.2)	7.2 (0.2)
90	7.9 (0.2)	8.5 (0.2)	8.2 (0.2)	8.2 (0.2)	8.3 (0.2)
CNN selection					
r_c	1-NN	RNLC		RNQC	
24	12.8 (0.2)	12.6 (0.3)	9.5 (0.2)	12.3 (0.4)	6.8 (0.2)

based on a random selection of R. The former are better than the latter, probably because the CNN representation set is optimized for the 1-NN. Also, as observed in Table 9.1 and in Fig. 9.3, the RNLC defined on the CNN representation set generalizes better than the RNLC defined on a random representation set.

The RNLC versus the RNQC. In general, the RNQC performs better than the RNLC for both dissimilarity data sets; compare the results in Fig. 9.3, left plots versus right plots. Since the RNQC relies on the class covariance matrices in a dissimilarity space, a larger number of samples is needed than for the RNLC to obtain reasonable estimates. The RNQC may reach a worse accuracy than the RNLC for identical T and R. However, following the curves of the RNQC's constant error, both smaller representation and training sets are needed for the same error when compared to the RNLC. The RNQC's curves are simply much steeper than those of the RNLC. Thereby, the RNQC outperforms the RNLC for large training sets (and small R). The most significant improvement can be observed for a small R. For instance, the training set of $n_c = 100$ examples allows the RNLC to reach the error of 0.049 when based on $r_c \geq 70$ prototypes, see Table 9.1, where the RNQC requires only between 5 and 30 prototypes for a similar performance; see Fig. 9.3(c). When the largest training sizes are considered (the best results in Table 9.1) for the fixed set R, the error of the RNQC decreases, yielding better results than the k-NN rule. Still, when the smallest errors of the RNLC and RNQC are compared, the RNQC generalizes better. Also, for the fixed training set T, i.e. in the vertical directions in Fig. 9.3, subplots (b) and (d), a small representation set R often allows the RNQC (trained on $D(T, R)$), to reach a better performance than the k-NN rule based on $D(T, T)$.

9.1.4 Discussion

Our experiments indicate that indeed a good classification performance can be reached by dissimilarity-based classifiers, an alternative to the k-NN method. Even if the classifiers are trained in n-dimensional dissimilarity space $D(R, R)$ determined by the dissimilarities to n prototypes, they may work better than the k-NN rule defined on the same R.

The experiments focus further on the dissimilarity-space approach and the role of the representation set R. They show that the RNLC constructed on the dissimilarity representations $D(T, R)$ may significantly outperform the k-NN rule based on the same R. This holds for the RNQC as well, provided that each class is represented by a sufficient number of training objects (they are needed to estimate the class covariance matrices reliably). Since for the evaluation of new objects the computational complexity (here indicated by the number of prototypes) is an important issue, our experiments are done with such an emphasis. We have found out that for the fixed representation set, larger training sets improve the performance of the RNLC/RNQC. When such results are compared to the k-NN based on the same R, they are often better. Also, for the fixed training set T, smaller (than T) representation sets allow the RNLC/RNQC, trained on $D(T, R)$, to gain a high accuracy. When R is only somewhat smaller than T, such classification errors can be smaller than the ones reached by the k-NN based on the entire training set T, i.e. $D(T_{te}, T)$.

The potentially good performance of the RNLC can be understood as follows. The RNLC is in fact a weighted linear combination of the dissimilarities between an object x and the prototypes. It seems practical to allow a number of representation examples of each class to be involved in a discrimination process. This is already offered by the k-NN rule, however, this decision rule provides an absolute answer (due to a mechanism based on the majority voting). The k-NN method is sensitive to noise, so the k nearest neighbors found might not include the best representatives of a class to which an object should be assigned. The training process of the RNLC, using a larger training set T, emphasizes prototypes which play a crucial role during discrimination, but it still allows other prototypes to influence the decision. The importance of prototypes is reflected in the classifier weights. In this way, a classifier is built, which takes all prototypes into account.

The RNQC includes also a sum of the weighted products between pairs of dissimilarities to R. By doing this, some interactions between the prototypes are emphasized. The RNQC is based on the class covariance matrices in a dissimilarity space, estimated separately for each class. Those matrices may really differ from class to class. Therefore, this decision rule might achieve a higher accuracy than the RNLC, where all class covariance matrices are averaged. However, a larger number of samples (with respect to the size of R) is required to obtain reasonable estimates for all covariance matrices, and, thereby, a good generalization ability of the RNQC.

9.2 Selection of the representation set: the dissimilarity space approach

In the dissimilarity space approach decision rules are functions of dissimilarities to the selected representation objects (prototypes). Assuming that the entire dissimilarity representation $D(T, T)$ is available, the question now arises how a small representation set R should be selected out of T to guarantee a good tradeoff between the recognition accuracy and the computational complexity. We know that a random selection of prototypes may work well [286, 287, 293, 296, 301], as also indicated in the previous section. Here, we will analyze some systematic procedures[6]. Since the selection of prototypes is usually investigated in the context of metric k-NN rules, before we move on, we will briefly discuss this point.

[6] The results presented here come from [300].

In the basic setup, the k-NN rule uses the entire training set as the representation set, hence $R=T$. Therefore, the usual criticism points at a space requirement to store the complete set T and a high computational cost for the evaluation of new objects. The k-NN rule also shows sensitivity to outliers, i.e. noisy or erroneously labeled objects. To alleviate these drawbacks, various techniques have been developed in feature spaces to tackle the problem of prototype optimization. So, some research efforts have been devoted to this task; see e.g. [78, 187, 330, 422]. From the initial prototypes (say, all training objects), the prototype optimization method chooses or constructs a small portion of them such that a high classification performance is achieved.

Two main types of algorithms can be identified: prototype generation and prototype selection. The first group focuses on merging the initial prototypes (i.e. the prototypes represented as vectors in a feature space are replaced e.g. by their average vector) into a small set of prototypes such that the performance of the k-NN rule is optimized. Examples of such techniques are the k-means algorithm [97] or a learning vector quantization algorithm [219]. The second group of methods aims at the reduction of the initial training set and/or the increase in the accuracy of the NN predictions. This leads to various editing or condensing methods. Condensing algorithms try to determine a significantly reduced set of prototypes such that the performance of the 1-NN rule on this set is close to the one reached on the complete training set [78, 187, 422]. This is the consistency property [78]. Editing algorithms remove noisy samples as well as close border cases, leaving smoother decision boundaries [86, 422]. They aim to leave homogeneous clusters in the data. Basically, they retain all internal points, so they do not reduce the space as much as other reduction algorithms do. Usually, they are followed by condensing methods.

Although the k-NN rule is often practiced with metric distances, there are problems when the designed dissimilarity measures are non-metric, such as the modified Hausdorff distance and its variants [93], Mahalanobis distance between probability distributions [97] or the normalized edit-distance [47, 262, 410]; see also chapter 5. Such non-metric measures seem to naturally arise in template matching processes applied e.g. in computer vision [93, 206]. If the dissimilarity measure is meaningful, the principle behind the voting among the nearest neighbors can be applied to non-metric dissimilarities and the k-NN rule may work well; see e.g. [296] or the subsequent sections. It is simply more important that the measure itself is discriminative and describes the classes in a compact way than its strict metric properties. However, many traditional prototype optimization methods are not appropriate for non-metric dissimilarities, especially if no accompanying feature-based representation is available, as they can be based on the triangle inequality, for instance. Moreover, there are also situations, where the classes are badly sampled due to the problem characteristics as e.g. in machine or health diagnostics, or due to the measurement costs. In such cases, the k-NN rule, even for a large k and a very large training set will suffer from noisy examples. Yet, we think that much more can be gained when other discrimination functions, such as linear classifiers in a dissimilarity space, are constructed. In general, as pointed in the previous section, such classifiers make their decisions by averaging the information from a number of prototypes and they seem to be more robust against local distortions.

9.2.1 Prototype selection methods

The selection of a representation set for the construction of classifiers in a dissimilarity space serves a similar goal as the selection of prototypes to be used by the NN rule: minimization of a set of dissimilarities to be measured for the classification of new incoming objects. There is, however, an important difference with respect to the demands. Once selected, the set of prototypes defines the NN classifiers independently of the remaining part of the training set. The selection of the representation set, on the other hand, is less crucial, as it will define a dissimilarity space in which the entire training set is used to train a classifier. For this reason, even a randomly selected representation

set may work well [293]. That is why, the random selection will serve as a basic procedure for comparing more advanced techniques.

Similar objects will yield a similar contribution to the representation. It may, thereby, be worthwhile to avoid the selection of objects with small dissimilarity values. Moreover, if the data describe a multi-modal problem, it may be advantageous to select objects related to each of the modes. Consequently, the use of procedures like vector quantization or cluster analysis can be useful for the selection of prototypes.

Assume c classes $\omega_1, \ldots, \omega_c$. Let T be a training set and let T_{ω_i} denote the training objects of the class ω_i. Each method selects K objects for the representation set R. If the algorithm is applied to each class separately, then k objects per class are chosen such that $ck = K$. The following procedures will be compared for the selection of a representation set: Random, RandomC, KCentres, ModeSeek, LinProg, FeatSel, KCentres-LP and EdiCon.

Random. A random selection of K objects from the training set T.

RandomC. A random selection of k objects per class (equal class prior probabilities are assumed).

KCentres. This is a representation-based clustering procedure, described in section 7.1.2. For each class ω_i, this algorithm chooses a set R_{ω_i} of k objects such that they are evenly distributed with respect to the dissimilarity information $D(T_{\omega_i}, T_{\omega_i})$. Since the final result depends on the initialization, some precautions are taken. To determine R_{ω_i}, we start from one center for the entire set $D(T_{\omega_i}, T_{\omega_i})$ and then more centers are gradually added. At any point, a group of objects belongs to each center. R_{ω_i} is enlarged by splitting the group of the largest radius into two and replacing its center by two other members of that group. This stops, when k centers are determined. The entire procedure is repeated 30 times, resulting in 30 potential representation sets. The final set R_{ω_i} is the one which yields the minimal of the largest subset radii. The representation set R consists of all sets R_{ω_i}.

ModeSeek. For each class ω_i the mode seeking algorithm [63] looks for a set R_{ω_i} consisting of the estimated modes of class distribution with respect to $D(T_{\omega_i}, T_{\omega_i})$. The final cardinality of R_{ω_i} depends on the specified neighborhood size s. The larger the neighborhood s, the smaller R_{ω_i}. If a representation set of a particular cardinality is searched, we select s such that it generates the largest set which is not larger than the demanded one. This algorithm is a clustering algorithm and it was introduced in section 7.1.2.

The procedures above may be called unsupervised, in spite of the fact that they are used in a class-wise way. They aim at various heuristics, but they do not consider the quality of the resulting representation set in terms of the class separability. A standard procedure to do that is by feature selection.

FeatSel. In traditional pattern recognition, the feature selection method determines an optimal set of K features according to some class separability measure. It is often done in the forward selection process [207] by using either the Mahalanobis distance or the leave-one-out 1-NN error. This standard approach is modified here to make use of a given dissimilarity representation. The entire dissimilarity matrix $D(T, T)$ is reduced to $D(T, R)$ by selecting an optimal set of K prototypes according to the leave-one-out 1-NN error. There is, however, a difference with respect to the standard feature selection procedure. Features are considered in a dissimilarity space, but the 1-NN error is computed on the given dissimilarities $D(T, T)$ directly, and *not* by the Euclidean distances derived from the given dissimilarity representation. The method is, thereby, fast as it is entirely based on comparisons and sorting. Ties can easily occur by the same number of misclassified objects for different representation sets. They are solved by selecting the set R for which the sum of dissimilarities is minimum.

LinProg. The selection of prototypes is done automatically by training a properly formulated separating hyperplane $f(D(x,R)) = \sum_{j=1}^{n} w_j\, d(x,p_j) + w_0 = \mathbf{w}^T D(x,R) + w_0$ in a dissimilarity space $D(T,R)$. R can be chosen identical to the training set T, but it can also be different. Here, we assume that $R{:=}T$. This linear function is obtained by solving a linear programming problem, where a sparse solution is imposed by minimizing the l_1-norm of the weight vector $\mathbf{w}$, $||\mathbf{w}||_1 = \sum_{j=1}^{n} |w_j|$. Such a minimization task is described in section 4.4.1. We focus on the formulation (4.15).

As a result, since a sparse solution $\mathbf{w}$ is obtained, many weights w_i tend to become zero. The objects from the initial set $R := T$ corresponding to non-zero weights are the selected prototypes, i.e. the representation set R_{LP}. Although the prototypes are found in the optimization for a particular separating hyperplane, they can be used by other discrimination functions as well. We have found out that the choice of the tradeoff parameter as $\gamma = 1$, see (4.15), seems to be reasonable for many problems, so we fix it in our experiments.

Such a prototype selection method is similar to a selection of features by linear programming in a standard classification task [41]. The important point to realize is that we do not have a control over the number of selected prototypes. This can be slightly influenced by varying the constant γ (hence influencing the tradeoff between the classifier norm $||\mathbf{w}||_1$ and the training classification errors), but not much. From the computational point of view, this procedure is advantageous for two-class problems, since multi-class problems may result in a large set R_{LP}. This occurs since different prototypes are often selected by different classifiers when a multi-class classifier is derived in the one-against-all strategy or even more severely in the pairwise strategy.

KCentres-LP. The KCentres algorithm is applied to a square dissimilarity representation $D(T,T)$ to pre-select a representation set R_{KC}. This is then followed by a reduction based on the LinProg procedure applied to $D(T,R_{KC})$. In this way, the number of resulting prototypes can be somewhat influenced. Still, if R_{KC} is not sufficiently large, the linear programming will make no reduction. Hence, this procedure reduces to the KCentres approach for a small R_{KC}.

EdiCon. An editing and condensing algorithm [86] is applied to the entire dissimilarity representation $D(T,T)$, resulting in a representation set R. Editing takes care that the noisy objects are first removed so that the prototypes can be chosen to guarantee a good performance of the 1-NN (k-NN) rule. Similarly as in the case of the LinProg, the number of prototypes is automatically determined.

9.2.2 Experimental setup

If a good dissimilarity measure is found, and a training set is sufficiently large and representative for the problem at hand, then the k-NN rule (based on $R{:=}T$) is expected to perform well. In other cases, a better generalization can be achieved by a linear or quadratic classifier built in dissimilarity spaces. The weights of such decision rules are optimized on a training set and large weights (in magnitude) emphasize prototypes which are essential for discrimination. In the previous section, as well as in our studies [293, 296, 301], we have found out that the linear and quadratic normal density based classifiers, the NLC and NQC, respectively, perform well in dissimilarity spaces.

Some experiments are conducted to compare various prototype selection methods for the classification in dissimilarity spaces. Smaller representation sets are of interest, because of a lower complexity for both representation and evaluation of new objects. Both linear (the NLC) and quadratic (the NQC) classifiers are considered in dissimilarity spaces. Here, we will present only the results for the NQC, since it generally performs better than the NLC. In higher-dimensional dissimilarity spaces, i.e. for larger representation sets, the NQC is, however, computationally more expensive than the NLC. Since we decided to compare all selection strategies by the performance of a single classifier, as a result, the LinProg was simply used for the selection of R and not as a discrimination

Table 9.2: Characteristics of the data sets used in experiments. α stands for the fraction of objects selected for training in each repetition.

Data	# classes	# objects per class (in total)	α per class
Polydisth	2	$2 \cdot 2000$	0.25
Polydistm	2	$2 \cdot 2000$	0.25
NIST-38	2	$2 \cdot 1000$	0.10
Zongker-12	2	$2 \cdot 100$	0.50
GeoSam	2	$2 \cdot 500$	0.50
GeoShape	2	$2 \cdot 500$	0.50
Wine	3	59/71/48	0.60
Ecoli-p08	3	143/77/52	0.60
ProDom	4	878/404/271/1051	0.35
Zongker-all	10	$10 \cdot 100$	0.50

Table 9.3: Properties of the data sets used in experiments. The following abbreviations are used: M - metric, E - Euclidean, nM - non-metric, nE - non-Euclidean. The values r_{mm}^{nE} and r_{rel}^{nE} indicate the deviations from the Euclidean behavior, as defined in formula (9.1) and r_{tr}^{nM} describes the percentage of disobeyed triangle inequalities.

Data	Dissimilarity	Property	$r_{mm}^{nE}[\%]$	$r_{rel}^{nE}[\%]$	$r_{tr}^{nM}[\%]$
Polydisth	Hausdorff	M, nE	25.1	38.1	0.00
Polydistm	Mod. Hausdorff	nM	11.0	31.4	0.01
NIST-38	Euclidean	E	0.0	0.0	0.00
Zongker-12	Template-match	nM	13.3	30.1	0.70
GeoSam	SAM [239]	M,nE	0.1	0.1	0.00
GeoShape	Shape l_1	M, nE	2.6	7.2	0.00
Wine	Euclidean distance	E	0.0	0.0	0.00
Ecoli-p08	$l_{0.8}$ distance	nM	13.4	24.7	3.84
ProDom	Structural	nM	1.3	0.9	10^{-5}
Zongker-all	Template-match	nM	38.9	35.0	0.41

function. (otherwise it would not be comparable to the performance of the NQC for based on some other representation set.)

In each experiment, each data set is divided into a training set T and a test set T_{te}. The NQC is trained on the dissimilarity representation $D(T, R)$ and tested on $D(T_{te}, R)$. $R \subset T$ is a representation set consisting of K prototypes chosen according to some specified criterion, as described in section 9.2.1. The 1-NN and the k-NN results defined on the entire training set (hence tested on $D(T_{te}, T)$ are provided as reference. Also, as a comparison, the k-NN rule is directly applied to $D(T_{te}, R)$, with R selected by the KCentres algorithm and to the Euclidean distances computed in the representation $D(T_{te}, R)$. (This corresponds to the k-NN performed in the dissimilarity space). The k-NN rule optimizes k over the training set T in the leave-one out manner [101].

Specification of the data sets. In all our experiments the data sets are divided into training and test sets of various sizes; details can be found in Table 9.2. We have chosen a number of problems possessing various characteristics: defined by both metric (Euclidean or non-Euclidean) and non-metric dissimilarity measures, as well as, concerning small and large sample size problems. Seven data sets are used in our study: randomly generated polygons, NIST scanned digits, geophysical spectra proteins and their localization sites and wine types, resulting in ten dissimilarity representations (for some data sets, two different measures are considered). The data sets refer to two-, three-, four- and ten-class classification problems. All data sets are described in Appendix A.1 and A.2.

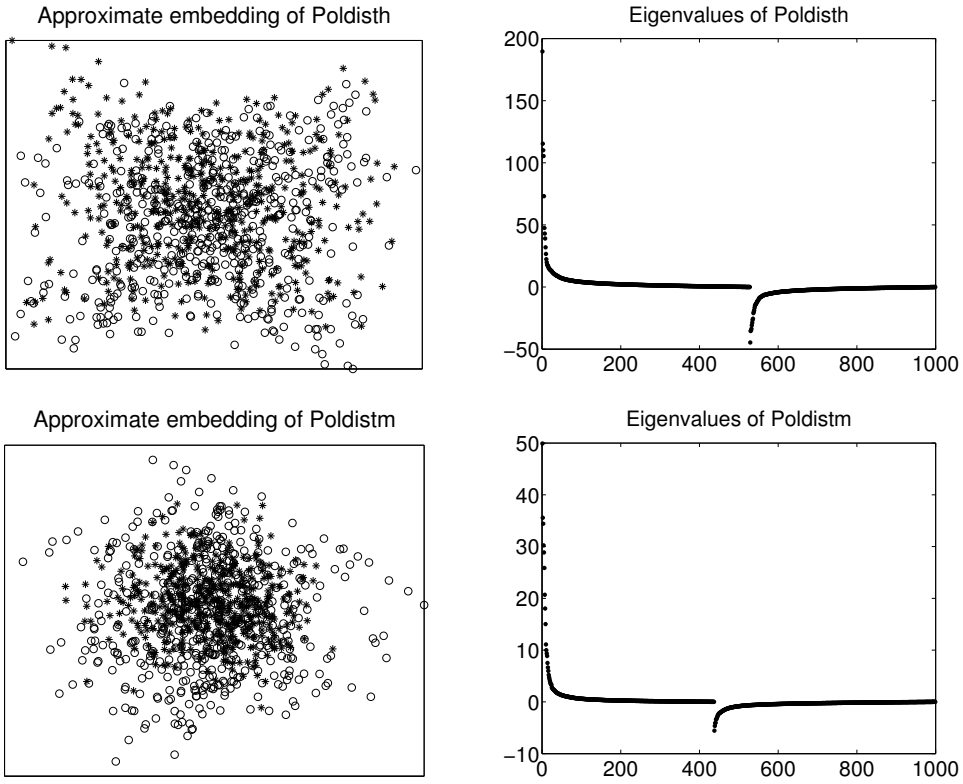


Fig. 9.4: Left: approximate 2D embedding of the dissimilarity representations $D(T,T)$ for the polygon data. Right: all the eigenvalues derived in the embedding process.

If a dissimilarity d is Euclidean, then the square $N \times N$ dissimilarity representation $D(T,T)$ can be perfectly embedded in a Euclidean space. This means that a configuration X can be found such that the Euclidean distances between the vectors of X correspond to the original ones. This is equivalent to the statement that the Gram matrix $G = -\frac{1}{2} J D^{*2} J$, where $D^{*2} = (d_{ij}^2)$ and $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, is positive semidefinite i.e. all its eigenvalues are nonnegative. A non-Euclidean representation D can be embedded in a pseudo-Euclidean space. The configuration X is determined in this space by eigendecomposition of the Gram matrix G as $G = Q \Lambda Q^T$, where Λ is a diagonal matrix of decreasing positive eigenvalues followed by decreasing (in magnitude) negative eigenvalues and then zeros, and Q is an orthogonal matrix of the corresponding eigenvectors. X is found as $X = Q_m |\Lambda_m|^{1/2}$, where m corresponds to the number of non-zero eigenvalues. See section 3.3 for details.

Let the eigenvalues be denoted by λ's. Hence, the magnitude of negative eigenvalues indicates the amount of deviation from the Euclidean behavior. This is captured by the following indices:

$$\begin{aligned} r_{mm}^{nE} &= \frac{|\lambda_{min}|}{\lambda_{max}} \cdot 100 \\ r_{rel}^{nE} &= \frac{\sum_{\lambda_i<0} |\lambda_i|}{\sum_{j=1}^{N} |\lambda_j|} \cdot 100. \end{aligned} \tag{9.1}$$

r_{mm}^{nE} is the ratio of the smallest negative eigenvalue to the largest positive one, while r_{rel}^{nE} describes the contribution of negative eigenvalues. Additionally, an indication of the non-metric behavior can be expressed by the percentage of disobeyed triangle inequalities, r_{tr}^{nM}.

Table 9.3 provides suitable information on the Euclidean and metric aspects of the measures considered. The Hausdorff representation of the polygon data are strongly non-Euclidean. The modified Hausdorff representation of the polygon data, as well as template-matching representation of the digits data are moderately non-Euclidean and non-metric. Concerning the geophysical data, the shape dissimilarity representation is slightly non-Euclidean, while the SAM representation is nearly

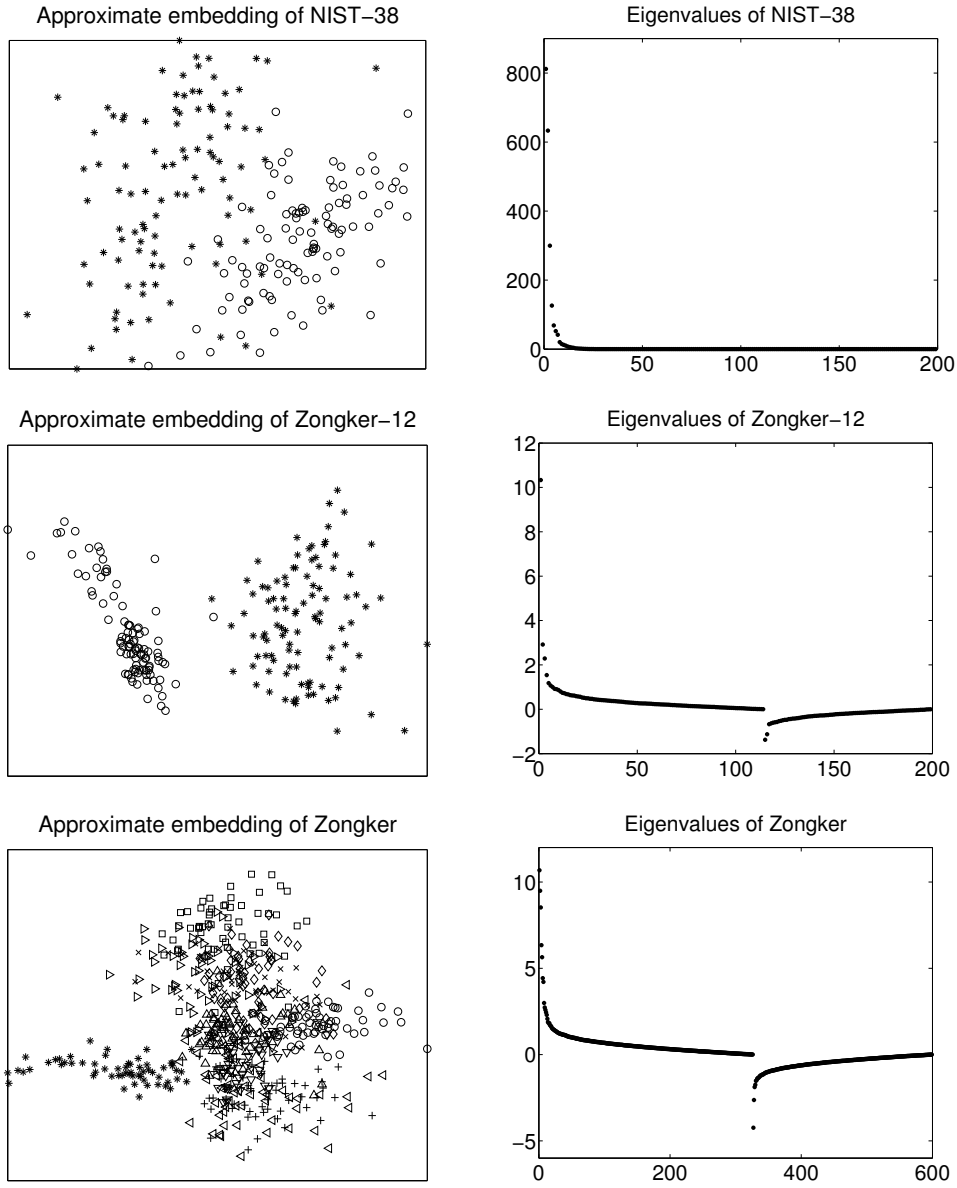


Fig. 9.5: Left: approximate 2D embedding of the dissimilarity representations $D(T,T)$ for the NIST data. Right: all the eigenvalues derived in the embedding process.

Euclidean. Both are metric. For the Ecoli data, the non-metric $l_{0.8}$ distance representation is used. ProDom representation is slightly non-metric and slightly non-Euclidean. The remaining two data sets: NIST digits and Wine have Euclidean representations.

For the purpose of visualization also 2D approximate embeddings of dissimilarity representations have been found. They rely on linear projections from the corresponding Gram matrices, as described above; see also section 3.3. The sum of the first two largest eigenvalues with respect to the total sum of all eigenvalue magnitudes indicates how much of the original dissimilarities is reflected in the projections. This can be observed in figures 9.4 - 9.9. There we also show all the eigenvalues of the Gram matrices (derived from the dissimilarity matrices), hence the deviation from the Euclidean behavior can be visually judged. The number of eigenvalues significantly different from zero indicates the intrinsic dimensionality of a problem. The approximate embeddings are used for the purpose of exploratory data analysis. As judged from two-class problems, figures 9.4 - 9.9, the polygon data seem the most complex, while the *Zongker-12* data seem the easiest. On the other hand, the ten-class *Zongker-all* data are the most complex.

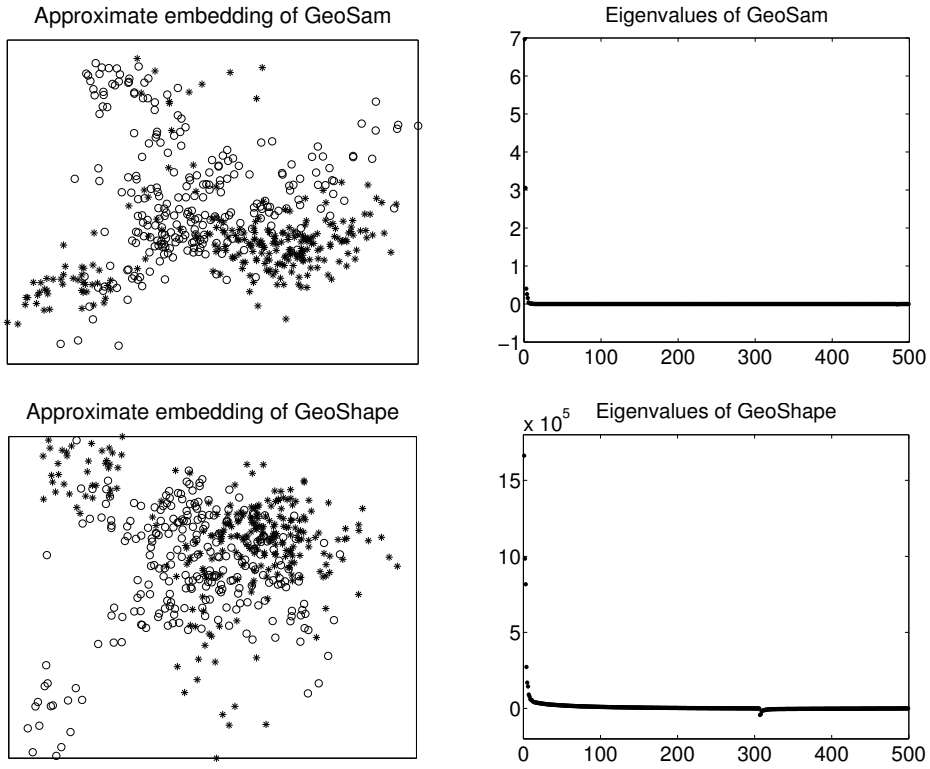


Fig. 9.6: Left: approximate 2D embedding of the dissimilarity representations $D(T,T)$ for the geophysical spectra data. Right: all the eigenvalues derived in the embedding process.

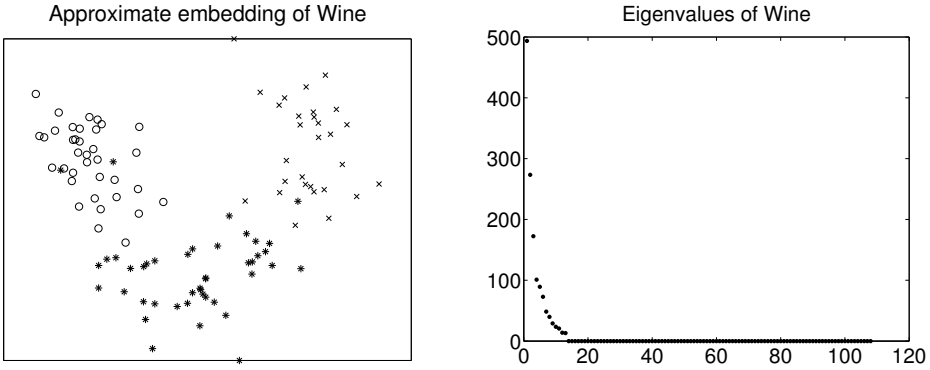


Fig. 9.7: Left: approximate 2D embedding of the dissimilarity representation $D(T,T)$ for the *Wine* data. Right: all eigenvalues derived in the embedding process.

9.2.3 Results and discussion

The results of our experiments are presented in Fig.9.10 - 9.16. They show the generalization errors of the NQC as a function of the number of prototypes chosen by various selection methods. These error curves are compared to some variants of the NN rule. Note that in order to emphasize a small number of prototypes, the horizontal axis is logarithmic. The prototype selection methods mentioned in the legends are explained in section 9.2.1. Concerning the NN methods, the following abbreviations are used. The 1-NN-final and the k-NN-final stand for the NN results obtained by using the entire training set T, hence such errors are plotted as horizontal lines. They are our reference. k-NN is the k-NN rule directly applied to $D(T,R)$, while the k-NN-DS is the Euclidean distance k-NN rule computed in $D(T,R)$ dissimilarity spaces (this means that a new Euclidean distance representation is derived from the vectors $D(x,R)$). In both cases, the representation set

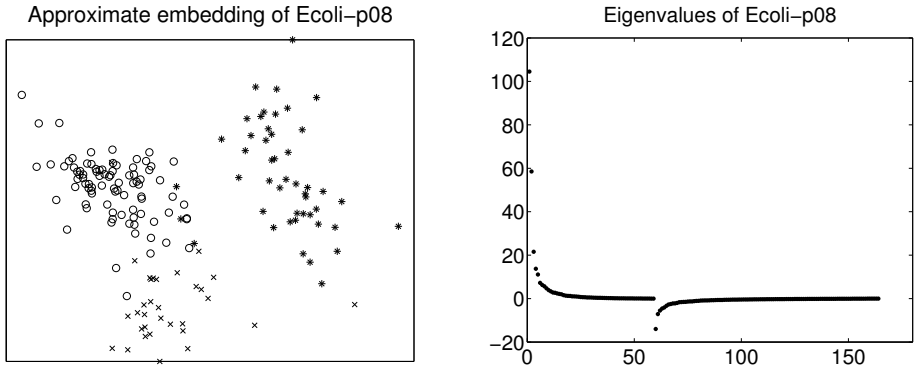


Fig. 9.8: Left: approximate 2D embedding of the dissimilarity representations $D(T,T)$ for the *Ecoli-p08* data. Right: all the eigenvalues derived in the embedding process.

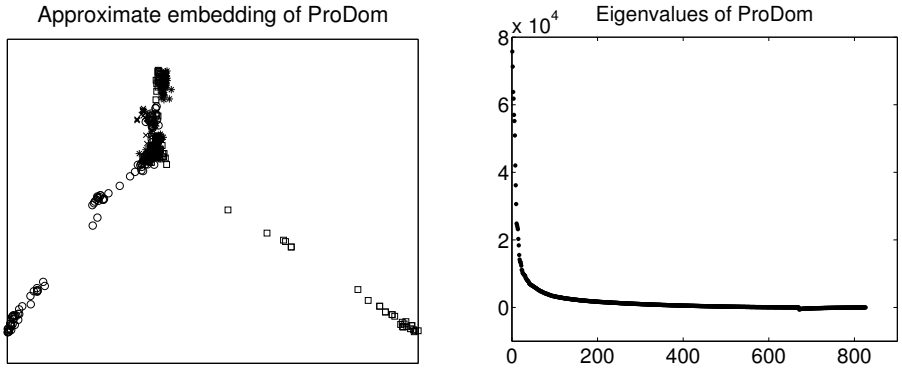


Fig. 9.9: Left: approximate 2D embedding of the dissimilarity representation $D(T,T)$ for the *ProDom* data. Right: all the eigenvalues derived in the embedding process.

R is chosen by the KCentres algorithm. EdiCon-1-NN presents the 1-NN result for the prototypes chosen by the editing and condensing (EdiCon) criterion. The optimal parameter k in all the k-NN rules used is determined by the minimization of the leave-one-out error on the training set. Sometimes, k is found to be 1 and sometimes, some other value.

The performances of all procedures mentioned in the legends, from the Random to EdiCon selections are based on the NQC in the dissimilarity space defined by the selected set of prototypes R. So, they need just the computation of the reduced set of similarities for testing purposes, but they profit indirectly from the availability of the entire training set T.

To enhance the interpretability of the results, the following patterns are used in the plots. The supervised methods, the KCentres-LP and FeatSel are plotted by continuous lines, the unsupervised, clustering selections are plotted by dash-dotted lines and the random methods are plotted by dashed lines.

Our experiments are based on M repetitions, that is M random selections of a training set. $M=10$ for the *Prodom* and *Zongker-all* dissimilarity data and $M=25$, otherwise. The remaining part of the data is used for testing. Different selection procedures used the same collections of the training and test sets. The averaged test errors are shown in the figures. We do not present the resulting standard deviations to maintain the clarity of the plots. In general, we found that the standard deviations vary between 3% and 7% of the averaged errors.

Fig. 9.10 presents the results for the two dissimilarity measures derived from the same set of polygons. Remember that the *Polydisth* is metric and *Polydistm* is not. The first striking observation is that in spite of its non-metric behavior, the *Polydistm* results are better: lower NN errors, less prototypes needed to yield a good result. Just 20 prototypes out of 1000 objects are needed to obtain

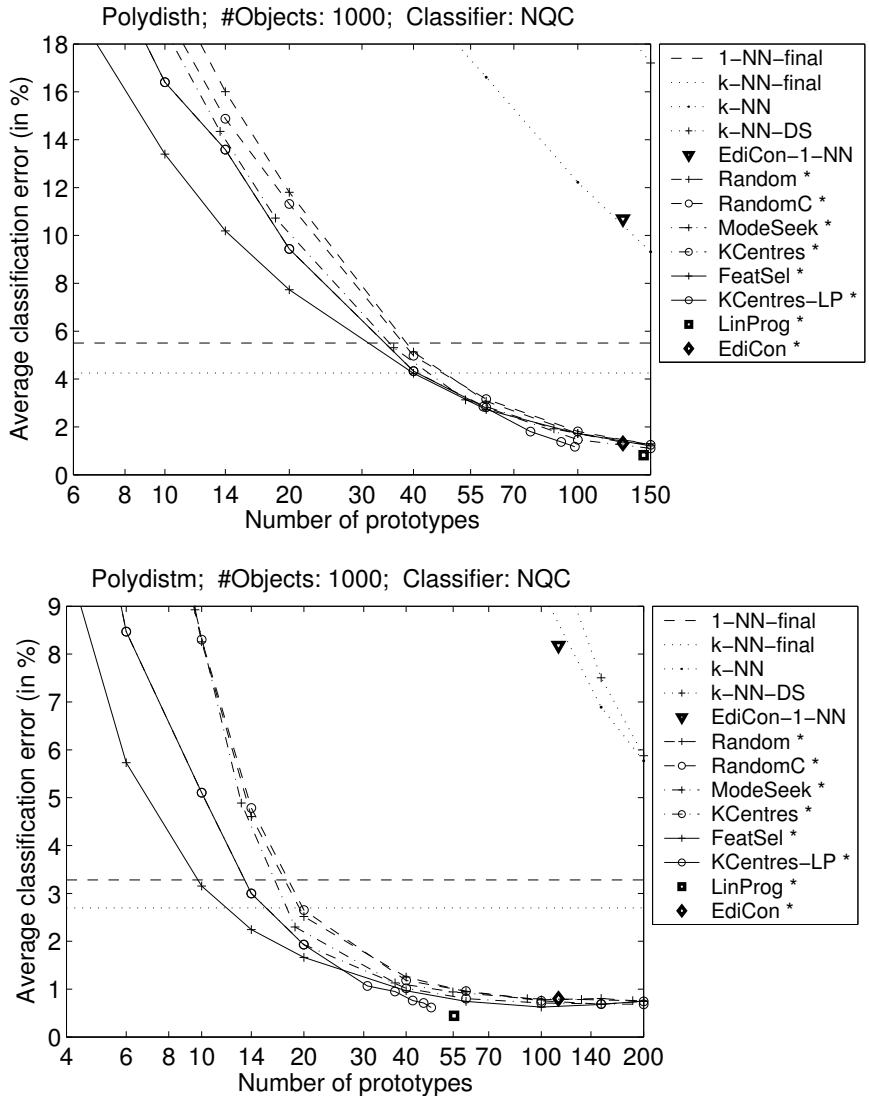


Fig. 9.10: Polygon data. Average classification error of the NQC* and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes.

a better error than found by the NN rules. In the k-NN classifiers, the average optimal k appeared to be 127 (*Polydisth*) or 194 (*Polydistm*). These large values correspond to the observation made before in relation to the scatter plots (Fig. 9.4) that this is a difficult data set. Nevertheless, in the case of the *Polydistm* data, the linear programming technique finds a small set of 55 prototypes for which the NQC error is very low (0.4%). The systematic procedures KCentres (KCentres-LP) and FeatSel perform significantly better than the other ones. The feature selection is also optimal for small representation sets. Notice also the large difference between the two results for editing and condensing. They are based on the same sets of prototypes, but the classification error of the 1-NN rule (in fact a nearest prototype rule), EdiCon-1-NN, is much worse than of the NQC, the EdiCon, which is trained on $D(T, R)$. This also remains true for all considered problems, as can be observed in other plots.

Fig. 9.11 shows the results for two of the NIST digit classification problems. The *NIST-38* data set is based on a Euclidean distance measure, while the *Zongker-12* relies on a non-metric shape comparison. The k-NN classifier does not improve over the 1-NN rule, indicating that the data set sizes (100 objects per class) are too small to model the digit variabilities properly. Again. the systematic procedures do well for small representation sets, but they are outperformed by the KCentres routine for a larger number of prototypes. The KCentres method distributes the prototypes evenly over the

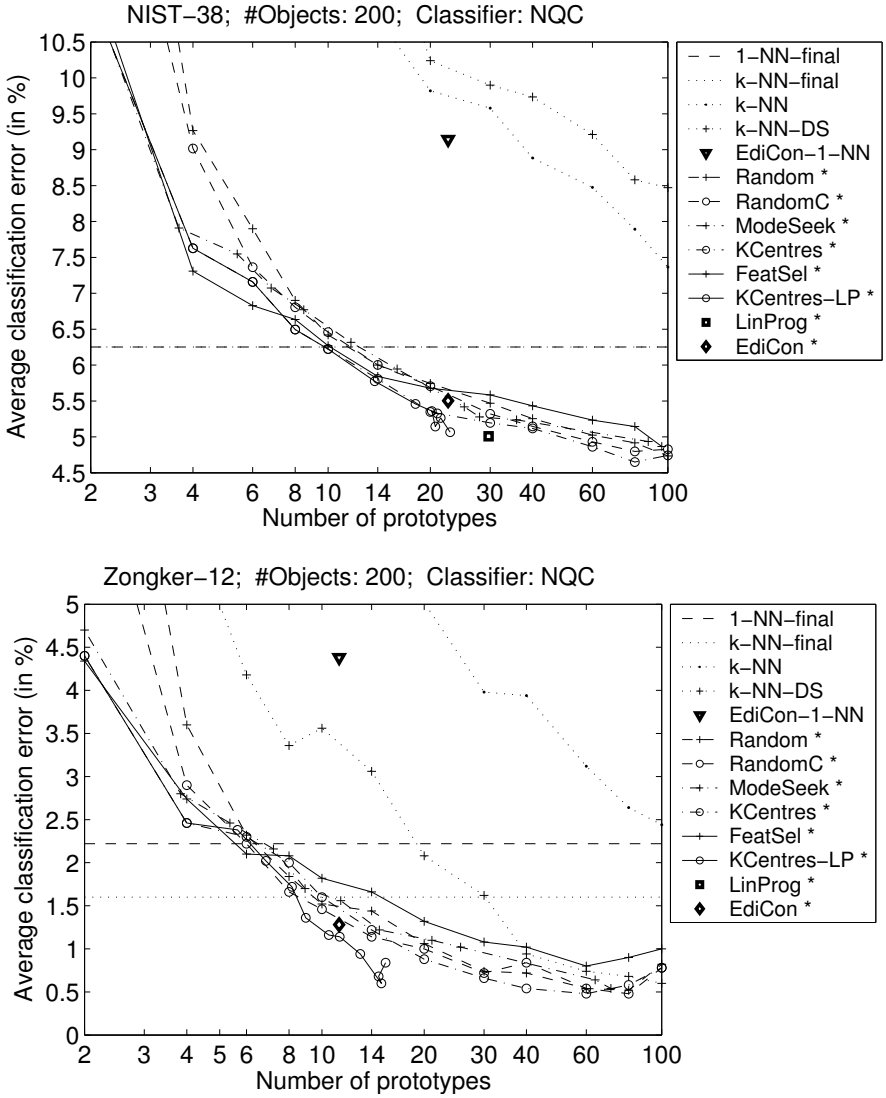


Fig. 9.11: NIST digit data. Average classification error of the NQC* and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes.

classes in a spatial way, that is related to the dissimilarity information. For small training sets (here 100 examples per class), this may be a better than an advanced optimization.

Fig. 9.12 presents the results for the two dissimilarity representations of the geophysical data sets. From other experiments it is known that they are highly multi-modal, which may explain the good performance of the ModeSeek for the *GeoShape* problem and the KCentres for the *GeoSam* problem. Editing and condensing does also relatively well. Feature selection works also well for a small number of prototypes. Overall, the linear programming yields good results. Recall that we take the KCentres results as a start (except from the final result indicated by the square marker that starts from the entire training set), so the KCentres curve is for lower numbers of prototypes underneath it. In this problem we can hardly improve over the NN performance, but still need just $5\% - 10\%$ of the training set size for prototypes. In the next subsection, however, it is shown that these results can still be significantly improved by modifying the dissimilarity measure.

So far, we have focused on two-class problems. In illustrate what may happen in multi-class situations, the following problems are also considered: the three-class *Wine* and *Ecoli* data, the four-class *ProDom* data and the ten-digit *Zongker-all* data. Although the *Wine* and *Ecoli* data are originally represented by features, their l_p distance representations can be used to show our point. In all the ex-

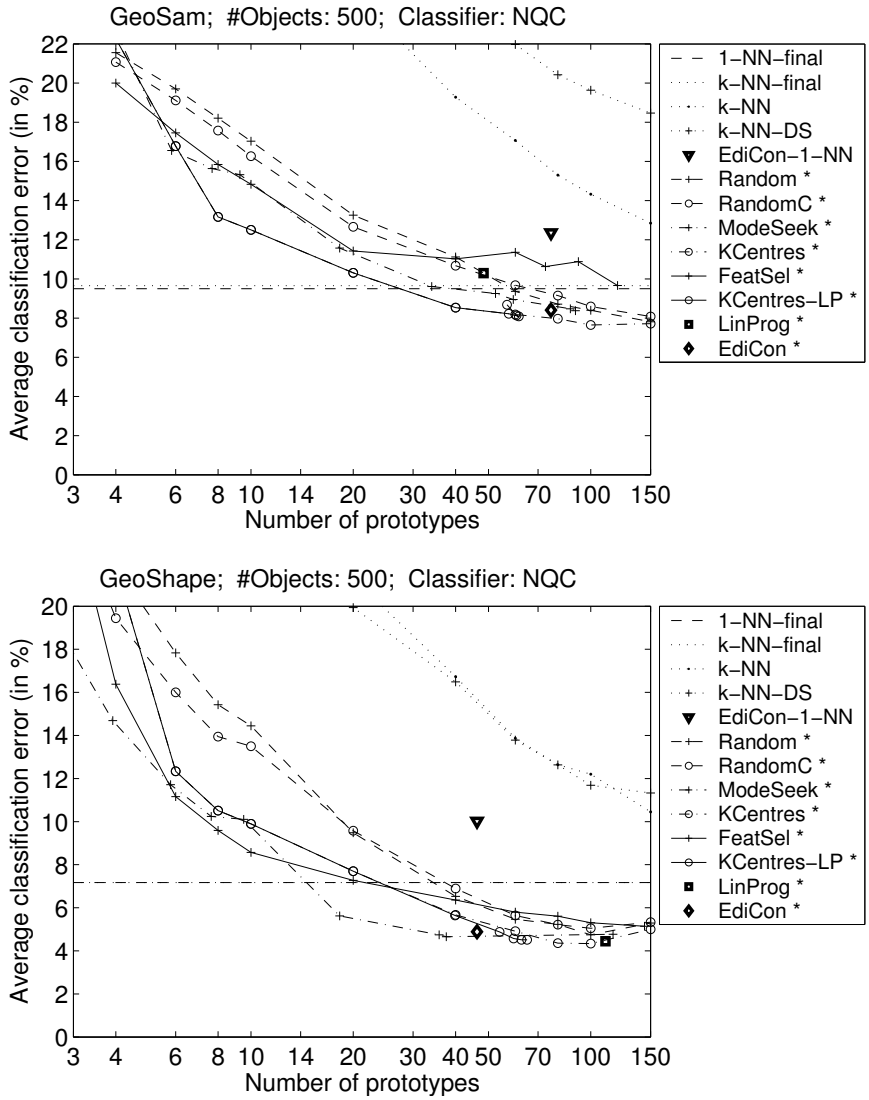


Fig. 9.12: Geophysical data. Average classification error of the NQC* and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes.

periments with the NQC, a small regularization is used $\lambda = 0.01$; see section 4.4.1. A regularization is necessary since for large representation sets, the number of training objects per class is insufficient for a proper estimation of the class covariance matrices. For instance, 100 training examples per class are used for the *Zongker-all* data. The results for R with more than 100 prototypes are based on the NQC trained in more than 100 dimensions. The peak for exactly 100 prototypes, see Fig. 9.16, upper plot, is caused by a dimension resonance phenomenon that has been fully examined for the linear normal density based classifier in [314]. When a larger regularization is used in this case, the NQC performs much better, as observed in the bottom plot of the same figure.

Fig. 9.13 shows the results for the Euclidean representation of the *Wine* data. The ModeSeek seems to work the best, however since the number of test objects is small (70 in total), all the selection procedures behave similarly for more than 10 prototypes. The latter observation also holds for the *Ecoli-p08* data, as observed in Fig. 9.14. The number of test objects is also small (107 in total). Here, however, the NQC does not improve over the k-NN on the complete training set. Still, 20 (or less) prototypes are needed for the same performance.

Fig. 9.15 illustrates the study on prototype selection for the *ProDom* data. The data are multi-modal,

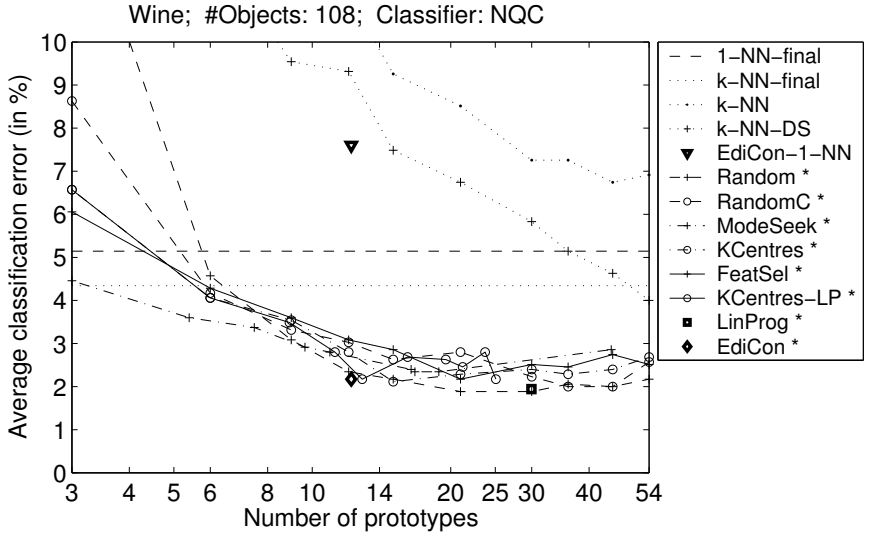


Fig. 9.13: Wine data. Average classification error of the NQC* and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes.

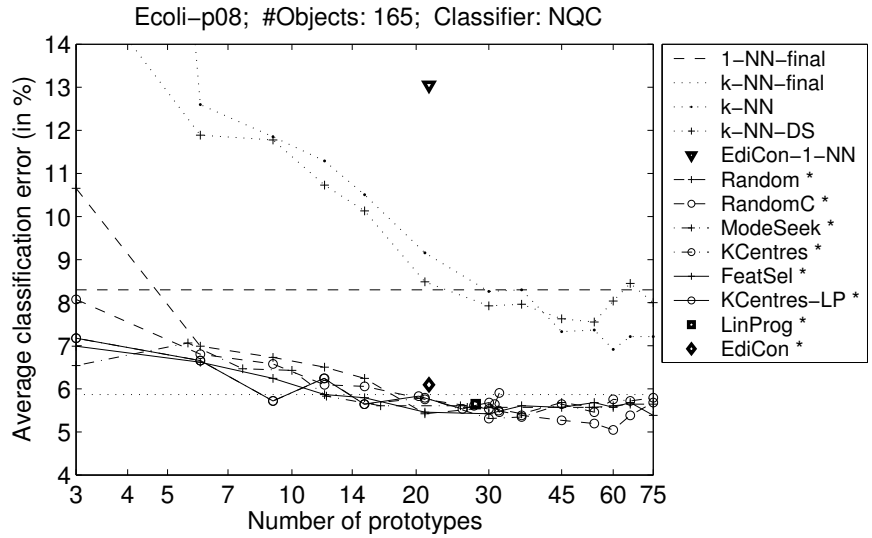


Fig. 9.14: *Ecoli-p08* data. Average classification error of the NQC* and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes.

as it can be judged from the 2D approximate embedding shown in Fig. 9.9. Some of the modes in the data seem to be very small, possibly some outliers. This may cause the *ModeSeek* procedure to focus on such examples, and be worse than the class-wise random selection. The KCentres and the FeatSel methods perform the best. For 100 (an more) prototypes, the NQC reaches the error of the k-NN on a complete training set, however, it does not improve it. This might be partly caused by unequal class cardinalities and too-small regularization parameter.

The *Zongker-all* data are highly non-Euclidean and non-metric. When a proper regularization ($\lambda = 0.05$) is used, the NQC significantly outperforms the best k-NN rule. However, when the size of the representation set is too large (450 prototypes in bottom plot), the NQC starts to suffer. Only 3% of the training examples allow this decision rule to reach the same performance as the k-NN rule on the entire training set. In general, the KCentres works the best. Edited and condensed set seems to give a good representation set, as well.

Some observations are of interest for multi-class problems. First, in contrast to the two-class prob-

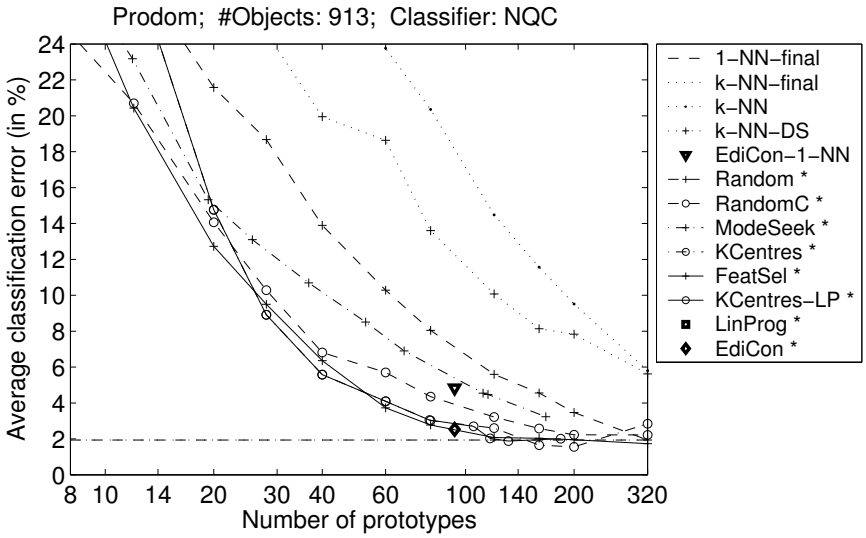


Fig. 9.15: Four-class *ProDom* problem. Average classification error of the NQC* and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes. The result for the *LinProg* is not visible, since it finds a representation set of 491 objects.

lems, a suitable regularization is necessary, since it can significantly influence the performance of the NQC. If the regularization is appropriate, a significant improvement over the k-NN results on the complete training set may be found by the use of a regularized NQC. Next, as in the two-class problems we find that just $3\% - 12\%$ of the training set gives a sufficient number of prototypes for the NQC to reach the same performance as the k-NN rule. Like before, systematic selections of prototypes perform best. Finally, the EdiCon works well and tends to determine less prototypes than the LinProg.

In summary, we see that systematic selections perform better than the random selection, but the differences are sometimes small. The way we have ranked the algorithms in the legends from the Random to the KCentres-LP selections, roughly corresponds to the way they globally perform over the set of conducted experiments.

Concave transformations of dissimilarity representations. Concave transformations of dissimilarity representations may improve the discrimination properties between the classes, when linear or quadratic classifiers are used in dissimilarity spaces. An example can be given by the sigmoidal transformation $f_{\text{sigm}}(x) = 2/(1+exp(-x^2/s^2))-1$ applied to the square dissimilarities in an elementwise way. The transformed representation becomes then $D_{\text{sigm}} = (f_{\text{sigm}}(d_{ij}^2))$. A nonlinear transformation is applied to square dissimilarities, which significantly changes the original dissimilarities. Note that the sigmoidal transformation is monotonically increasing, so the k-NN rule behaves identically as for the original dissimilarities.

To illustrate possible benefits of a sigmoidal transformation, an experiment for the *GeoSam* representation has been performed for a fixed number of $K = 20$ and $K = 60$ prototypes. From Fig. 9.12, top row, we can observe that for 20 prototypes, the best average performance of the NQC is approximately 10%. When a suitable parameter s of the sigmoidal transformation is chosen, the best average performance of the same classifier is 6%, which can be improved to 4% when 60 prototypes are considered. This can be observed in Fig. 9.17. So, the gain in performance is significant. The k-NN error based on the entire training set T of 500 objects (hence tested on $D(T_{te}, T)$) is 9.6%. Note, however, that such nonlinear transformations do not immediately guarantee the improved performance. It is simply related to the discriminative properties of the dissimilarity measure used.

The parameter s has been investigated in the range of $[0.5d_{me}, 10d_{me}]$, where d_{me} is the average

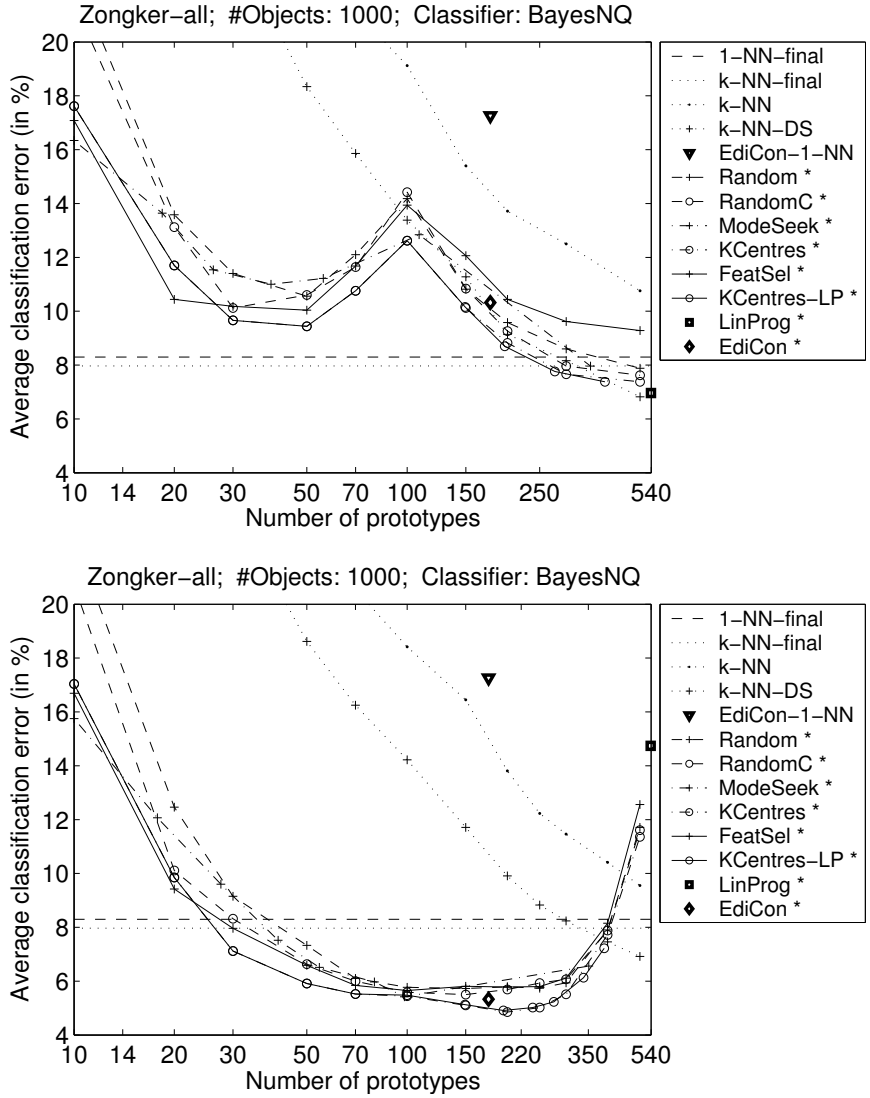


Fig. 9.16: Ten-class *Zongker* problem. Average classification error of the NQC, with the regularization of $\lambda = 0.01$ (upper plot) and $\lambda = 0.05$ (bottom plot), and the k-NN classifiers in dissimilarity spaces, as well as the direct k-NN as a function of the selected prototypes.

distance of the original representation $D(T,T)$. The best classification accuracy is reached for $s \approx 3\,d_{me}$. It may be observed, however, that a specific choice of s is not very crucial. For a range of possible values of s, a significant performance improvement is achieved compared to the original representation. The NQC defined on the representation set R determined by the KCentres algorithm performs somewhat worse than in the case of a randomly selected R.

The interesting point is that the transformed dissimilarity representations are strongly non-metric and non-Euclidean. When s is very small, however, then D_{sigm} is nearly metric and nearly Euclidean. For $s \in [d_{me}, 4d_{me}]$, on average 70.8% of triangle inequalities are disobeyed. The deviation of the Euclidean behavior is on average $r_{rel}^{nE} = 28.2$ and $r_{mm}^{nE} = 30.5$, which suggests large negative eigenvalues of the corresponding Gram matrices.

Conclusions. Prototype selection is an important topic for dissimilarity-based classification. By using a few, but well chosen prototypes, it is possible to achieve a better classification performance in both speed and accuracy than by using all the training samples. Usually, prototype selection methods are investigated in the context of the metric k-NN classification considered for feature-based representations. In our proposal, a dissimilarity representation $D(T,T)$ is interpreted as a vector

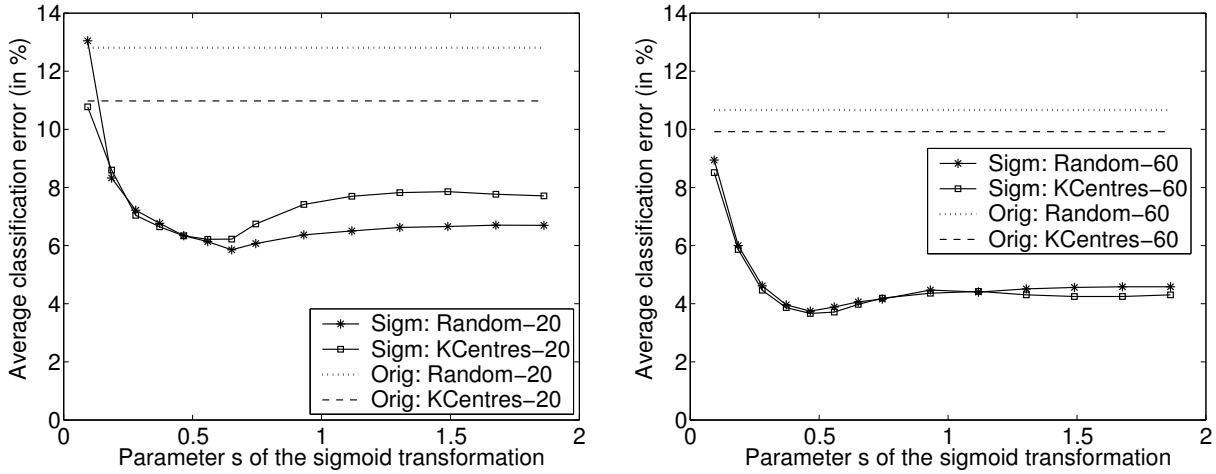


Fig. 9.17: *GeoSam:* classification error (averaged over 25 runs) of the NQC in a dissimilarity space based on 20 prototypes (left) and 60 prototypes (right) chosen either randomly or by the KCentres algorithm. The prototypes are selected for both the original dissimilarity representation (Orig) and its sigmoidal transformation (Sigm) as a function of the parameter s. The horizontal lines correspond to the classification errors for the original representations based either on 20 or 60 prototypes. The standard deviations of the means are less than 0.5%. The k-NN error defined on the training set T of 500 examples, i.e. derived from $D(T_{te}, T)$ is 9.6% for the *GeoSam*. Since the sigmoidal transformation is monotonic, the k-NN results remain unchanged.

space, where each dimension corresponds to a dissimilarity to an object from T. This allows us to construct traditional decision rules, such as linear or quadratic classifiers on such representations. Hence, the prototype selection relies on the selection of the representation set $R \subset T$ such that the chosen classifier performs well in a dissimilarity space $D(\cdot, R)$. Since the classifier is then trained on $D(T, R)$, a better accuracy can be reached than by using the k-NN rule defined on the set R.

Various random and systematic selection procedures have been empirically investigated for the normal density based quadratic classier (NQC) built in dissimilarity spaces. The k-NN method, defined both on a complete training set T and a representation set R is used as a reference.

The following conclusions can be made from our study with respect to the investigated data sets:

1. By building the NQC in dissimilarity spaces just a very small number of prototypes (such as 3% - 12% of the training set size) is needed to obtain a similar performance as the k-NN rule on the entire training set.
2. For large representation sets, consisting of, for instance 20% of the training examples, significantly better classification results are obtained for the NQC than for the best k-NN. This holds for two-class problems and not necessarily for multi-class problems, unless a suitable regularization parameter is found.
3. Overall, a systematic selection of prototypes does better than a random selection. Concerning the procedures which have a control over the number of selected prototypes, the KCentres procedure performs well, in general. In other cases, the linear programming performs well for two-class problems, while editing and condensing sets should be preferred for multi-class problems.

In our investigation, multi-class problems are more difficult as they need a proper regularization for the NQC discrimination function. Moreover, this classifier becomes computationally more expensive. Therefore, there is a need for a further research to study more suitable classifiers and other prototype selection techniques for multi-class problems.

9.3 Selection of the representation set: the embedding approach

In the embedding approach, one considers an embedding of the symmetric dissimilarity data $D(T,T)$ into a k-dimensional pseudo-Euclidean space $\mathcal{E} := \mathbb{R}^{(p,q)}$, $k = p+q$ such that the original dissimilarities are perfectly preserved. However, many dimensions can turn out to be non-informative since the variance in the data are close to zero. The variances of the projected data are specified by the eigenvalues derived in the embedding; see sections 3.3.3 - 3.3.6 for details. In fact, one determines the dimensionality $m = p' + q'$ based on eigenvalues which are significantly different from zero. The remaining $k-m$ dimensions are simply neglected as corresponding to noise and non-significant information. If m is much smaller than $N = |T|$, then the question arises whether N objects are necessary to determine the m-dimensional space. In fact, only $m+1$ objects can define a linear space: one object will serve as a reference to the origin and m objects will correspond to the basis vectors. This is computationally attractive, since only dissimilarities to these $m+1$ objects need to be computed. The task can now be formulated as follows. Given the representation X in $\mathbb{R}^{(p,q)}$ that preserves the original dissimilarities, choose the representation set R of $m+1$ objects such that the projection defined by R, (hence the space defined by $D(R,R)$ with the remaining $T \backslash R$ objects projected later to this space) gives a configuration which is close to X (according to some criterion). A set R, spanning the space $\mathbb{R}^m = \mathbb{R}^{(p,q)}$ such that $\mathbb{R}^m$ is defined by m leading principal axes, might not, however, exist. To avoid an intractable search over all possible subsets, an error measure between the approximated and original configurations can be defined to be minimized, e.g. in a greedy approach [409]. Here, our ultimate goal, however, is not the best approximation of the given configuration X, but, good classification results in an embedded space. In fact, R should be chosen such that the discrimination between the classes is preserved or even improved. The following procedures are considered for the selection of R: Random, KCentres, MaxProj, APE, LAE, Pivot objects and NLC-err, as explained below.

Random. $m+1$ objects are randomly chosen from all training objects.

KCentres. $m+1$ center objects are chosen such that they minimize the maximum of the dissimilarities over all training objects to their nearest neighbors; see also section 9.2.1.

Note that the two procedures above do not guarantee a faithful representation of the originally embedded X. The procedures below focus more on this aspect. We start our reasoning from X, whose mean vector coincides with the origin. To simplify the approach, the origin of the embedded space will now be fixed to the projection of the object p_0 which is the closest to the origin. Such an object is easily detected as the one whose average square dissimilarity to T is the smallest [152, 293, 301]. Having determined p_0, the entire configuration X is shifted to the new origin. So, since now on, X refers to a shifted configuration. Starting from p_0, objects are now successively added in each step until $m+1$ objects are found. In each step, an object is selected that minimizes a specified criterion. This does not guarantee the overall optimal solution, however, it guarantees the best immediate solution.

Let $R_0 = \{p_0\}$ and let R_{j-1} be the representation set after the $(j-1)$-th step.To assure that the chosen objects are linearly independent and to make the selection a feasible process, in the j-th step, only M objects $Z^j = \{z_1^j, ..., z_M^j\} \subset T \backslash R_{j-1}$ with the largest (in magnitude) projections on the j-th principal axis are pre-selected to be tested against the specified criterion. M is assigned to e.g. 10% of the training size. This holds for all criteria introduced below.

MaxProj. In each step, this criterion chooses an object yielding the largest (in magnitude) projection on the j-th dimension.

Average Projection Error (APE). Let $\mathcal{E}_{j-1}$ be a j-dimensional subspace of the complete embedded space $\mathcal{E} := \mathbb{R}^{(p,q)}$ $(p+q=k)$, where $\mathcal{E}_{j-1}$ is determined by $R_{j-1} = \{p_0, p_1, \ldots, p_{j-1}\}$. Based on the

properties of the inner products and the embedding, and given that p_0 is projected as $\mathbf{x}_1$ at the origin, the square pseudo-Euclidean distance between a vector $\mathbf{x}_i \in \mathbb{R}^k$ and its projection $\mathbf{x}_i^{\mathcal{E}_{j-1}}$ onto $\mathcal{E}_{j-1}$, the approximation error can be expressed as:

$$e_{\text{apr}}(\mathbf{x}_i) = ||\mathbf{x}_i - \mathbf{x}_i^{\mathcal{E}_j}||^2_{\mathcal{E}} = ||\mathbf{x}_i||^2_{\mathcal{E}} - ||\mathbf{x}_i^{\mathcal{E}_j}||^2_{\mathcal{E}_j} = d^2(p_i, p_0) - (\mathbf{g}_{.i}^{(n)})^T \, G^{-1} \, \mathbf{g}_{.i}^{(n)}, \tag{9.2}$$

where $\mathbf{g}_{i.}^{(n)}$ is the i-th column of the cross-Gram matrix $G^{(n)}$ and G is the Gram matrix, where both G and $G^{(n)}$ refer to the representations in $\mathcal{E}_j$ defined by pairwise dissimilarities between $j+1$ objects (i.e. the origin and the basis)[7]. Having chosen the set $R_{j-1} = \{p_0, p_1, ..., p_{j-1}\}$, in the j-th step, an object $z \in Z^j$ is selected as p_j such that the average projection error $\sum_{\mathbf{x}_i \in T} e_{\text{apr}}(\mathbf{x}_i)$ onto the space $\mathcal{E}_j$, defined by $\{R_{j-1}, z\}$ (hence $\mathcal{E}_j$ is determined by projecting $D([R_{j-1}, z], [R_{j-1}, z])$) is the smallest.

Largest Approximation Error (LAE). Having chosen the set $R_{j-1} = \{p_0, p_1, ..., p_{j-1}\}$, in the j-th step, an object $z \in Z^j$ is selected as p_j as the one which yields the largest approximation error (9.2) of z onto the space $\mathcal{E}^{j-1}$, defined by R_{j-1}. Since in the first step, the inner products cannot be defined yet, $e_{\text{apr}}(\mathbf{x}_i)$ is assumed to be equal to $d^2(p_i, p_0)$, where p_0 is the object closest to the origin in the embedded space, as described before.

NLC-err. Starting from $R = \{p_0\}$, in the j-th step, an object $z \in Z^j$ is selected as p_j as the one for which the embedded configuration X_j of $D(T, R_j)$ allows for reaching the smallest 5-fold cross-validation error of the NLC (or other chosen classifier). In case of ties, an object with the largest projection on the j-th axis is chosen.

Pivots. Choose $m/2$ times two pivot objects as described in the FastMap algorithm in section 3.4.1.

The above criteria select the representation set R as appropriately defined by R_m. Their results can be judged by various measures. For instance, to see how much distortion was introduced by the approximation step (hence the selection of R), the mean square error between the original and approximated dissimilarities can be computed. Another possibility is the computation of the average between-class square distance to the average within-class square distance, again on both original and approximated dissimilarities. It gives an indication on the class separability. Since, in fact, our purpose is the classification task, it is not crucial that the distances are well preserved when the classification performance is good. For this reason, we focus on the resulting classification error.

9.3.1 Experiments and results

Most of the data sets that are used in our study are the one analyzed for prototype selection methods in the dissimilarity space approach; see section 9.2.2. The experiments are performed $M = 25$ times for two-class data and $M = 10$ times for multi-class data, and the results are averaged. In each run, data sets are randomly split into the training and test sets, as indicated in Table 9.2. In each experiment, $m+1$ prototypes are either directly selected by the Random or the KCentres approaches,

[7] Given a symmetric matrix $D(R, R)$, a linear embedding into $\mathcal{E} := \mathbb{R}^m = \mathbb{R}^{(p',q')}$ can be constructed such that the origin coincides with the vector representation of e.g. $\mathbf{x}_1$. Since by our assumption $||\mathbf{x}_i||^2_{\mathcal{E}} = ||\mathbf{x}_i - \mathbf{0}||^2_{\mathcal{E}} = d^2_{\mathcal{E}}(\mathbf{x}_i, \mathbf{x}_1) = d^2(p_i, p_0)$ holds, then the Gram matrix (a matrix of inner products) $G = \{g_{ij}\}$ for the vector representation $\{\mathbf{x}_1, ..., \mathbf{x}_n\} \in \mathcal{E}$ is expressed by using the pseudo-Euclidean distances as $g_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle = -\frac{1}{2}\left[d^2(\mathbf{x}_i, \mathbf{x}_j) - d^2(\mathbf{x}_i, \mathbf{x}_1) - d^2(\mathbf{x}_j, \mathbf{x}_1)\right]$. By the eigendecomposition of $G = Q\Lambda Q^T = (Q|\Lambda|^{1/2})\mathcal{J}_{pq}(Q|\Lambda|^{1/2})^T$, X can be represented in the space $\mathcal{E}$ as $X = Q_m|\Lambda_m|^{1/2}$, where m reflects the number of eigenvalues, significantly different from zero. Novel objects $D^{(n)} := D(T^{(n)}, R)$ are then orthogonally projected onto $\mathcal{E}$ as $X^{(n)}$. Based on the matrix of inner products $G^{(n)} = \{g_{ij}^{(n)}\}$ consisting of $g_{ij}^{(n)} = -\frac{1}{2}[d^2(\mathbf{x}_i^{(n)}, \mathbf{x}_j) - d^2(\mathbf{x}_i, \mathbf{x}_1) - d^2(\mathbf{x}_j^{(n)}, \mathbf{x}_1)]$, $X^{(n)}$ is given by $X^{(n)} = G^{(n)}X|\Lambda|^{-1}\mathcal{J}_{pq}$ or $X^{(n)} = G^{(n)}G^{-1}X$. This is similar the projection presented in section 3.3 with the difference that a specified object is mapped to the origin.

Table 9.4: Dissimilarity data sets used in the experiments. T and T_{te} correspond to the training and test sets, respectively. $|\cdot|$ stands for the set cardinality. A rough estimation of the effective intrinsic dimensionality ID relies on the number of significant eigenvalues in the embedding of $D(T,T)$, while ID refers to the number of indicative dimensions, in general.

Data	Dissimilarity	Property	Effective ID	ID
Polydisth	Hausdorff	M, nE	18	60
Polydistm	Mod. Hausdorff	nM	13	50
NIST-38	Euclidean	E	7	20
Zongker-12	Template-match	nM	6	15
GeoSam	SAM [239]	M,nE	5	8
GeoShape	Shape l_1	M, nE	6	10
ProDom	Structural	nM	18	80
Zongker-all	Template-match	nM	10	80

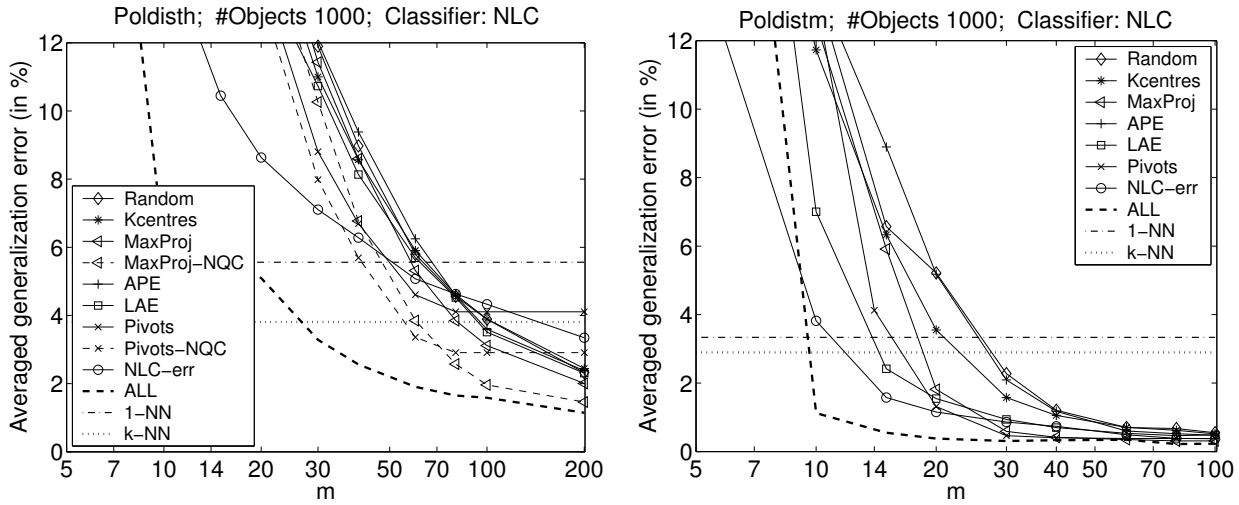


Fig. 9.18: Polygon data: classification error (averaged over 25 runs) of the NLC in an m-dimensional embedded space as a function of m for the *Polydisth* data (left) and the *Polydistm* data (right). Except for 'ALL', other criteria choose a representation set R of $m+1$ objects, which serves for the determination of an embedded space and training the NLC there. 'ALL' stands for the NLC results, where the m-dimensional embedded space is found by using *all* training objects. The NN results are based on $R:=T$, i.e. 600 objects and are given as a reference. For the *Polydisth* data, the error curve corresponding to the random selection is not visible, since it lies above the given scale. Additionally, also the classification error of the NQC for the *Polydisth* is shown for R chosen as pivot objects or by MaxProj criterion.

or based on the dissimilarity matrix $D(T,T)$. First the complete k-dimensional representation of $D(T,T)$ is found and then the set R of $m+1$ objects is chosen according to some specified criterion. Next, the approximated space, defined by objects from R is determined (i.e. the mapping based on $D(R,R)$ only), where additional $T\backslash R$ objects are projected. The NLC (equivalent to the FLD for two equally probable classes) is then trained both in the reduced and approximated spaces and the generalization error is computed for the test set. Here, we have decided for a fixed and simple classifier, the NLC, although, in some cases it is not the best choice. As a reference, the results of the 1-NN and the best k-NN rule on the entire training set T, i.e. determined by $D(T_{te},T)$, are provided.

The results of our experiments are presented in Fig. 9.19 - 9.22. The standard deviations are not shown there to maintain the clarity of the plots. In general, the standard deviations vary from 2% to 7% of the averaged classification errors. The number m of important dimensions (hence an

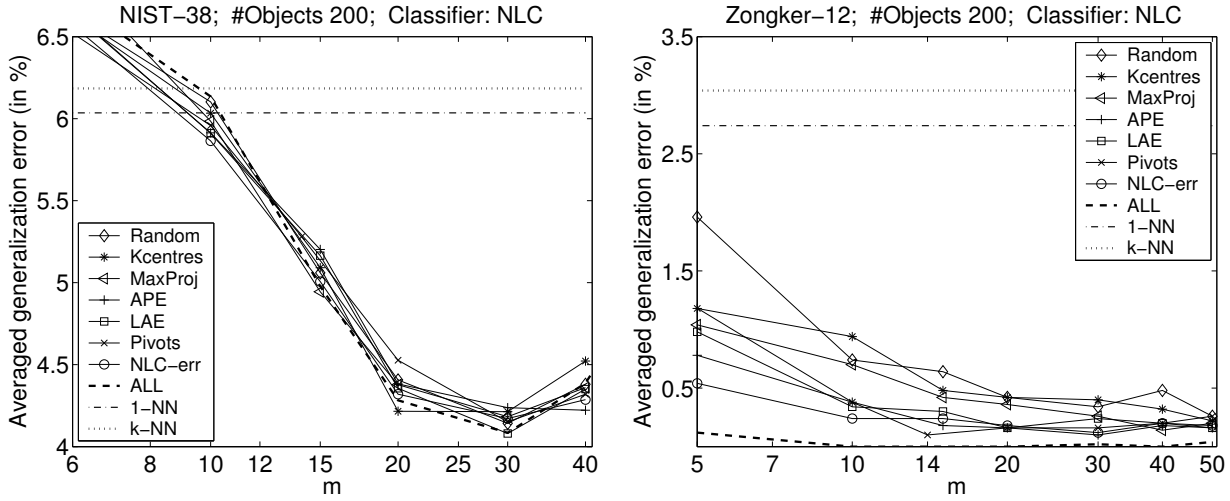


Fig. 9.19: NIST data: classification error (averaged over 25 runs) of the NLC in the m-dimensional embedded space as a function of m for the *NIST-38* data (left) and the *Zongker-12* data (right). All, but 'ALL' criteria choose a representation set R of $m+1$ objects, which serves for the determination of an embedded space and training the NLC there. 'ALL' stands for the NLC results, where the m-dimensional embedded space is found by using *all* training objects. The NN results are based on $R := T$, i.e. 200 objects and are given as a reference. Note that scale differences.

indication on the cardinality of the set R, since $|R| = m+1$) is related to complexity of the given classification problem. This is somewhat related to the intrinsic dimensionality. As observed in Fig. 9.4 -9.6, every dissimilarity problem has a different intrinsic dimensionality m (determined by significant eigenvalues in the embedding). By a visual judgment, the estimations can be made; see Table 9.4. So, ideally, our selected representation set R could consist of $m+1$ objects. This, however, might not be sufficient, simply, because an approximation is made by using only the set R (instead of T) to define an embedded space. Moreover, the additional difficulty arises when the classes are not linearly separable. If the linear classifier is not adequate for the embedded configuration (because the boundary is e.g. quadratic), the classification error might be large. So, the choice of the representation set as well as the discrimination function plays a significant role in solving the classification task for the given R. In our study, the NLC has been selected, which might not be optimal.

As observed before in Fig. 9.4 - 9.9, the following observations are important for the embedded space approaches:

- Both *Polydisth* and *Polydistm* dissimilarity data are strongly non-Euclidean. The intrinsic dimensionality is smaller for the *Polydistm* than for the *Polydisth*. Also, as judged from the 2D spatial maps, the classes for these problems are overlapping (in a 2-dimensional approximate embedding space), yet, they are more compact for the *Polydistm* than for the *Polydisth*. For the *Polydisth* embedding, the classes may seem to be uniformly distributed.
- The NIST digits appear to be linearly separable as shown in Fig. 9.5 for the 2D approximate embeddings. The intrinsic dimensionality is small for the *NIST-38* case, while larger for the *Zongker-12* data.
- The multi-modality of the geophysical data can be observed in cluster tendencies that are visible for the 2D approximate embeddings. Both sets seem to have a low intrinsic dimensionality.
- The *ProDom* data are nearly Euclidean.

Concerning the classification performance in embedded spaces, as observed in Fig. 9.18, the *Polydisth* problem is more difficult than the *Polydistm* problem. Indeed, the *Polydistm* classes are linearly

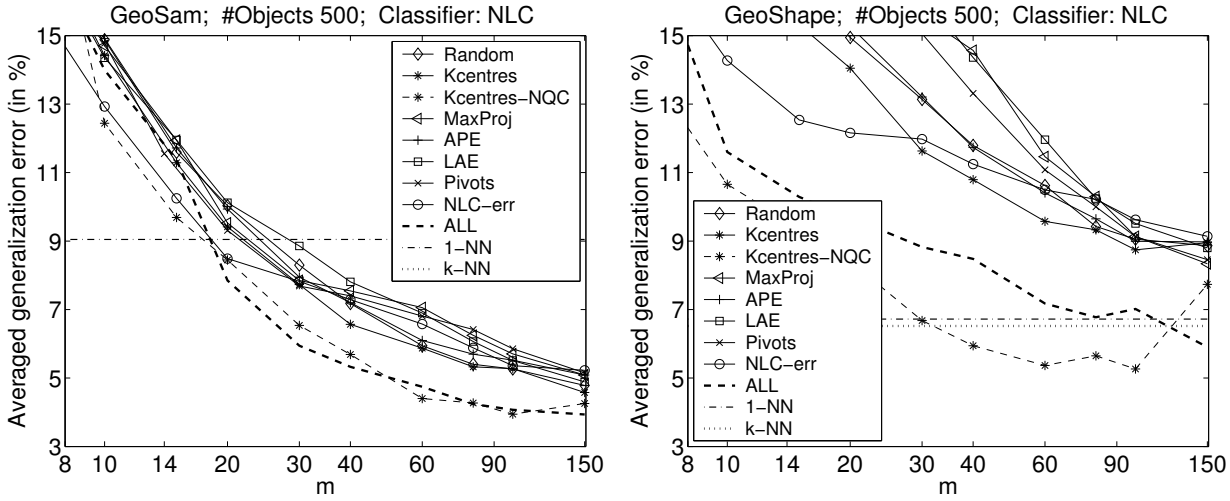


Fig. 9.20: Geophysical data: classification error (averaged over 25 runs) of the NLC in the m-dimensional embedded space as a function of m for the *GeoSam* data (left) and the *GeoShape* data (right). All, but 'ALL' criteria choose a representation set R of $m+1$ objects, which serves for the determination of an embedded space and training the NLC there. 'ALL' stands for the NLC results, where the m-dimensional embedded space is found by using *all* training objects. The NN results are based on $R := T$, i.e. 500 objects and are given as a reference. Additionally, also the performance of the NQC is shown for the *GeoShape* and R chosen by the KCentres procedure.

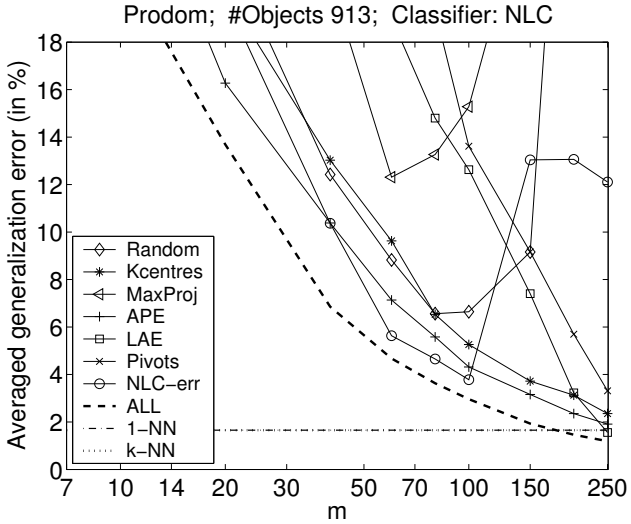


Fig. 9.21: Four-class *Prodom* data: classification error (averaged over 25 runs) performance of the NLC in the m-dimensional embedded space as a function of m. All, but 'ALL' criteria choose a representation set R of $m+1$ objects, which serves for the determination of an embedded space and training the NLC there. 'ALL' stands for the NLC results, where the m-dimensional embedded space is found by using *all* training objects. The NN results are based on $R := T$, i.e. 913 objects and are given as a reference. The lack of a proper regularization in the NLC makes some of the error curves grow up.

separable and the effective intrinsic dimensionality is small. The NLC based on all objects gives nearly a zero-error. The same can be achieved for 41 prototypes in the representation set R. Only 14-20 objects in the set R, chosen in some systematic way, allow the NLC to perform better than the best k-NN rule defined on $R = 600$ objects.

Since there is a 'big gap' between the NLC error curve in an embedded space defined on all training examples and the NLC error curves in an embedded space determined by a number of prototypes only, we tend to think that the NLC might be not the most suitable classifier for the problem.

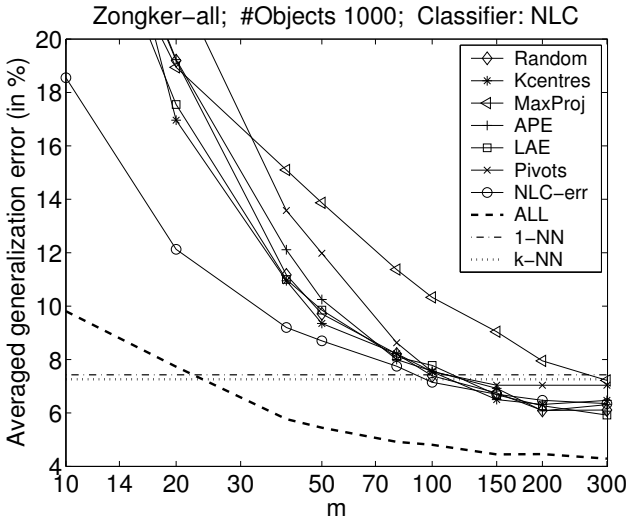


Fig. 9.22: Ten-class *Zongker* data: classification error (averaged over 25 runs) performance of the NLC in the m-dimensional embedded space as a function of m. All, but 'ALL' criteria choose a representation set R of $m+1$ objects, which serves for the determination of an embedded space and training the NLC there. 'ALL' stands for the NLC results, where the m-dimensional embedded space is found by using *all* training objects. The NN results are based on $R{:=}T$, i.e. 1000 objects and are given as a reference.

Additionally, the NQC error curve is presented for the MaxProj and Pivots selection criteria of R (they correspond to the best results). The generalization error decreases, however it does not improve over the NLC result found in an embedded space defined by all objects. The representation set R of 70 objects chosen by the Pivots or by the MaxProj method allows the NLC to reach a similar performance as the k-NN based on all training objects. Note also that the NLC-err criterion should be preferred selects for small representation sets, however.

As observed in Fig. 9.19, the NLC in an embedded space defined by 10 prototypes for the *NIST-38* data and defined by 5 prototypes for the *Zongker-12* data outperforms the best k-NN defined on all (200) training objects. The *Zongker-12* problem is linearly separable and the NLC defined on $|R| = 20$ objects reaches a nearly zero error for the Pivots and the LAE selection methods. The prototype selection procedures also seem to work well to fit the *NIST-38* data, since both systematic and random approaches allow one to reach an accuracy close to the one reached by the NLC in an embedded space based on all training examples.

From our earlier observations, we already know that the geophysical data are multi-modal. This means that a linear classifier in an embedded space will not fit the problem well. Yet, as observed in Fig. 9.20, the classes can be reasonably separated for the *GeoSam* problem. The representation set R of 30 examples defines an embedded space such that the NLC constructed there outperforms the best k-NN rule based on 500 training objects. However, for the *GeoShape* problem, the NLC performs much worse than for the *GeoSam*. In fact, the NLC does not outperform the best 1-NN rule. This becomes, however, possible, when a quadratic classifier is used (see Fig. 9.20, right) for the KCentres criterion.

In four-class *Prodom* problem, Fig. 9.21, some error curves grow with the increasing m. This is the side-effect of the lack of proper regularization in the NLC. The KCentres and the APE criteria seem to work well, however, in this case, the k-NN rule based on all training examples is the best.

Concerning the ten-class *Zongker* problem, Fig. 9.22, at least 120 objects in the representation set R are needed such that the NLC in an embedded space defined on $D(T, R)$ outperforms the k-NN.

All in all, there is no single selection method that works the best for all m (which is also the size of the representation sets). For small representation sets, the NLC-err, the supervised selection based

on the cross-validation NLC error in an embedded space is always the best. This is not surprising, since an embedded space is chosen to guarantee the best NLC performance. However, for larger representation sets, this method may become significantly worse than the other systematic selection procedures. The KCentres approach seems to be good for multi-modal problems (the *GeoSam*, the *GeoShape* and the the-class *Zongker* data), since the found prototypes represent the clusters. The two methods that especially focus on the preservation of the original embedded configuration, i.e. the APE and the LAE, are not significantly better than the other approaches. This again may suggest that the goal of classification should determine the way the objects are chosen for R. In principle, all systematic approaches considered here may work well. The random selection, although not best, but it is also never the worst.

In comparison to the prototype selection methods investigated in the dissimilarity space approach, section 9.2, somewhat different conclusions can be drawn with respect to the specific data (compare plots in section 9.2.3 with the plots in the current section). The *GeoSam* is judged as an easier problem than in the dissimilarity space approach, while the *GeoShape*, the other way around. Also, the *Polydisth* problem seems to be better attacked by the dissimilarity space approach, while the *NIST-38* can be better discriminated in an embedded space. Such observations indicate that both dissimilarity and embedding space approaches should be studied for choosing the best recognition strategy.

Conclusions. Important conclusions can be drawn from our study on dissimilarity data embedded in pseudo-Euclidean spaces. First of all, the NLC, built in an embedded space defined by all training objects can significantly outperform the k-NN rule. Secondly, a representation set R of less than 20% of the training size can be selected, on which the approximated space is defined. In such an approximated embedded space, the NLC can reach the same or even a much higher accuracy than the best k-NN rule based on *all* training objects (this holds for the *GeoShape* provided that the NQC is considered instead). Thirdly, the KCentres procedures work well for multi-modal data. For a small number of prototypes and a non-separable classification problem, the criterion based on the classification error (here, the NLC-err) should be recommended. Finally, we have observed that similarly as in the dissimilarity space approach, a random selection is also beneficial.

In this study, $m+1$ objects were used to define an m-dimensional approximated embedded space. It is also possible to use more objects, which remains an issue for further research.

9.4 On corrections of dissimilarity measures

In the dissimilarity space approach or the embedding approach we do not require metric properties of a dissimilarity measure d (d should be nonnegative and obey the reflexivity condition, Def.2.30). We demand that the compactness hypothesis is fulfilled by designing a measure which yields small values for objects that share many commonalities. This guarantees that such a measure is meaningful for the problem, i.e. the classes of objects will have some compact description. Ideally, we would like to guarantee a *true representation* which requires that by a comparison of dissimilar objects, a large dissimilarity value is obtained. More research is needed to study these issues.

Although our approaches to dissimilarity representations can handle arbitrary measures, an open question refers to possible benefits of correcting the measure to make it metric or even Euclidean [70, 319]. Metric or Euclidean distances can be interpreted in appropriate spaces, which posses many useful algebraical properties and where an arsenal of discrimination functions exists. This might also be interesting for the k-NN rule, since metric properties allow for a construction of a faster approximation rule; see e.g. [273]. Here, we investigate some ways of making a dissimilarity measure either 'more' Euclidean or 'more' metric and the influence of such corrections on the

performance of some decision rules[8]. We will experimentally show that the corrected measures do not necessarily guarantee a better discrimination.

9.4.1 Going more Euclidean - an experimental investigation

From section 3.3, it is known that the Gram matrix $G = -\frac{1}{2} J D^{*2} J$ is positive semi-definite (psd) iff a symmetric distance matrix D is Euclidean. Consequently, if G has p positive and q negative eigenvalues, D is non-Euclidean and a perfectly embedded Euclidean configuration X cannot be constructed. However, D can be corrected such that it becomes Euclidean, which is equivalent to making the corresponding Gram matrix G psd. Some possible approaches to address this point were discussed in section 3.3.2. Here, they are briefly mentioned:

- *Clipping*. Only p positive eigenvalues are considered yielding a p-dimensional configuration $X = Q_p \Lambda_p^{1/2}$. Now, after neglecting the negative contributions, the resulting Euclidean representation overestimates the actual dissimilarities.
- *Adding* 2τ. There exists a positive $\tau \geq -\lambda_{\min}$, where $\lambda_{\min}$ is the smallest (negative) eigenvalue of G, such that $D_{2\tau} = [D^{*2} + 2\,\tau\,(\mathbf{1}\mathbf{1}^T - I)]^{*1/2}$ is Euclidean [171, 301]. This means that the corresponding G_τ is positive definite. In practice, the eigenvectors of G and G_τ are identical, but the value τ is added to the eigenvalues, giving rise to the new diagonal eigenvalue matrix $\Lambda_\tau := \Lambda_k + \tau\, I$. The original dissimilarities are distorted significantly if τ is large.
- *Adding* κ. There exists a positive $\kappa \geq \lambda_{\max}$, where $\lambda_{\max}$ is defined in Theorem 3.40, such that such that $D_\kappa = D + \kappa\,(\mathbf{1}\mathbf{1}^T - I)$ is Euclidean. The corresponding Gram matrix G_κ has the eigenvalues and eigenvectors which are different than these of the original Gram matrix G .
- *Power or Sigmoid transformation*. There exists a parameter p such that $D_p = (g(d_{ij}; p))$ is Euclidean for a concave function g defined as $g(x) = x^p$ with $p < 1$ or as a sigmoid $g(x) = 2/(1 + e^{-x/s}) - 1$ [70]. In practice, p is determined by a trial and error.

These approaches transform the problem such that a Euclidean configuration can be found. It is, however, still possible that the applied corrections are less than required for imposing the Euclidean behavior. In such cases, the measure is simply made 'more' Euclidean (hence, also 'more' metric), since the influence of negative eigenvalues become smaller after some proper transformations. An additional point to realize is that in case of approximate embeddings of a fixed dimensionality the spaces derived from D and $D_{2\tau}$ will differ. This is caused by the fact the dimensions corresponding to the negative eigenvalues become now the less important (by adding τ to all eigenvalues, the negative ones become the closest to zero) in the latter case, so they will not be selected. So, if the negative eigenvalue contributions are large, the corresponding eigenvectors will represent the space obtained from D. This means that the spaces obtained from an approximate embedding of the original dissimilarity data and the corrected ones are very different if the dissimilarity measure is highly non-Euclidean.

Five dissimilarity data are used in our study; see appendix A for details. The first two sets refer to the dissimilarity representations built on the contours of pen-based handwritten digits [31]. The digits are represented by strings of vectors between the contour points for which an edit distance with a fixed insertion and deletion costs and with some substitution cost is computed. The substitution costs such as an angle and a Euclidean distance between the vectors lead to two different representations [47], denoted as *Pen-dist* and *Pen-angle*, respectively. Both measures are non-Euclidean and non-metric. Here, only a part of the data consisting of 3488 examples, is considered. The values are also scaled by some constant to bound the dissimilarities. The digits are unevenly represented; the class cardinalities vary between 334 and 363. Another dissimilarity data set consisting of 2000

[8] The results presented here come from [296]

Table 9.5: Non-Euclidean and non-metric aspects of some dissimilarity representations used for experiments in section 9.4.1. The ranges of r_{mm}, r_{neg} and c indicate the smallest and largest values found for $D(R,R)$, where $|R|$ varies between $30-500$ or $10-200$ for the digit and polygon data, respectively. As a reference, the last two columns present the average and maximum dissimilarity for the complete data.

DR	r_{mm}^{nE} [%]	r_{rel}^{nE} [%]	c	avr. dissim.	max dissim.
Pen-angle	$[10.6, 12.2]$	$[\ 9.4, 24.1]$	$[0.0, 0.3]$	7.1	20.0
Pen-dist	$[13.8, 14.3]$	$[14.2, 27.8]$	$[0.3, 1.0]$	4.0	12.5
Zongker	$[27.5, 35.5]$	$[10.6, 35.5]$	$[0.1, 0.5]$	0.6	1.0
Polydisth	$[13.0, 25.5]$	$[\ 5.4, 31.6]$	0	1.2	3.1
Polydistm	$[\ 5.0, 13.0]$	$[\ 1.8, 24.6]$	$[0.0, 0.1]$	0.7	1.6

examples evenly distributed over ten classes describes the NIST digits [420]. Here, the dissimilarity measure based on deformable template matching [207] is used. The data are referred as the Zongker dissimilarity data. The last two representations are derived for randomly generated polygons. They consist of convex quadrilaterals and irregular heptagons. The polygons are first scaled and then the Hausdorff and modified Hausdorff distances, defined in section 5.4, between their vertices are computed, yielding the *Polydisth* and the *Polydistm* dissimilarity data. The two classes are equally represented by 2000 objects.

If the dissimilarity d is Euclidean, then for a symmetric $D=(d_{ij})$, all eigenvalues λ_i of the corresponding Gram matrix G are non-negative. Hence, the magnitudes of negative eigenvalues show the deviation from the Euclidean behavior. An indication of such a deviation is given by $r_{mm}^{nE} := |\lambda_{min}|/\lambda_{max} \cdot 100$, that is the ratio of the smallest negative eigenvalue to the largest positive one. The overall contribution of negative eigenvalues can be estimated by $r_{rel}^{nE} := \sum_{\lambda_i<0} |\lambda_i| / \sum_{j=1}^{n} |\lambda_j| \cdot 100$. Both these indices come from formulas (9.1). Any symmetric D can also be made metric by adding a suitable value c to all off-diagonal elements of D. Such a constant can be found as $c = \max_{p,q,t} |d_{pq} + d_{pt} - d_{qt}|$. A smaller value imposing a metric behaviour of D was determined by us in a binary search. Table 9.5 provides suitable information on the Euclidean and metric aspects of the measures considered. The following observations can be made:

- The *Pen-angle* data set is moderately non-Euclidean and nearly metric.
- The *Pen-dist* data set is both moderately non-Euclidean and non-metric.
- The *Zongker* data set is highly non-Euclidean and highly non-metric.
- The *Polydisth* data set is highly non-Euclidean, yet metric.
- The *Polydistm* data set is moderately non-Euclidean and slightly non-metric.

The experiments are repeated 50 times for the representations sets of various cardinalities and the results are averaged. The representation objects are randomly selected. The cardinality $|R|$ varies from 3 to 50 examples per class (ten classes) for the digit data sets and from 5 to 100 examples per class (two classes) for the polygon data. For each $|R|$, two cases for the training set T are considered: $T=R$ and T consisting of 100 or 200 objects per class for the digit and the polygon dissimilarity representations, respectively. In the latter case, the ratio of $|T|/|R|$ becomes smaller with the growing $|R|$. The test sets consist of 2488, 1000 or 3600 examples for the pen-digit, NIST digit and polygon data, correspondingly. For each dissimilarity representation, the k-NN rule is considered, as well as the linear discriminant, the NLC, built in both embedded and dissimilarity spaces. The embedding is derived from $D(R,R)$, but additional objects $T \setminus R$, if available, are projected there and used for constructing the classifiers. To denoise the data and avoid the curse of dimensionality, the dimensionality of the embedded space was fixed to $0.3|R|$, so the dimensions corresponding to insignificant (small in magnitude) eigenvalues are neglected. Also the principal component analysis (PCA) [97, 138] was applied in the dissimilarity space $D(\cdot,R)$ to reduce the

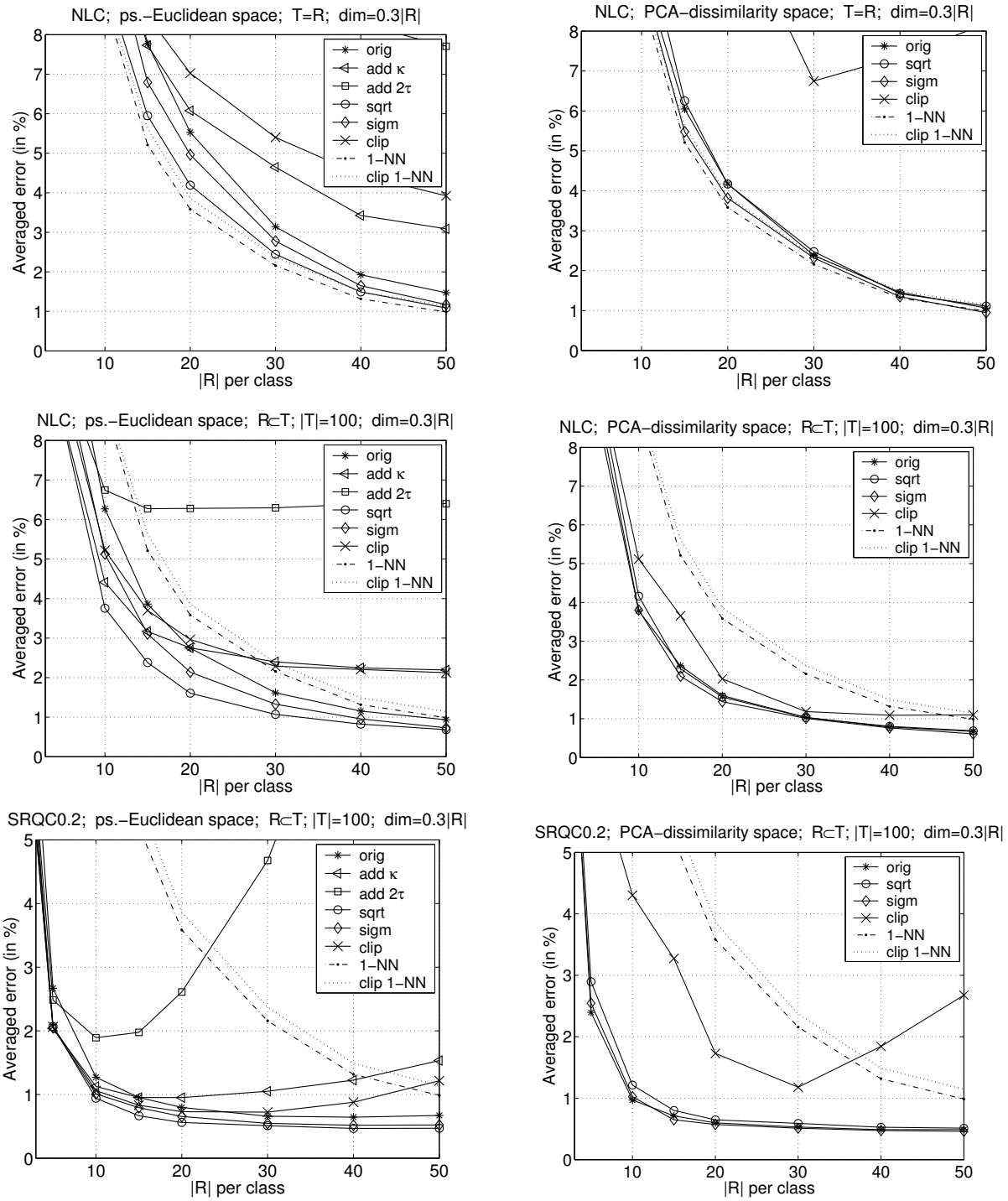


Fig. 9.23: Averaged (over 50 runs) classification error of the NLC (top and middle rows) and the SQRC (bottom row) for the *Pen-angle* dissimilarity data as a function of the number of representation objects.

dimensionality to $0.3|R|$. In both cases, although the dimensionalities are reduced, the spaces are defined by all representation objects R.

9.4.2 Results and conclusions

Adding a constant to the dissimilarities or applying a concave transformation preserves their order, hence it does not influence the behavior of the k-NN rule. However, during clipping (where all negative eigenvalues are neglected in the embedding process), the recomputed Euclidean distances non-monotonically differ from the original ones, hence the k-NN rule will behave differently. Also

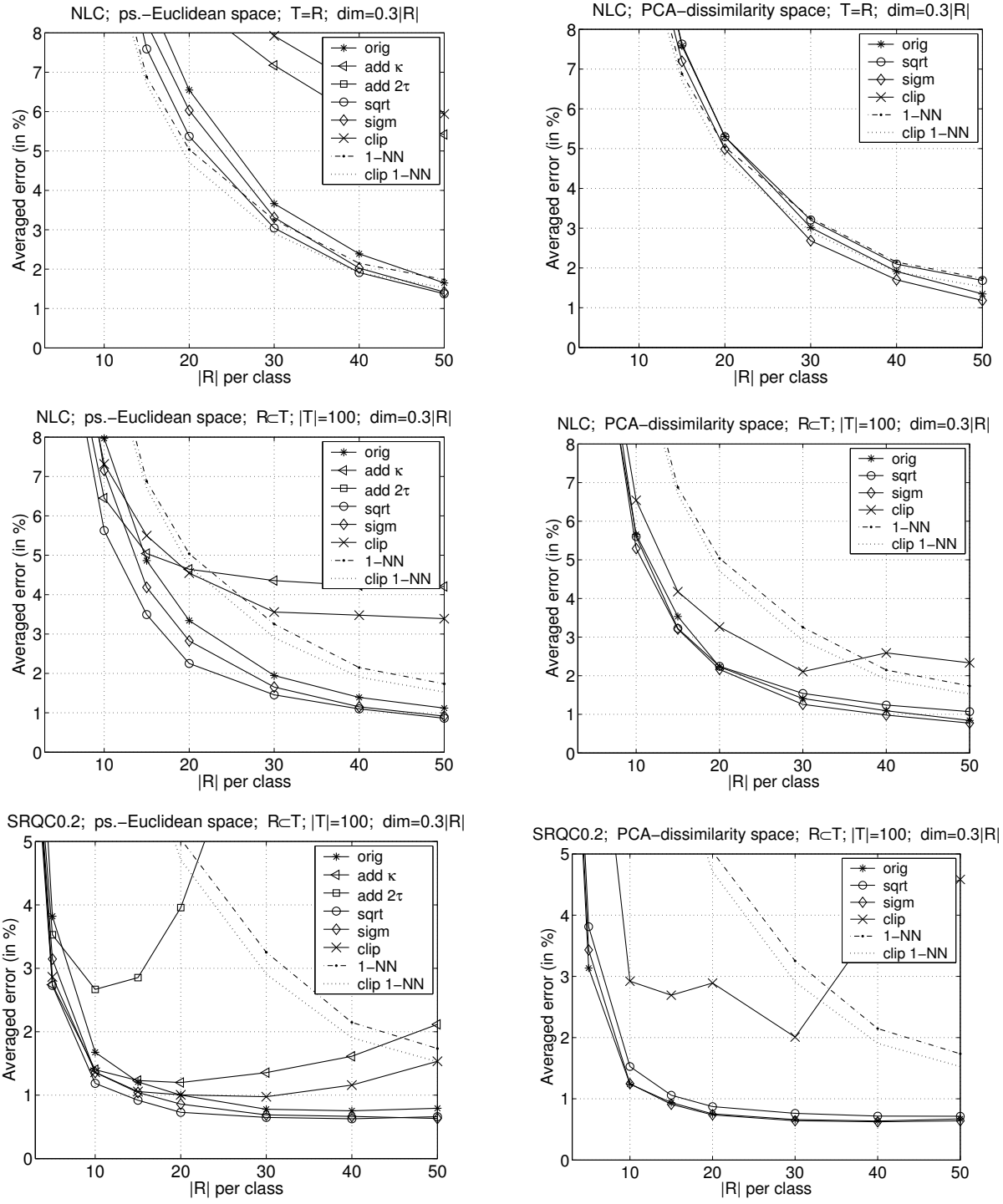


Fig. 9.24: Averaged (over 50 runs) classification error of the NLC (top and middle rows) and the SQRC (bottom row) for the *Pen-dist* dissimilarity data as a function of the number of representation objects.

both embedded and dissimilarity spaces change, so a linear classifier will change as well[9]. In our experiments, we study the influence of such corrections on the given measures for various representation sets R. For this purpose, a proper κ and a proper τ guaranteeing the Euclidean behavior are determined. Two concave transformations are also additionally considered: the square

[9] Adding a constant is not worth doing in a dissimilarity space, since a constant shift is then applied to all D_{ij}, but the self-dissimilarity stays the same, that is $D_{ii} = 0$. Because of that, the classifier performance is expected to stay the same or worsen somewhat. On the other hand, if we apply the shift s to *all* dissimilarities, the constructed classifiers should be the same, since all vectors $D(\cdot, R)$ are shifted by the same vector $s\mathbf{1}$.

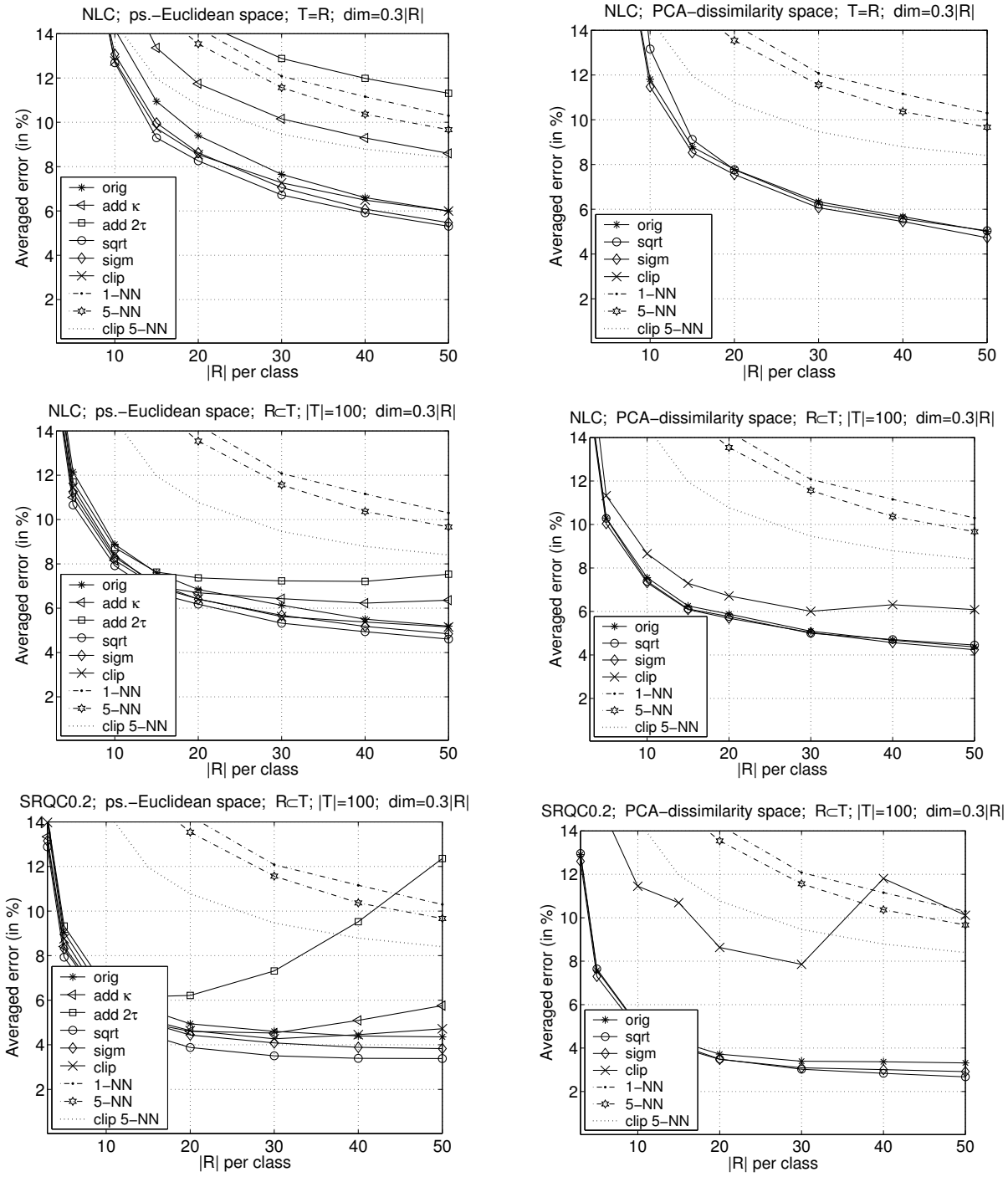


Fig. 9.25: Averaged classification error of the NLC (top and middle rows) and the SQRC (bottom row) for the *Zongker* dissimilarity data as a function of the number of representation objects.

root (which makes the dissimilarity measures closer to Euclidean, yet still non-Euclidean) and the sigmoid with the slope s being the average dissimilarity between the representation objects. The measures are non-Euclidean, but less than the ones originally given as judged by the magnitudes of negative eigenvalues in the linear embeddings.

The results of our experiments compare the averaged performance of the NLC and the 1-NN rule and the best k-NN rule (if $k > 1$). They are presented in Fig. 9.23-9.27. The standard deviations (for all the data) reach on average 0.3% and maximally $0.8 - 1.4\%$ for very small R. Additionally, the performance of the SRQC (strongly regularized quadratic classifier with the regularization of 0.2; see section 4.4.1) for the digit data is shown in Fig. 9.23 - 9.25 to indicate that such a classifier can

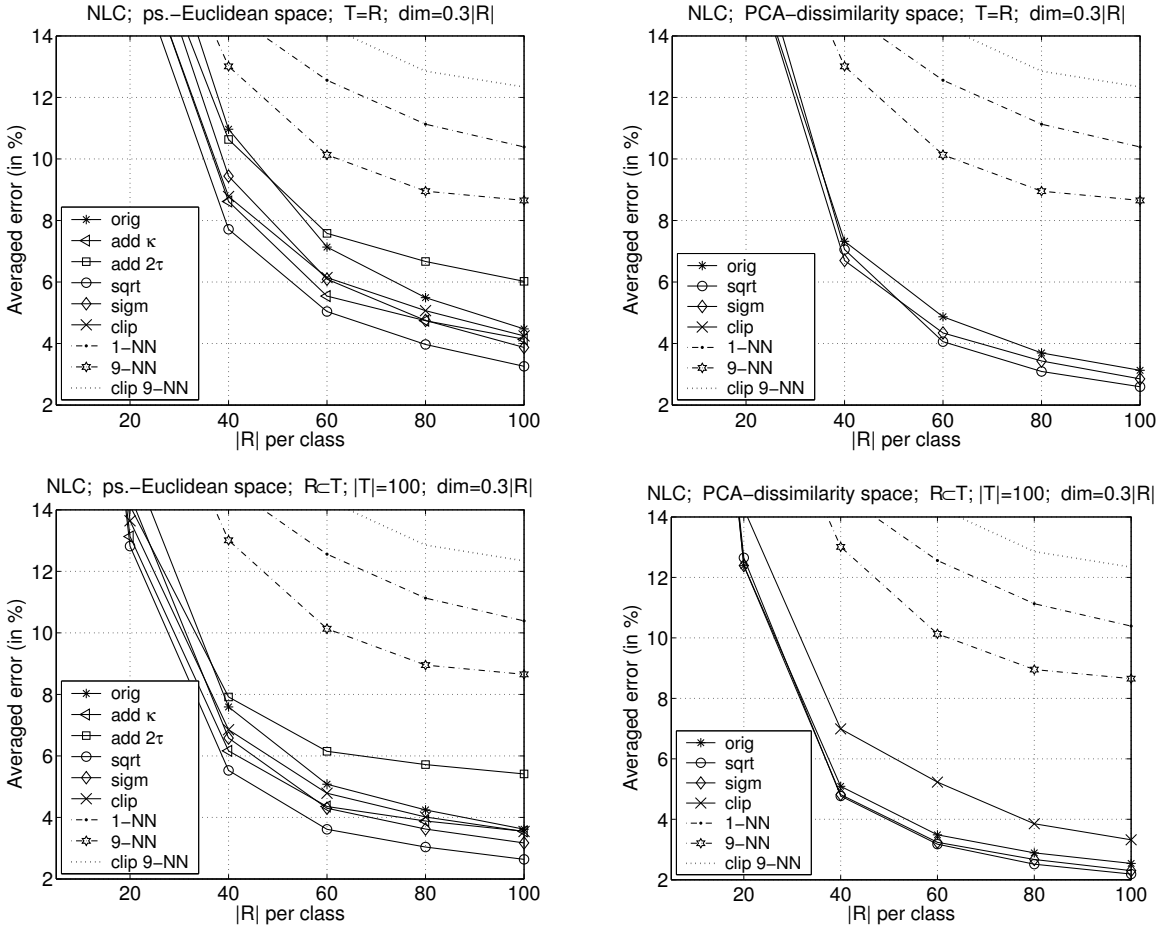


Fig. 9.26: Averaged (over 50 runs) classification error of the NLC for the *Polydisth* dissimilarity data as a function of the number of representation objects.

reach even a better accuracy than the linear one. The legends refer to the following transformations:

- orig - the original dissimilarities; no transformation is applied.
- add κ / add 2τ - a constant value is added to the off-diagonal dissimilarities; $D(R, R)$ becomes Euclidean.
- sqrt/sigm - a square root or a sigmoidal transformation of the dissimilarities; $D(R, R)$ becomes 'more' Euclidean.
- clip - only positive eigenvalues from the linear embedding are used to derive the Euclidean distance representation.

The k-NN rules are directly applied to the dissimilarities $D(T_{te}, R)$ to derive the classification error. The clip k-NN rules operate on the Euclidean distances derived from the clipped version of the linear embedding of D obtained by taking only the positive contributions (neglecting the negative eigenvalues).

Conclusions. By analyzing our results in Fig. 9.23 – 9.27, the following conclusions can be made:

1. The correction of D based on adding 2τ to all square dissimilarities different than the self-dissimilarities yields worse results than by adding κ to the dissimilarities, while the NLC is trained in the corresponding embedded spaces. The former results are missing on some plots since they are worse than the chosen scale.
2. The NLC and the SRQC in the (corrected or not) dissimilarity spaces perform similarly or better than in the pseudo-Euclidean spaces. It can be observed by comparing right and left

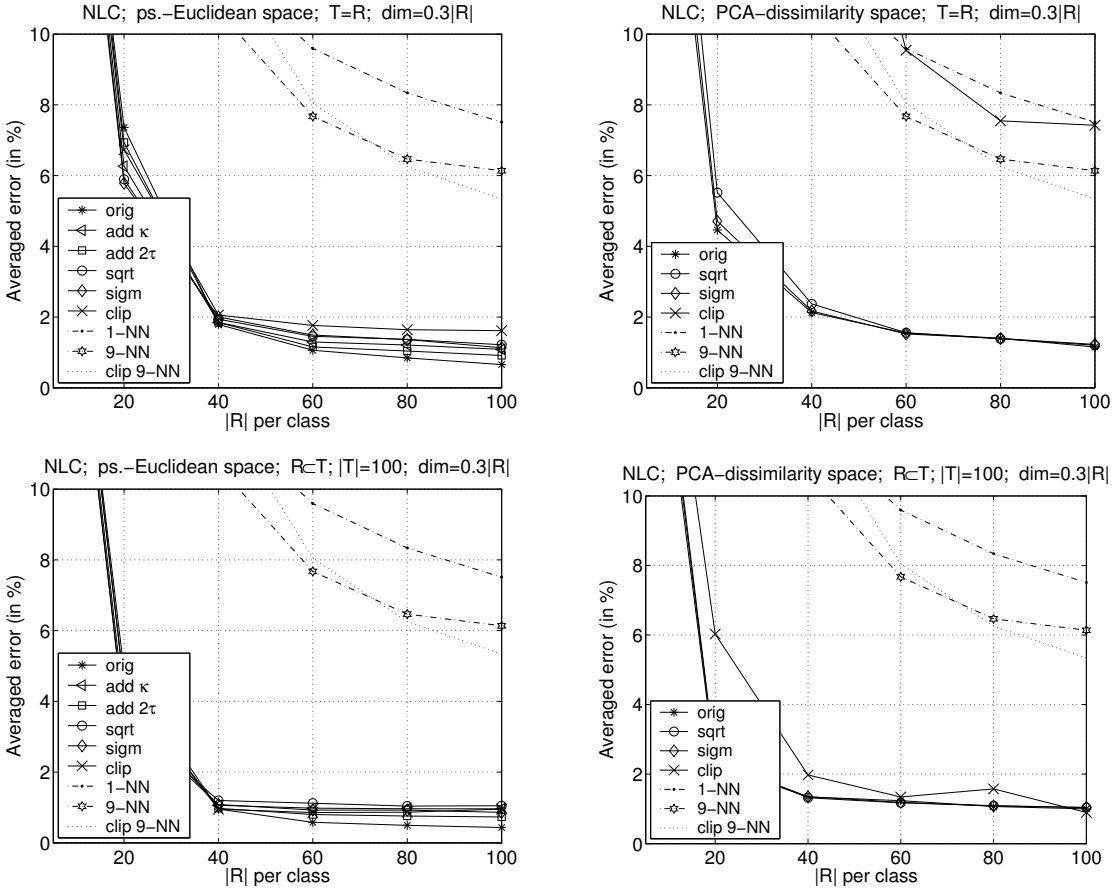


Fig. 9.27: Averaged classification error of the NLC for the *Polydistm* dissimilarity data as a function of the number of representation objects.

plots in all figures.

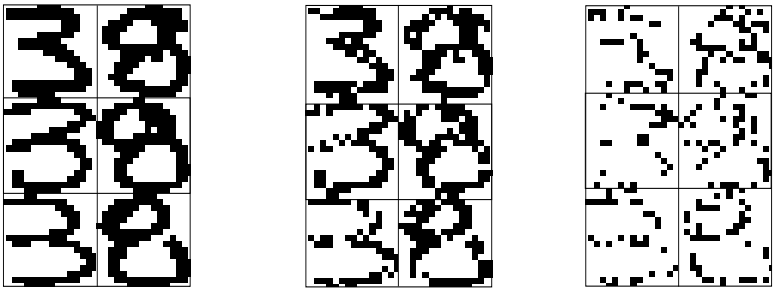
3. For large T and small $R \subset T$, the NLC and the SRQC in both the embedded and dissimilarity spaces (original or transformed by the square root or sigmoidal transformation) significantly outperforms the k-NN and the clipped k-NN rules. This can be observed in the bottom rows in all figures. For $T = R$, this phenomenon is much less pronounced; the k-NN might even become somewhat better the alternative classifiers, as seen for the *Pen-angle* data in Fig.9.23, bottom row.
4. Concave transformations (here the square root and the sigmoid function) have minor effect with respect to the original dissimilarities, when the NLC or the SRQC are built in dissimilarity spaces. On the contrary, these classifiers deteriorate their performance while they are constructed on the 'clipped' Euclidean distance spaces.
5. Concave transformations of the dissimilarities seem beneficial for the NLC and the SRQC in the corresponding pseudo-Euclidean spaces. These classifiers may perform better in such spaces than in embedded spaces derived from the original dissimilarities or in the Euclidean spaces obtained from the embedding of otherwise corrected dissimilarities. Interestingly, the results of the NLC and the SRQC in the original dissimilarity spaces are comparable or even better. In general, the square root transformation seems to work well.

If for small representation sets the k-NN is far from optimal, linear (quadratic) classifiers built in both embedded or dissimilarity spaces can significantly outperform the k-NN rule. Concave transformations of dissimilarities are somewhat beneficial for the classifiers built in embedded spaces, however, they may have no essential effect in dissimilarity spaces (as judged from right plots in all figures). None of the transformations considered here allows the NLC and the SQRC for reach-

ing a considerably better performance than reached in original dissimilarity spaces. Thereby, we conclude that the potential advantages of the imposed Euclidean behavior are doubtful, that is they cannot be always guaranteed. It is more important that the measure itself describes compact classes than its strict Euclidean or metric properties. This can be influenced by concave transformations which aim at diminishing the relative effect of large dissimilarities and not by making them really Euclidean[10].

9.5 Some remarks on a simulated missing value problem

We think that dissimilarity representations are suitable for handling missing value problems. In order to study their applicability for that purpose, a missing value problem has been simulated for the recognition of the NIST digits 3 and 8 [420]. Here, images re-sampled to a 16×16 raster are studied. To analyze the performance of classifiers as a function of the number of missing values, the images of 3 and 8 have been randomly corrupted. The level of corruption (degradation) is governed by a probability P that a particular image pixel is unknown. Four different degradation levels are used in our experiments, i.e. $P = \{0.0, 0.2, 0.4, 0.6\}$; see Fig. 9.28. Because the images are binary, the missing values can be just assigned to the background pixels. This is in agreement with one of the approaches to the missing value problem, where the unknown value becomes either the average or the most common value among all other present values.



No degradation; $P = 0$ $P = 0.2$ $P = 0.6$

Fig. 9.28: Simulation of a missing value problem by degradation. Degradation of 16×16 binary images of digits 3 and 8. The level of degradation is governed by the probability P that an individual pixel is set to background.

The usual way of computing dissimilarities on the binary data is to construct a similarity measure first and then to transform it to the corresponding distance. For the binary objects i and j the similarity measures are often based on the variables a, b, c and d reflecting the number of elementary matches between the objects, as explained in section 5.1. Three dissimilarity measures were chosen for the analysis: Jaccard, $d_{ij} = \sqrt{1 - \frac{a}{a+b+c}}$ (Euclidean), simple matching, $d_{ij} = 1 - \frac{a+d}{a+b+c+d}$ (non-Euclidean metric) and Yule, $d_{ij} = 1 - \frac{ad-bc}{ad+bc}$, (non-metric); see also Table 5.1. The Jaccard measure is of interest, since it is the overlap ratio excluding all non-occurrences, and, thereby, disregarding the information on matches between the background pixels. On the contrary, the simple matching measure describes the proportion of the matches with respect to the total number of pixels. Hence, it counts the matches between the background pixels, where some of them are in fact the unknown value. The Yule dissimilarity is a cross-product ratio.

Our aim is to compare the behavior of the classification methods on these dissimilarities. For each level of degradation, complete distance representations were computed. We assume that the training

[10] Note that the beneficial effect of a nonlinear transformation of dissimilarities for a random prototype selection and the NQC trained in transformed dissimilarity spaces has been already observed in Fig. 9.17.

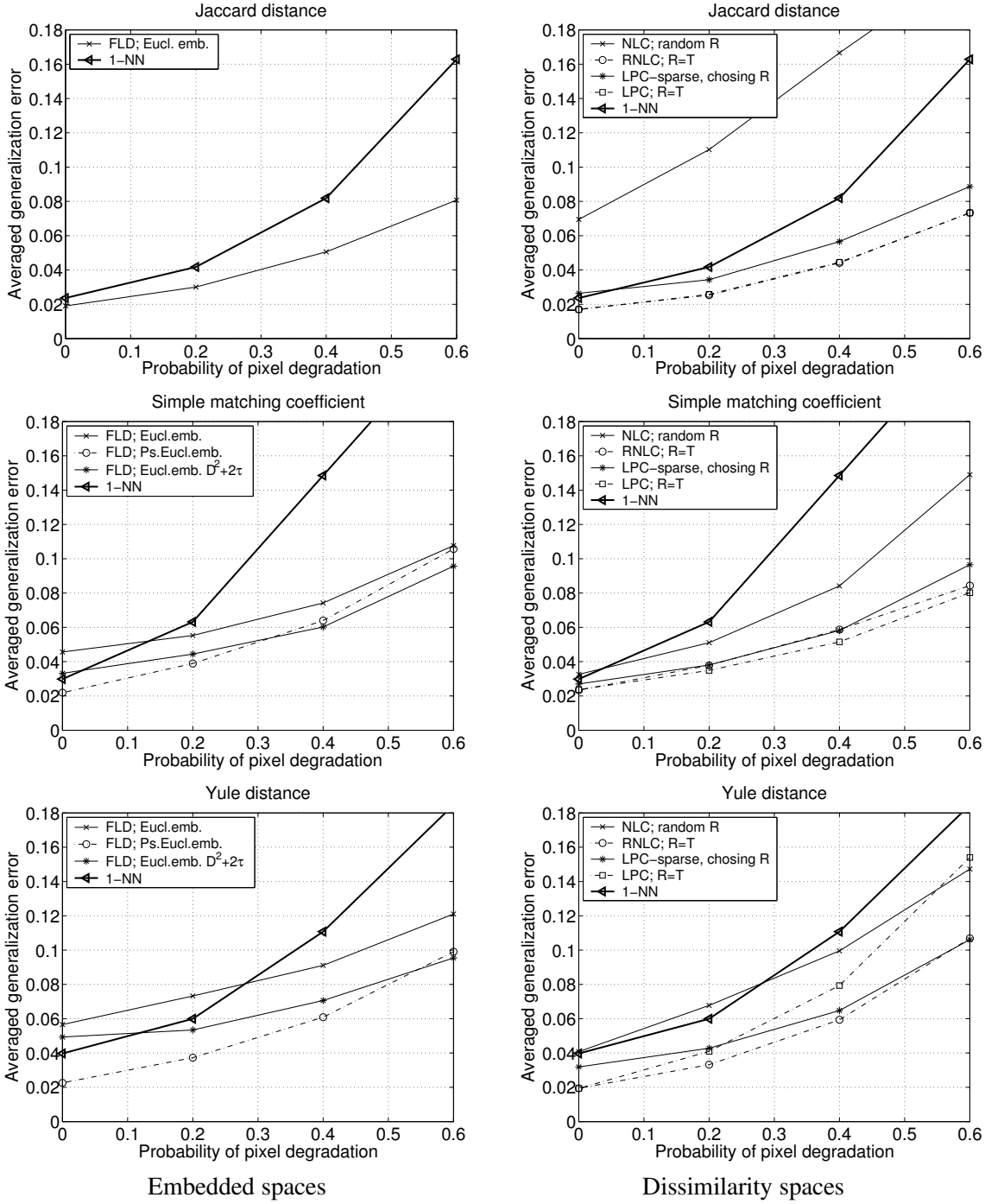


Fig. 9.29: Comparison of classification approaches in embedded spaces (left) and in dissimilarity spaces (right) on three different dissimilarity representations: Jaccard (top), simple matching coefficient (middle) and Yule (bottom). The standard deviations of the averaged results are less than 0.2% for the degradation level $P \leq 0.2$ and less than 0.4% for the larger P.

and the test sets are degraded in a similar way. A training set of a fixed size of 100 samples per class was randomly chosen. All the classifiers are tested on an independent test set of 500 samples per class. The testing procedure is repeated 20 times and the results are averaged. Both the training and testing sets have now the fixed sizes and the varying quantity is the level of image degradation.

The Fisher linear discriminant (FLD), section 4.5, is trained in embedded spaces: the Euclidean space created by the restriction of the complete pseudo-Euclidean embedding, pseudo-Euclidean embedded space and the corrected Euclidean space (section 3.3.2). All the spaces are retrieved with

a large dimensionality corresponding to the 99.9% of the preserved variance; see section 3.3.4. In the dissimilarity space approach, the following classifiers were used: the RNLC with the regularization of $\lambda = 0.01$ and both sparse and non-sparse linear programming classifiers (LPC) built on the entire dissimilarity representations $D(T,T)$, formulations (4.14) and (4.15), respectively. The sparse LPC selects, in fact, its own representation set. The NLC was also built on the representation $D(T,R)$ with R consisting of 25% randomly chosen objects out of T.

Fig. 9.29 presents the generalization error rate as a function of the increasing data degradation for the Jaccard, simple matching and Yule measures and three approaches: the 1-NN rule, the embedding approach and the dissimilarity space approach. The following conclusions can be drawn:

- The performance of all considered decision rules deteriorate with the increasing corruption (missing information) level. Still, the best decision rules reach the error of $8\% - 10\%$ for $P = 0.6$, while the 1-NN rule reaches the error of $\approx 18\%$ for the same degradation level.
- Most of the linear classifiers, both in the embedded and dissimilarity spaces, outperform the 1-NN rule. They are also more robust against the missing values. Comparing all results, the 1-NN rule deteriorates the most.
- The NLC in a dissimilarity space defined by R based on 25% randomly chosen training examples often yields worse results than the other classifiers (right column of Fig. 9.29).
- On average, the Jaccard dissimilarity allows for a better separability of classes than the Yule and simple matching distances. Two methods give identical errors: the RNLC and the LPC (both with $R = T$) in dissimilarity spaces. They also achieve the smallest overall errors, which for the non-degraded images equals 1.7%.

As a reference, we report the best results for other, more sophisticated representations based on the Euclidean distance between the Gaussian-smoothed 128×128 images and the modified-Hausdorff between the digit contours. For the training set of 100 objects per class, the best linear classifiers in the embedded and dissimilarity spaces reach $\approx 4\%$ for the Euclidean representation and 6% for the modified-Hausdorff representation, while the 1-NN error is $\approx 6\%$ for both of them; see also [301].

It is interesting that a simple distance measure (like Jaccard), operating on binary images of digits, outperforms the modified-Hausdorff dissimilarity, computed on the contours. A possible explanation is that the Euclidean and modified-Hausdorff dissimilarities are computed on the original 128×128 images, while in the first case, the images were rescaled to a lower raster and by this, the digits became aligned. The binary dissimilarity measures are also considerably robust against the data degradation. The FLD in an embedded space and the RNLC and the LPC in dissimilarity spaces applied to the degraded images at the level of $P = 0.2$ still perform comparably to the best results on the Euclidean or modified-Hausdorff distances. This still remains true for the degradation level of $P = 0.4$ and the Jaccard distance representation.

In summary, we conclude that the presented binary dissimilarity measures (especially the Jaccard one) are robust against missing (corrupted) information, when the classifiers are built in the embedded or dissimilarity spaces. Among the classifiers considered, the 1-NN rule shows the highest sensitivity to data degradation, which is to be expected due to its sensitivity to noisy examples. For imperfect dissimilarity measures, the 1-NN method can be outperformed by more sophisticated classifiers, taking into account a number of representative objects, thus becoming more global in their decisions.

9.6 The existence of zero-error dissimilarity-based classifiers

In the statistical approach to pattern recognition numerical features are used to describe objects as vectors in a vector space. Usually, such features are reduced descriptions of objects. Some (sig-

nificant) information is lost and, as a consequence, essentially different objects may be represented as the same vectors in the feature space. If this occurs for objects of different classes, the classes overlap. There is no way of distinguishing such objects in the feature space and, thereby, any recognition scheme based on such a feature representation has a non-zero classification error. As a result, an error free recognition system is even asymptotically (for infinite training sizes) impossible. To handle this, traditional statistical classifiers estimate the class probability density functions and built the decision rules by minimizing the estimated class overlap.

A dissimilarity-based approach to pattern recognition relies on the dissimilarities computed between pairs of objects, while making use of their biological variability in the training set (which is observed by the variations in the dissimilarities). If the dissimilarities are directly found on the raw measurements (which contain all significant information on the objects), the loss of information by the reduction to features, may be avoided. Under some circumstances, an assumption of a zero-error classification (hence no class overlap) holds for dissimilarity representations[11]. Here, we will discuss when such an assumption holds.

The NN rule is often practiced on the dissimilarity data, usually metric distances. In such a case, the training set T can be used for the selection of prototypes R, but when R is chosen, the remaining objects $T \backslash R$ are not used for training. Other decision rules constructed either in embedded or dissimilarity spaces make use of all training objects. They may demand less prototypes than the NN rule for reaching the same performance and, thereby, a smaller computational complexity. As mentioned above, under some conditions, the class overlap related to the use of feature spaces can be avoided by the use of dissimilarities. The question arises whether it is possible to build classifiers that exploit this in practice. In other words, whether we can construct classifiers that have asymptotically (for increasing training set sizes) a zero classification error. Note that for non-overlapping classes (and metric dissimilarities), the asymptotic error of the 1-NN rule is zero [87]. This may be, however, impractical to reach, since it may demand an infinite training set to be stored and handled.

9.6.1 Asymptotic separability of classes

If the dissimilarity measure is zero if and only if the corresponding objects are identical, and if real objects can be unambiguously labeled, then the class overlap may be avoided. This assumption can be exploited by trying to construct zero-error classifiers [103], which should make use of the property of non-overlapping classes and define the decision function in the 'gap' between them. In fact, this implies that the 1-NN rule will constitute such a zero-error classifier. It may demand, however a very large training set. As classifiers in both dissimilarity and embedded spaces appear to be much more efficient than the 1-NN by requiring a small number of prototypes for the construction, the question arises whether these classifiers may also have an asymptotic zero-error.

Assumptions. The discussion is based on the following assumptions:

(1) Real, physical classes of objects are separable, i.e. there is no physical object that belongs to more than one class.

(2) Raw measurements of objects are such that this separability is maintained[12].

(3) The dissimilarity measure $d(x,y)$ between the objects x and y constructed on their raw measurements (e.g. scanned images) is such that $d(x,x)=0$ and $d(x,y) \geq \delta > 0$ if x and y belong to different classes. This assumption states that there exists some 'gap' between the classes of the δ-size: the objects of different classes have a dissimilarity of at least δ.

[11] This section relies on [103]. The idea of dissimilarity-based zero-error classifiers comes from dr Robert Duin.

[12] One way to inspect this is to let the objects be labeled by humans based on the measurements (e.g. a video screen that displays the object image to be used for a further processing). The possibility of labeling objects correctly should still exist after scanning and display.

(4) The raw measurements of objects x and y are continuous functions of the parameters $\boldsymbol{\theta}$ that influence their generation (e.g. lighting conditions, small rotations or sensor deviations). Hence, the dissimilarity $d(x(\boldsymbol{\theta}), y(\boldsymbol{\theta}))$ is continuous in $\boldsymbol{\theta}$. The noise is such that for any two measurements x and x' of the same physical object $d(x, x') < \delta$ holds.

(5) The digitalization of the measurements and, thereby, the computer representation of the objects is such that the minimum class gap is preserved.

In general, the role of a dissimilarity measure is to capture the notion of commonality or closeness between the objects, i.e. it should be small for similar objects, and possibly large for distinct objects. Consider now a set of objects X. A possible formalization of the notion of closeness between the objects can be achieved by the use of neighborhoods, i.e. a collection of subsets of X for each element $x \in X$. Neighborhoods provide a general tool for describing relations between the elements of X. Such neighborhoods can be defined by the use of dissimilarities as a special case; see sections 2.1 and 2.2 for details. The ε-ball neighborhood of x is given as $b_\varepsilon(x) = \{y \in X : d(x, y) < \varepsilon\}$. The nested neighborhood basis becomes $\mathcal{E}(x) = \{b_\varepsilon(x) : \varepsilon \geq 0\}$ and the space $(X, \mathcal{E})$ is pretopological; see Theorem 2.44.

Elements of each neighborhood show a specific level of similarity and in practical applications only neighborhoods for some chosen, data dependent values of ε can be considered. Since, later on, we want to define classifiers on finite sets, we will restrict ourselves to a local basis. The neighborhood basis of x, $\mathcal{E}_{\varepsilon_x}(x)$ is the set of all y which belong to the ε_x-ball centered at x, i.e. $\mathcal{E}_{\varepsilon_x}(x) = b_{\varepsilon_x}(x)$ for some specified $\varepsilon_x > 0$. Note that ε_x may depend on x. ε_x is chosen such that there exists a distinct object x' in the same class for which $d(x, x') < \varepsilon_x$ holds (e.g. $\varepsilon_x = 1.0001 \cdot d(x, nn(x))$, where $d(x, nn(x))$ is the dissimilarity to the nearest neighbor of x). Consequently, $N \in \mathcal{P}(X)$ is a neighborhood of x if $\mathcal{E}_{\varepsilon_x}(x) \subseteq N$ and the neighborhood system $\mathcal{N}(x)$ is the collection of all neighborhoods of x. Consider now two classes of objects, denoted as ω_1 and ω_2.

Observations. Based on our assumptions, the following observations can be made:

(1) For a sufficiently small positive ε_x and any object x in the class ω_i, there exists a distinct object in the same class such that the dissimilarity between them is smaller than ε_x, $\forall_{x \in \omega_i}\ \exists_{\varepsilon_x > 0}\ \exists_{y \in \omega_i}\ \ (y \neq x \wedge d(x, y) < \varepsilon_x); \quad i = 1, 2.$

(2) $\forall_{x \in \omega_i}\ \exists_{y \in \omega_i}\ \forall_{N \in \mathcal{N}(x)}\ (y \neq x \wedge y \in N); \quad i = 1, 2.$

(3) The neighborhood basis of all x in ω_1 contains no elements of ω_2, that is $\forall_{x \in \omega_1}\ \forall_{y \in \omega_2}\ y \notin \mathcal{E}_{\varepsilon_x}(x) \wedge x \notin \mathcal{E}_{\varepsilon_y}(y)$ and vice versa.

(4) All x from the class ω_1 have a neighborhood that contains no elements of the class ω_2, $\forall_{x \in \omega_1}\ \exists_{N \in \mathcal{N}(x)}\ N \cap \omega_2 = \emptyset$. Equivalently, $\forall_{x \in \omega_2}\ \exists_{N \in \mathcal{N}(x)}\ N \cap \omega_1 = \emptyset$.

This brings us to the existence of a rule that correctly assigns each $x \in \omega_1$ to the class ω_1 and each $x \in \omega_2$ to the class ω_2. The objects outside $\omega_1 \cup \omega_2$ will be mainly rejected. Some of them, sufficiently close (in terms of neighborhoods) either to ω_1 or ω_2, will be assigned to these classes. All objects form the classes ω_1 and ω_2 will, however, be correctly classified. This is a zero-error classifier (with a rejection option), provided that we only deal with the objects from either ω_1 or ω_2.

Theorem* 9.1 Assume two classes ω_1 and ω_2. The following decision rule correctly classifies any x for $x \in \omega_1$ or $x \in \omega_2$:

1. If $\exists_{N \in \mathcal{N}(x)} N \cap \omega_1 = \emptyset \wedge N \cap \omega_2 = \emptyset$, then reject x,
2. else if $\exists_{N \in \mathcal{N}(x)} N \cap \omega_2 = \emptyset$, then assign x to the class ω_1,
3. else if $\exists_{N \in \mathcal{N}(x)} N \cap \omega_1 = \emptyset$, then assign x to the class ω_2.

Proof. Assume that $x \in \omega_1$. By observation (1), one has $\forall_{N \in \mathcal{N}(x)} \exists_{y \in N}\ y \neq x \wedge y \in \omega_1$. Hence, $N \cap \omega_1 \neq \emptyset$ and consequently, rule 1 does not apply. However, by observation (3), $\exists_{N \in \mathcal{N}(x)} N \cap \omega_2 = \emptyset$, which means that

Fig. 9.30: Digits misclassified by the NN rule (top row), their nearest neighbor of the '3'-class (middle row) and their nearest neighbor of the '8'-class (bottom row) for the *Hamming-NIST-38* data.

the rule 2 applies. As a result, x is assigned to the class ω_1. If $x \in \omega_1$, rule 1 does not apply, since some of its neighborhoods have just elements in ω_1. Assume now that $x \in \omega_2$, then as a consequence of rule 3, as rule 2 does not apply, x is classified as a member of the class ω_2. ■

Theorem 9.1 just shows that an error-free classifier exists. It does not describe how such a decision rule may be constructed based on a finite set of training examples. Rule 1 above should take care that the objects not belonging to one of the two classes, $x \notin \omega_1 \cup \omega_2$, are rejected. Some $x \notin \omega_1 \cup \omega_2$, however, having sufficiently small dissimilarities to the objects of at least one of the classes will also be classified either as ω_1 or ω_2. This does not contradict the theorem as it considers the elements $x \in \omega_1 \cup \omega_2$ only. In other words, rule 1 rejects all objects that belong neither to the closure of ω_1 nor to the closure of ω_2. Rule 2 assigns x to ω_1 if x does not belong to the closure of ω_2. Rule 3 assigns x to ω_2 if x does not belong to the closure of ω_1. Rule 4 rejects x if it does not belong to both closures. Furthermore, for each subset A of ω_1, the closure $cl(A)$ of A will be classified as ω_1 (objects which belong to the border of ω_1 have neighborhoods with no elements of ω_2 and vice versa). So, the classes become closed sets.

Experimental investigation. Consider the binary images of the digits 3 and 8 from the NIST database [420]. A Hamming distance (section 5.3) representation *Hamming-NIST-38* is derived between the 32×32 re-sampled images. The first question that arises is whether this set fulfills the assumptions formulated above. All the nearest neighbor relations are checked for this purpose. In Fig. 9.30, the objects misclassified by the NN rule together with their nearest neighbors in both classes are presented. For some objects, it may be concluded that they are badly segmented as they contain isolated dots. As a consequence, they do not fulfill the assumption 4. Object representations based on segmentation errors are not expected to have close neighbors. In a practical situation, they may be removed from the training set. New objects, having such defects, are, thereby, expected to be misclassified. For practical problems, it might be, therefore, difficult to construct a zero-error classifier.

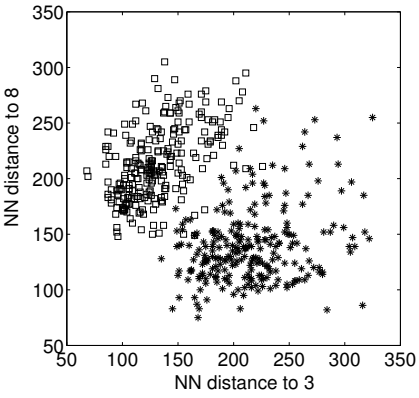


Fig. 9.31: The scatter-plot of the distances to the nearest neighbors of both classes for the *Hamming-NIST-38* data.

In Fig. 9.31, the distances to the nearest neighbors in the Hamming distance representation are shown for a part of the data. In a very few cases, the nearest neighbor belongs to a different class. This causes a classification error. The total leave-one-out 1-NN error estimate is 1.85%. The figure, however, suggests that except for a few cases, a gap between the classes exists. In the following experiments, we try to construct some classifiers in this area. We use a fixed training set of 500 objects per class. The remaining 500 objects per class are used for testing. The following classifiers are considered. The Fisher linear discriminant (FLD) in a dissimilarity space and the FLD in an embedded space, both defined by systematically growing representation sets, chosen from the training set T. Starting from a few objects in the set R, the systematic selection is done iteratively. In each step, the FLD is trained and the training object that is the closest to the current decision boundary is added to the R. For the construction of an embedded space, the eigenvectors corresponding to the

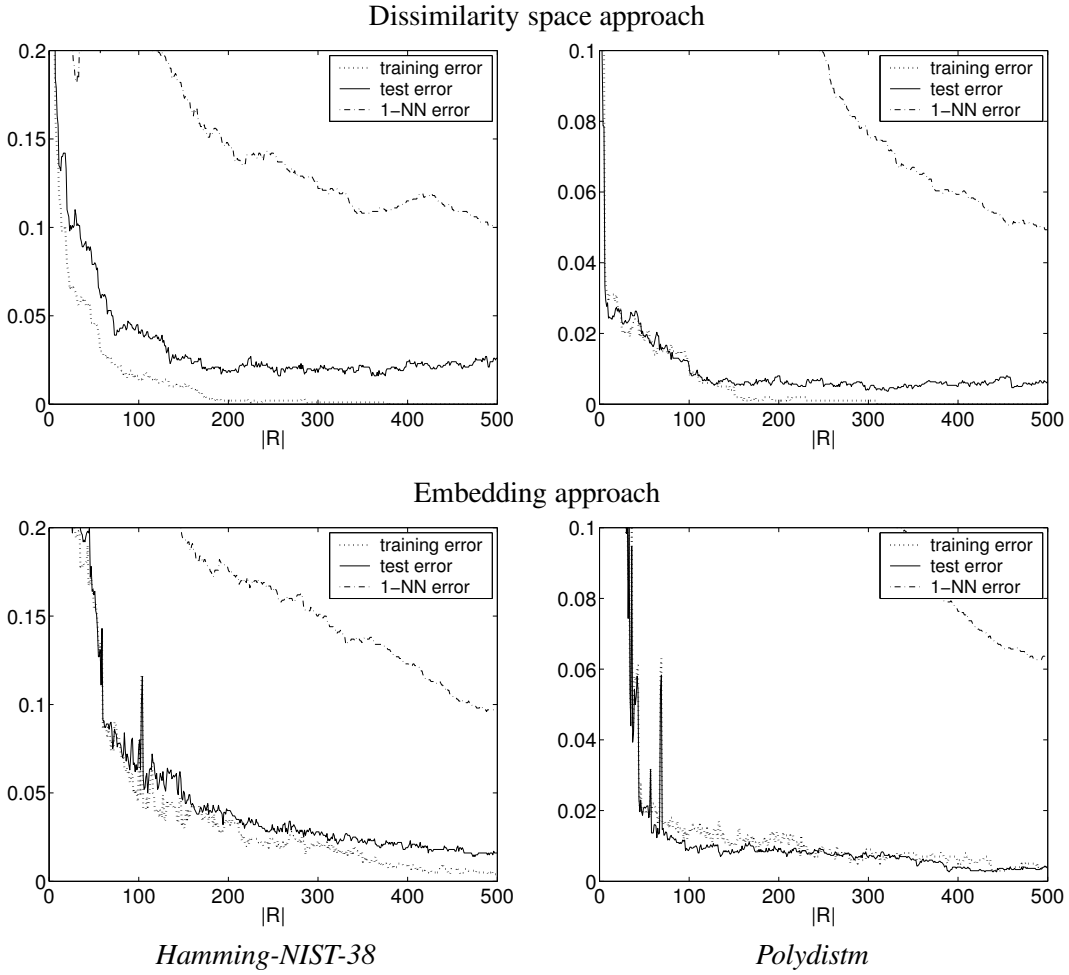


Fig. 9.32: The performance of the FLD in a dissimilarity space (top row) and in an embedded space (bottom row) as a function of the cardinality of R per class for the Hamming representation of the NIST-38 digits (left) and the *Polydistm* data (right). R is a systematically growing subset of T. Both training and test errors are shown. The 1-NN error on the representations set is given as a reference. Note the scale differences.

largest eigenvalues, jointly explaining 70% of the generalized variance are used. The classifiers are trained on $D(T, R)$.

In Fig. 9.32, the classification errors on the training set and test set are plotted as functions of the cardinality of R. The test errors for the 1-NN rule on $D(T_{te}, R)$ are presented as well. This figure shows that a zero-error classifier can be constructed for the training set, but it appears difficult to obtain this result also for the test set. Since the assumption 4 is not fulfilled for these data (some nearest neighbors belong to different classes), this might be an indication of its importance. To judge it fully, however, other experiments are needed. Note the instability of the results for the embedding procedure (bottom plots) for small representation sets.

Additionally, the two-class *Polydistm* polygon data of randomly generated convex quadrilaterals and irregular heptagons is considered; see also section 9.2.2. The care is taken that the heptagons do not degenerate to quadrilaterals. In our experiments with growing representations sets, 500 objects per class are used for training and the remaining 1500 objects per class are used for testing. The results are shown in the right plots of Fig. 9.32. It can be observed that in a dissimilarity space a zero-error classifier can be constructed for the training objects. Although the generalization error oscillates in the neighborhood of zero, a perfect discrimination is not reached for the test data. It should be, therefore, concluded that an error-free classification on an independent test set is hard to obtained.

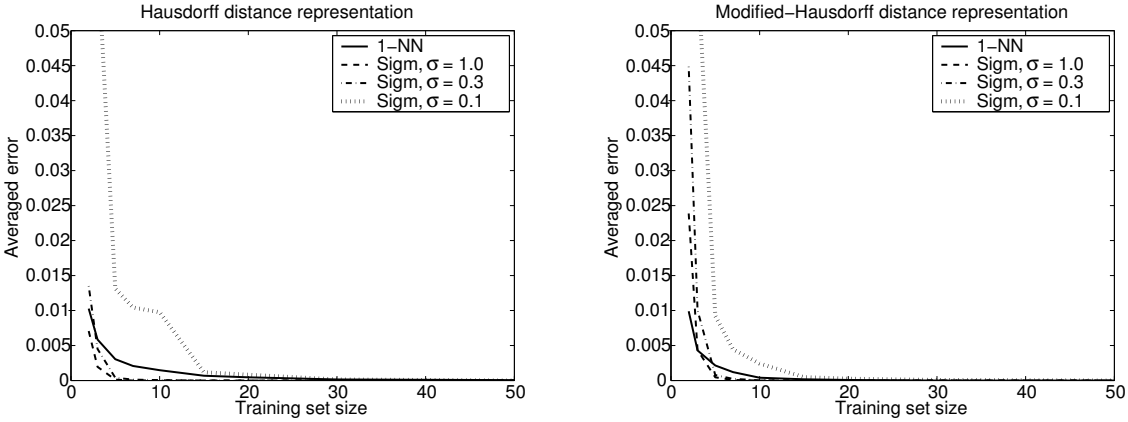


Fig. 9.33: Error curves for the 1-NN rule and the exponential classifier for the discrimination on the *Convex polygon* data represented by the Hausdorff (left) or modified-Hausdorff distances (right).

Further considerations. A zero-error classifier operating on non-Euclidean dissimilarities representing two classes of convex polygons, the *Convex polygon* data, was constructed more carefully. Two classes of polygons are considered: pentagons (class ω_1, based on $t=5$ points) and heptagons (class ω_2, $t=7$). For the generation of a polygon, t vertices (points) are first regularly positioned on the unit circle, i.e. the distances between two consecutive vertices are equal. Next, two-dimensional noise is added to each vertex to perturb the polygons; see appendix A.1. A training set of $2 \cdot 100$ polygons and a test set of $2 \cdot 1000$ polygons are generated. Two dissimilarity measures are studied, the Hausdorff and the modified Hausdorff distances, as defined section 5.4. The distance of a polygon to itself is zero and it is positive for any pair of non-identical polygons. Distances vary in a continuous way with the changes in the vertex positions.

The classifier defined in Theorem 9.1 can be described as a continuous function of the dissimilarities to a finite set of objects. Any object can be correctly classified using a rule based on the nearest neighbors. The function $f(d(z,x);\sigma) = \sum_{x\in\omega_1} \exp\left(-d(z,x)/\sigma\right) - \sum_{x\in\omega_2} \exp\left(-d(z,x)/\sigma\right)$ is a continuous decision function assigning y to the class ω_1 iff $f(D(z,x);\sigma) > 0$ and to the class ω_2 iff $f(D(z,x);\sigma) < 0$. It performs the same classification. It classifies any object correctly if σ is sufficiently small, i.e. if $0 < \sigma < (\delta - \varepsilon)/\log\left(\max(|P_{\omega_1}|, |P_{\omega_2}|)\right)$ as for that value of σ the term with the nearest neighbor object dominates. In fact, this is a linear classifier after an exponential transformation, hence we will denote it here as *exponential* classifier.

In Fig. 9.34 the maximum within-class NN distances ($\hat{\varepsilon}_{\omega_1}$ and $\hat{\varepsilon}_{\omega_2}$) and the minimum between-class NN distances ($\hat{\delta}$) are listed. These numbers may be interpreted as the approximations of the ε-balls in the neighborhood bases $\varepsilon_{\omega_1} = \{b_{\varepsilon_{omega_1}}(x)\}$ and $\varepsilon_{\omega_2} = \{b_{\varepsilon_{omega_2}}(x)\}$ for the two classes and δ is the 'gap' between the classes as discussed in our assumption list. They indicate that a zero-error classifier may be constructed for both training and testing sets, since $\hat{\varepsilon}_{\omega_1} < \hat{\delta}$ and $\hat{\varepsilon}_{\omega_2} < \hat{\delta}$. With the increase of the number of training samples, $\hat{\varepsilon}_{\omega_1}$ and $\hat{\varepsilon}_{\omega_2}$ will decrease to zero, but $\hat{\delta}$ will approach δ. Therefore, for sufficiently large training sets, the assumptions can always be satisfied. In our case, 100 training objects in total appears to be sufficient.

	Hausd.	mod.-Hausd.
$\hat{\varepsilon}_{\omega_1}$	0.306	0.207
$\hat{\varepsilon}_{\omega_2}$	0.297	0.170
$\hat{\delta}$	0.360	0.225

Fig. 9.34: Estimated values of ε_{ω_1}, ε_{ω_2} and δ.

Random subsets of m polygons per class ($2 \le m \le 100$) are drawn from the training set, resulting in a $2m \times 2m$ dissimilarity representation. A sigmoidal classifier is then trained. Test polygons are classified on the basis of their $2m$ dissimilarities to the training objects. The experiment is repeated 20 times (different random subsets of the same training set). The averaged errors are presented in Fig. 9.33 for the Hausdorff and modified-Hausdorff distances for some chosen values

of the scaling parameter σ. Errors are compared with those of the 1-NN classifier. It shows that the linear classifier may perform better, in agreement with our earlier findings [293, 295, 301] and that zero-error classifiers on the tests sets are found for small representation sets.

Discussion. The overlap of pattern classes may be avoided by a dissimilarity representation constructed from the data if the assumptions as listed in section 9.6.1 are fulfilled. We showed that linear classifiers in dissimilarity spaces can outperform the nearest neighbor rule, even for large training set sizes for which a good performance of the NN-rule may be expected. Although the classes are separable, we cannot always succeed in our attempts to construct a zero-error solution for the test set. This result certainly depends on the distance measure, the cardinality of R in relation to the chosen classifier. At the moment, a suitable gap is constructed, a zero-error classification is possible.

The challenge, we see for the future, is to construct more locally sensitive classifiers that need just a fraction of the training examples for the representation set. Further research is needed to find out how distance measures may be constructed such that the potentially zero-error result can be obtained in practice.

9.7 Discussion

This chapter discusses classification aspects on dissimilarity representations. Dissimilarity measures with different properties: Euclidean, non-Euclidean metric and non-metric have been analyzed for this purpose. Our approaches to dissimilarity representations allow one to handle non-metric measures as well. In our experiments, we have demonstrated that simple linear or quadratic classifiers constructed either in dissimilarity or embedded spaces can significantly outperform the k-NN rule for small representation sets, irrespectively of the properties of the dissimilarity measure. We argue that, in fact, it is more important that the measure itself is discriminative for the problem than its metric or Euclidean properties.

Various prototype selection procedures have been studied for both approaches, indicating that systematic procedures, where prototypes are chosen in a supervised way, by making use of the label information, are beneficial, especially for a small number of prototypes. The selection based on the KCentres procedure can be considered as a good approach, since it is fast and works on average well. Also the prototypes (support objects) chosen by the sparse LP formulation are a candidate for a good representation set in dissimilarity spaces. To gain some control over the number of selected prototypes, the KCentres-LP procedure can be considered as it combines the advantages of both procedures. In embedded spaces, the prototypes chosen as the ones which yield the largest approximation error may be an alternative selection to the KCentres. Additionally, we have observed that for the representation sets consisting of 20% of the training objects, a random selection is advantageous in both dissimilarity and embedded spaces.

In conclusion, our results encourage us to explore meaningful dissimilarity information in new, advantageous ways, of which our proposals are an example. Under some constraints on the unambiguous labeling of objects and properties of the dissimilarity measure, the 1-NN rule will allow for zero-error recognition. This, however, might require very large training sets, hence infeasible in practice. The study of proper dissimilarity measures and suitable domain-based classifiers (like the 1-NN rule, yet less local in their decisions), instead of probabilistic reasoning is open for further research.

10. Combining

What is a committee? A group of the unwilling, picked from the unfit, to do the unnecessary.

RICHARD HARKNESS IN THE NEW YORK TIMES, 1960

Fusing information from different sources or combining individual learning strategies may be effective for designing a good-performing pattern recognition system. The basic idea (and assumption) is that an assembly of experts (say classifiers, approaches) tends to make a better decision than a single one does. This can be expected if the experts are different, possibly independent in their opinions, i.e. if their decisions are based on different principles. In classification, it means that the sets of misclassified examples should differ among the classifiers such that if an individual classifier makes a mistake, the others are able to correct it. Therefore, instead of relying on a single strategy, all suitable strategies can be used for the derivation of the final consensus.

Combining is usually done for the increase in efficiency and/or accuracy of the classification systems. The former can be met by designing hierarchical combination rules, where simple and computationally inexpensive classifiers are used first for the recognition of non-difficult objects and more advanced classifiers are applied to more specific cases later on. To increase the performance (hence also the robustness), a care should be taken that the individual (base) classifiers differ. This can be achieved, for instance, by using various feature-based representations or different training sets, e.g. sampled versions of the original one [42, 196, 367]. An important study on classifier diversity measures has been conducted by Kuncheva et al; see e.g. [231–233].

Two basic classifier combination scenarios can be distinguished. In the first case, the individual classifiers are designed on the same representation or its various subsets. The classifier outputs can be interpreted e.g. as fuzzy membership values, evidence values or posterior probabilities, or transformed as such. In the probabilistic framework, classifiers can be assumed to estimate the same posterior probability. Practically, it means that classifier ensembles are constructed in the same feature space (having the same type of features) or based on the same dissimilarity description. In the second scenario, the classifiers are built on different representations derived from physically different types of measurements (sensors), e.g. audio- and video-related representations of a biometrical identification, or from a focus on different aspects of raw measurements, e.g. representations defined for shape and color characteristics in images. Since classifiers operate in different measurement spaces, the estimated posterior probabilities do not refer to the same principal value.

A common theoretical framework for classifier combination has been discussed in [217], where *fixed* rules such as the sum rule, product rule, min rule, max rule, median rule and majority voting, are derived for general cases. Fixed combining rules operate on the classifier outputs and use some strategy (like a sum) for the final decision. Alternatively, the classifier outputs can be considered as new features on which a final output classifier is trained [100].

Many combination schemes have been proposed in feature spaces and it has been experimentally demonstrated that some of them consistently outperform the single best classifier, see e.g. [230, 266, 267]. So, we do not aim at developing new approaches, but we rather focus on the specific type of representations that we are dealing with. A learning problem can be now approached by using a set of classifiers on a chosen dissimilarity representation. In practice, however, it is advantageous to use various dissimilarity measures focusing on different data aspects or even different measurement

data, especially, if the provided information is complementary. This leads to various dissimilarity representations. As discussed above, except for combining various classifiers designed on a single representation, we can combine classifiers built on different representations. One may go even a step further to combine not the classifiers, but the representations themselves, which was our proposal [292, 305]. It is believed that discriminative properties of different[1] representations can be enhanced by a proper fusion.

In this chapter, we study both the combined dissimilarity representations, on which a single classifier is trained, as well as fixed and trained combiners applied to the outputs of the base classifiers, trained on single representations. An experimental assessment is performed for the one-class and two-class classification problems, as also published in [292, 305]. Our results show that both combining approaches allow for a significant improvement in a classification performance over the results achieved by the best single classifiers. Concerning the computational cost, the use of combined representations might be more advantageous.

In the process of combining classifiers, variability between base classifiers is essential for constructing a robust ensemble. Although various measures and many combining rules have been already suggested, the problem of designing optimal combiners is still heavily studied. The diversity between the base classifiers is therefore important. We propose to analyze the conceptual dissimilarity representation describing the pairwise diversity between the classifiers judged e.g. by their disagreements. The classifier projection space obtained as a spatial configuration of the diversities is proposed by us as a visualization tool for analyzing the differences between base classifiers and as an argument for the selection of good combining rules. This relies on our publication [304].

10.1 Combining in one-class classification problems

As studied in chapter 8, a one-class classification (OCC) problem is characterized by the presence of the target class. Additionally, non-target examples may be provided, yet, they are known to be non-representative or with unknown priors[2]. Since the non-target class is ill-defined, in complex problems, an effective set of features for the discrimination between targets and non-targets cannot be easily found. Hence, it seems appropriate to build a representation on the raw data. The dissimilarity representation, describing objects by their dissimilarities to the target examples, may be effective for such problems, since it naturally protects the target class against unseen novel examples. Optimal representations and dissimilarity measures cannot be found if one class if provided and the other is missing or badly sampled. On the other hand, when one analyzes a particular phenomenon, the model knowledge can be captured by various dissimilarity representations describing different problem characteristics. In this way each additional representation may incorporate useful information and a problem is tackled from a wider perspective. Moreover, it seems logical to follow if no convincing arguments exist to prefer one dissimilarity measure over another. Combining OCCs becomes, thereby, a natural technique needed for solving ill-defined (or unbalanced) detection problems.

Although such problems are often met in practice, representative standard data sets do not yet exist. Our procedures here are not intended for general multi-class problems for which other, more suitable, techniques exist. Our methodology is applicable to difficult problems where the target examples are provided with or without additional outlier examples. For that reason, the effectiveness of the proposed procedures is illustrated with just a single, yet complex, application. Given

[1] We want to emphasize that by *different* representations, we mean not only mathematically different formulations, but more importantly, representations based on different principles of the given phenomenon or different measurements. For instance, in the support vector learning, one may consider kernels with various nonlinearity aspects.

[2] Remember that standard two-class classifiers should be preferred if the non-target class is well represented.

autofluorescence spectra, the aim is to detect diseased mucosa in an oral cavity.

10.1.1 Combining strategies

As before, dissimilarity representations $D(T, R)$ will be interpreted in three learning frameworks: the *pretopological* approach, where the dissimilarity values directly denote the neighborhoods, the *embedding* approach, which builds on an embedded pseudo-Euclidean configuration and the *dissimilarity space* approach, where the the features are defined by the dissimilarities to particular representative objects. See sections 4.3- 4.4 for more details. Here, we assume that the representation set R consists of the target objects only. The following OCCs are considered as examples of the three approaches: the nearest neighbor data description (NNDD), defined in section 8.2.1, the generalized mean-class data description (GMDD), introduced in section 8.2.2 and the linear programming dissimilarity data description (LPDD), described in section 8.2.3.

To study the behavior of the one-class classifiers, a ROC curve [41, 389] is applied. It is a function of the true positive (target acceptance) ratio versus the false positive (outlier acceptance) ratio. To compare the performance of various classifiers, when misclassification costs are not exactly known, the AUC measure is used [40]; see section 8.1. In our experiments, we will present the AUC performance as $AUC \cdot 100$.

Two approaches are compared within this application. The first one focuses on combining dissimilarity representations into a single one, while the second approach considers a combiner operating on the outputs of the OCCs.

Combined representations. Given various feature spaces (representations), one usually combines the classifier outputs of classifiers trained on different representations. Learning from distinct dissimilarity representations (DRs) can be realized by fusing them into a new representation, followed by training a single OCC. As a result, a more powerful representation is hoped to be obtained, allowing for a better discrimination. Suppose that κ representations $D_s^{(\tau)}(T, R)$, $\tau = 1, 2, \ldots, \kappa$, all based on the same representation set R, are given. The dissimilarity measures are similarly bounded, since they have been scaled appropriately by a non-decreasing functions f_τ (such as linear, logarithmic or sigmoidal functions), i.e. $D_s^{(\tau)}(T, R) = f_\tau(D^{(\tau)}(T, R))$. This step is important, since only then the dissimilarity values can be related to each other; otherwise, we would need to compare not the direct values, but the corresponding percentiles. The dissimilarity representations can be combined into D_{comb} in the following ways:

$$D_{\text{avr}}(t_i, p_j) = \frac{1}{\kappa} \sum_{\tau=1}^{\kappa} \alpha_\tau \, D_s^{(\tau)}(t_i, p_j) \tag{10.1}$$

$$D_{\text{prod}}(t_i, p_j) = \sum_{\tau=1}^{\kappa} \log\left(1 + \alpha_\tau \, D_s^{(\tau)}(t_i, p_j)\right) \tag{10.2}$$

$$D_{\text{min}}(t_i, p_j) = \min_{\tau} \{\alpha_\tau \, D_s^{(\tau)}(t_i, p_j)\} \tag{10.3}$$

$$D_{\text{max}}(t_i, p_j) = \max_{\tau} \{\alpha_\tau \, D_s^{(\tau)}(t_i, p_j)\} \tag{10.4}$$

The nonnegative weights α_τ are additionally used to emphasize the importance of some measures. Ideally, they should be learned for the problem at hand. If an OCC is built by using both target and outlier examples, the importance of each representation can be weighted by its overall performance (the AUC measure) on the training data (or the validation data, if available). Having the AUC measures a_i, $i = 1, 2, \ldots, \kappa$ in a training (or validation) stage, the DRs can be weighted by their normalized versions $\alpha_i = a_i / \sum_{i=1}^{\kappa} a_i$. If the target examples are only provided for training or if there is no a priori knowledge, all the weights α_τ are assumed to be equal.

The DRs can be seen are combined into one representation by using fixed rules, usually applied when the outputs of two-class classifiers are combined. The reason behind the use of a combined representation is the fact that DRs can be interpreted as a collection of weak classifiers, where each of them is understood as a dissimilarity $D_s^{(\tau)}(\cdot, p_i)$ to a particular object p_i. In contrast to probabilities, a small dissimilarity value $D_s^{(\tau)}(t_j, p_i)$ is an evidence of a good 'performance', indicating here that the object t_j is similar to the target p_i. In general, different dissimilarity measures focus on different aspects of the data. Hence, each of them estimates a proximity $D_s^{(\tau)}(x, p_i)$ of an object x to the target p_i in its own way, so to say, by using partial knowledge. Combining such estimates by fixed rules is recommended [217]. So, D_{avr} yields an average proximity estimator. When, the dissimilarity measures are independent (e.g. one built on statistical and the other on structural object properties), the product combiner is of interest. Logically, both D_{avr} and D_{prod} should integrate the strengths of various representations. Here, D_{prod} is expressed by a logarithmic transformation of the product of the dissimilarities, so that very small numbers (hence numerical inaccuracies) can be avoided when close-to-zero dissimilarities are multiplied. The $\min$ operator chooses the minimal dissimilarity value $D^{(\tau)}(x, p_i)$, $\tau = 1, \ldots, \kappa$, hence the maximal evidence for an object x resembling the target t_i. The $\max$ operator works the other way around.

Combined classifiers. OCCs are in practice realized by some proximity function $f_{\text{prox}}(x, \omega_T)$ of an object x to the target class ω_T. To decide whether an object belongs to the target class or not, a threshold γ on f_{prox} should be determined. A standard way is to supply a fraction r_{fn} of (training) target objects to be rejected by the OCC (a false negative ratio) [387, 389]. So, γ is set up such that $\int \mathcal{I}(f_{\text{prox}}(x, \omega_T) > \gamma)\, d\mu(x) = r_{\text{fn}}$, where μ is some measure.

One usually combines classifiers based on their posterior probabilities. However, the OCCs do not directly estimate the posterior probabilities, since they rely on the information on a target class. Moreover, the soft (proximity-related) outputs of the OCCs trained on different representations might not be comparable. One possibility is to convert such proximities (e.g. distances to the class boundary) to the estimates of probabilities. This can be achieved e.g. by the following heuristic mapping $\hat{p}(\omega_T|x) = e^{-f_{\text{prox}}(x, \omega_T)/s}$, where s is a parameter to be fitted based on training objects, as proposed in [389]. Note that $1 - \hat{p}(\omega_T|x)$ is an estimation that x is an outlier. Consequently, standard fixed combiners, such as mean, product and majority voting, can be considered. Additionally, the raw or transformed proximity outputs can be further used as features for training a final OCC combiner.

10.1.2 Data and experimental setup

The data consist of autofluorescence spectra acquired from healthy (target) and diseased (outlier) mucosa in the oral cavity. The measurements were taken by using six different excitation wavelengths 365, 385, 405, 420, 435 and 450 nm. After preprocessing [406], each spectrum consists of 199 bins. In total, 856 and 132 spectra representing healthy and diseased tissue, respectively, were obtained for each excitation wavelength. This means that one deals with six different measurement data: $M_1, \ldots, M_6$, corresponding to six excitation wavelengths. The spectra are normalized so that they yield a unit area; see also section A.2. The measurement sets are divided into the training and testing sets in the ratio of $60 : 40$ with respect to both target and outlier class. So, there are 594 training (514 target and 80 outlier) examples and 396 testing (342 target and 52 outlier) examples.

Two cases are here investigated: combining various dissimilarity representations derived for the spectra of a *single* excitation wavelength of 365 nm (experiment I) and combining representations derived for *all* excitation wavelengths (experiment II). In both experiments, the combined representations and the combined classifiers are used. The basic difference between these lies in the use of single measurement data or multiple measurement data. So, in the experiment I, the derived dissim-

Table 10.1: Experiment I: AUC measure (AUC ·100), averaged over 30 runs, derived either for the OCCs built on the combined DRs or for fixed and trained combiners applied to the OCC outputs. All dissimilarity representations are considered for a single measurement data (the excitation wavelength of 365 nm). SO denotes support objects. The standard deviations of the means are given in parenthesis.

Single DRs: OCCs trained on a single DR						
DR	$\mathcal{C}_{3-\text{NNDD}}$	$\mathcal{C}_{\text{GMDD}}$	$\mathcal{C}_{\text{LPDD}}$	#SO	$\mathcal{C}^{\text{out}}_{\text{LPDD}}$	#SO
D_1	80.9 (0.5)	77.0 (0.6)	72.3 (0.7)	2.5	79.6 (0.5)	5.5
D_1^{der}	86.0 (0.4)	78.4 (0.5)	72.0 (0.7)	2.8	83.1 (0.5)	5.8
D_1^{2der}	86.7 (0.4)	78.1 (0.6)	78.1 (0.7)	2.9	84.2 (0.5)	5.3
D_{SAM}	81.8 (0.5)	76.6 (0.6)	68.0 (0.9)	2.9	80.2 (0.5)	6.1
D_{BH}	85.5 (0.4)	77.3 (0.5)	75.1 (0.6)	2.1	80.1 (0.5)	2.5
Ia. Combined DRs: OCCs trained on D_{comb}						
D_{comb}	$\mathcal{C}_{3-\text{NNDD}}$	$\mathcal{C}_{\text{GMDD}}$	$\mathcal{C}_{\text{LPDD}}$	#SO	$\mathcal{C}^{\text{out}}_{\text{LPDD}}$	#SO
D_{avr}	95.5 (0.2)	94.6 (0.3)	93.0 (0.3)	4.1	93.4 (0.3)	5.1
D_{prod}	95.7 (0.2)	94.9 (0.3)	93.6 (0.3)	4.6	93.6 (0.4)	7.6
D_{min}	85.6 (0.4)	84.6 (0.4)	84.7 (0.5)	14.6	87.1 (0.9)	15.7
D_{max}	93.5 (0.3)	90.6 (0.4)	84.7 (0.8)	7.1	89.0 (0.6)	10.5
Ib. Fixed combiners applied to the OCC outputs from $D_1 - D_{BH}$						
Combiner	$\mathcal{C}_{3-\text{NNDD}}$	$\mathcal{C}_{\text{GMDD}}$	$\mathcal{C}_{\text{LPDD}}$		$\mathcal{C}^{\text{out}}_{\text{LPDD}}$	
Mean	98.0 (0.2)	94.4 (0.4)	90.7 (0.6)	—	93.8 (0.3)	—
Prod	98.0 (0.1)	81.3 (0.6)	87.8 (0.5)	—	91.1 (0.3)	—
Min	93.3 (0.2)	91.0 (0.3)	88.8 (0.4)	—	92.0 (0.3)	—
Max	89.6 (0.4)	79.0 (0.5)	74.1 (0.6)	—	81.9 (0.4)	—
Voting	98.3 (0.1)	95.9 (0.2)	95.5 (0.2)	—	97.0 (0.2)	—
Ic. Trained combiners built on the LPDD outputs from $D_1 - D_{BH}$						
Combiner	$\mathcal{C}_{3-\text{NNDD}}$	$\mathcal{C}_{\text{GMDD}}$	$\mathcal{C}_{\text{LPDD}}$	#SO	$\mathcal{C}^{\text{out}}_{\text{LPDD}}$	#SO
LPDD	—	—	90.1 (0.5)	4.9	95.8 (0.2)	5.0
5-means	—	—	88.0 (0.4)	—	91.1 (0.4)	—
Parzen	—	—	90.5 (0.4)	—	94.5 (0.3)	—

Table 10.2: Experiment II: AUC measure (AUC ·100), averaged over 30 runs of single OCCs built on DRs for six measurement data sets (six excitation wavelengths). Only the worst and the best AUC values are provided. 'ALL' refers to the results on all 6×3 (six wavelengths and three measures) dissimilarity representations. The number of support objects in LPDDs varies between 2 and 7.

Single DRs: OCCs trained on single DRs for the measurement data M_1-M_6				
	D_1	D_1^{der}	D_1^{2der}	ALL
$\mathcal{C}_{3-\text{NNDD}}$	80.9 - 84.8 (0.5)	82.8 - 87.0 (0.5)	83.5 - 88.8 (0.5)	80.9 - 88.8 (0.5)
$\mathcal{C}_{\text{GMDD}}$	77.0 - 79.4 (0.7)	77.9 - 81.7 (0.6)	75.4 - 81.6 (0.6)	75.4 - 81.7 (0.7)
$\mathcal{C}_{\text{LPDD}}$	62.8 - 72.4 (0.8)	65.5 - 72.8 (0.8)	70.7 - 77.5 (0.8)	62.8 - 77.5 (0.8)
$\mathcal{C}^{\text{out}}_{\text{LPDD}}$	78.3 - 81.7 (0.9)	73.5 - 83.1 (0.7)	77.7 - 83.2 (0.6)	73.5 - 83.2 (0.6)

ilarity representations are different with respect to the measure applied to the data M_1, while in the experiment II, the computed dissimilarity representations are basically different with respect to the data sets $M_1, \ldots, M_6$, so in fact a single measure might be used for their computation. Hence, all combining scenarios (combining classifiers on the combining representations, each considered on the same or different measurement data sets) are captured in our experiments.

Five dissimilarity representations are used for the normalized spectra in experiment I (the wavelength of 365 nm); see also section 8.3.2, where some of these representations were already investigated. The first three DRs are based on the l_1 (city block) distances computed between the smoothed spectra themselves (D_1) and their first and the second order Gaussian-smoothed ($\sigma = 3$ samples) derivatives (D_1^{der} and D_1^{2der}, respectively). D_{SAM} is based on the spherical geodesic distance, also known as spectral angular mapper [239], $d_{\text{SAM}}(\mathbf{x}, \mathbf{y}) = \arccos(\mathbf{x}^T\mathbf{y})$. D_{BH} is based on

the Bhattacharyya distance, a divergence measure between two probability distributions as defined in section 5.2. This measure is applicable, since the normalized spectra can be considered as unidimensional histogram-like distributions. As a result, all dissimilarity representations emphasize different aspects of the spectra.

In experiment II, dissimilarity representations are derived for six measurement data: M_1-M_6. Only the first three measures D_1, D_1^{der} and D_1^{2der}, described above are used.

As mentioned in the previous section, three base one-class classifiers are considered: the nearest neighbor data description, the generalized mean-class data description (GMDD) and the linear programming dissimilarity data description (LPDD). The classifiers will be denoted as $\mathcal{C}_{3-\text{NNDD}}$, $\mathcal{C}_{\text{GMDD}}$ and $\mathcal{C}_{\text{LPDD}}$. Additionally, since the LPDD is able to incorporate the information on outlier examples, if they are used, the resulting classifier will be denoted as $\mathcal{C}_{\text{LPDD}}^{\text{out}}$.

To describe the experiments more clearly, the following division is introduced:

- Ia or IIa denotes the experiments with the combined representation D_{comb} for which a single OCC is trained. The dissimilarity representations are first scaled by the largest training value and then combined into D_{comb} according to (10.4). Although the weights were estimated based on the AUC performance on the training set (using outlier objects), they yielded little variability. So, for simplicity, equal weights are assumed (we have also found experimentally that the results for a weighted average are not significantly different than from a usual average). In the experiment II, for each measure considered, six dissimilarity representations are combined over various measurement data $M_1, \ldots, M_6$ and in the end, all 18 DRs (three measures and six data sets) are combined, as well.

- Ib or IIb denotes the experiments with the fixed combiners applied to the OCC outputs. The OCC outputs are first converted to the estimates of posterior probabilities, as described in section 10.1.1 and then traditional mean, product, min, max and voting rules are used for the final decision.

- Ic or IIc denotes the experiments with the trained combiners applied to the OCC outputs. Here, we like to proceed with the exact (proximity-related) OCC outputs. To design a trained combiner, we focus on the LPDDs as the base classifiers. Let us denote, for convenience, the dissimilarity representations as $D^{(\tau)}$, $\tau = 1, \ldots \kappa$. Each LPDD is determined by a hyperplane $H^{(\tau)}$ in a dissimilarity space $D^{(\tau)}(T, R)$. The distances to the hyperplane are realized by weighted linear combinations of the form $d_H^{(\tau)}(t_i) = \sum_{w_j^{(\tau)} \neq 0} w_j^{(\tau)} D^{(\tau)}(t_i, p_j) - \rho$. As a result, one may construct an $n \times \kappa$ dissimilarity matrix $D_H = [d_H^{(1)}(T), \ldots, d_H^{(\kappa)}(T)]$ expressing the non-normalized signed distances between the n training objects and κ base classifiers. Hence, again an OCC can be trained on D_H. This means that an OCC becomes a trained combiner now, retrained by using the same training set (ideally, an additional validation set should be used). The LPDD can be used again as a combiner, as well as some other feature-based OCCs. (Although the values of D_H become negative for the targets and positive for the outliers, they are bounded, so the LPDD can be constructed based on the same principles.) Two other standard data descriptions (OCCs) are used, where a proximity of an object to the target class relies either on the information to the chosen k-mean vectors (k-means) or a density estimation by the Parzen kernels [387] (Parzen), respectively. They interpret the LPDD outputs in a vector space.

The experiment itself (I or II) decides whether single or multiple measurement data are used.

Table 10.3: Experiment II: AUC measure (·100), averaged over 30 runs, derived either for the OCCs built on the combined DRs or for fixed and trained combiners applied to the OCC outputs. The representations and classifiers are combined over six measurement sets $M_1 - M_6$ (related to six excitation wavelengths) and a fixed dissimilarity representation. 'ALL' refers to the results on all 6×3 (six wavelengths and three measures) representations. SO denotes support objects. The standard deviations of the means are given in parenthesis.

IIa. Combined DRs: OCCs trained on D_{comb} combined over $M_1 - M_6$								
$\mathcal{C}_{3-\text{NNDD}}, D_{\text{comb}}$	D_1		D_1^{der}		D_1^{2der}		ALL	
D_{avr}	97.7 (0.2)	—	97.6 (0.2)	—	96.8 (0.1)	—	99.6 (0.0)	—
D_{prod}	97.7 (0.2)	—	97.7 (0.2)	—	96.9 (0.1)	—	99.7 (0.0)	—
D_{min}	89.7 (0.4)	—	89.5 (0.4)	—	89.8 (0.3)	—	90.4 (0.4)	—
D_{max}	96.8 (0.2)	—	96.0 (0.2)	—	94.4 (0.2)	—	97.5 (0.1)	—
$\mathcal{C}_{\text{GMDD}}, D_{\text{comb}}$	D_1		D_1^{der}		D_1^{2der}		ALL	
D_{avr}	97.2 (0.2)	—	97.2 (0.2)	—	96.0 (0.1)	—	99.6 (0.0)	—
D_{prod}	97.3 (0.2)	—	97.4 (0.2)	—	96.3 (0.1)	—	99.6 (0.0)	—
D_{min}	96.4 (0.2)	—	93.5 (0.3)	—	90.9 (0.4)	—	97.7 (0.2)	—
D_{max}	93.7 (0.3)	—	93.3 (0.3)	—	91.0 (0.3)	—	95.3 (0.2)	—
$\mathcal{C}_{\text{LPDD}}, D_{\text{comb}}$	D_1	#SO	D_1^{der}	#SO	D_1^{2der}	#SO	ALL	#SO
D_{avr}	96.6 (0.3)	5.2	97.1 (0.3)	4.2	95.6 (0.2)	3.6	99.5 (0.1)	4.3
D_{prod}	96.9 (0.2)	5.7	97.2 (0.3)	4.0	95.8 (0.2)	3.7	99.6 (0.0)	4.9
D_{min}	95.1 (0.3)	42.2	94.0 (0.3)	33.5	92.0 (0.5)	27.7	96.1 (0.2)	50.0
D_{max}	89.6 (0.6)	11.2	89.8 (0.8)	9.0	85.4 (0.9)	8.9	92.3 (0.4)	11.1
$\mathcal{C}^{\text{out}}_{\text{LPDD}}, D_{\text{comb}}$	D_1	#SO	D_1^{der}	#SO	D_1^{2der}	#SO	ALL	#SO
D_{avr}	96.7 (0.1)	5.1	97.1 (0.1)	4.0	95.6 (0.1)	3.6	99.5 (0.0)	4.5
D_{prod}	96.8 (0.1)	7.3	97.2 (0.2)	5.8	95.8 (0.1)	5.0	99.6 (0.1)	6.6
D_{min}	95.1 (0.3)	42.7	94.1 (0.2)	34.8	92.2 (0.2)	29.5	96.3 (0.1)	50.4
D_{max}	89.5 (0.2)	11.2	90.8 (0.2)	8.4	86.3 (0.4)	8.0	92.9 (0.1)	10.3
IIb. Fixed combiners applied to the OCC outputs								
$\mathcal{C}_{3-\text{NNDD}}$ outputs	D_1		D_1^{der}		D_1^{2der}		ALL	
Mean	97.8 (0.1)	—	98.0 (0.1)	—	98.2 (0.2)	—	99.6 (0.1)	—
Prod	98.6 (0.1)	—	98.5 (0.1)	—	98.6 (0.1)	—	99.6 (0.0)	—
Voting	97.6 (0.1)	—	98.7 (0.1)	—	98.6 (0.1)	—	99.8 (0.0)	—
$\mathcal{C}_{\text{GMDD}}$ outputs	D_1		D_1^{der}		D_1^{2der}		ALL	
Mean	94.3 (0.4)	—	94.2 (0.3)	—	94.3 (0.3)	—	98.3 (0.2)	—
Prod	96.0 (0.2)	—	96.4 (0.1)	—	96.7 (0.1)	—	99.7 (0.0)	—
Voting	96.7 (0.2)	—	97.4 (0.1)	—	97.6 (0.1)	—	99.6 (0.1)	—
Fixed (IIb) and trained (IIc) combiners applied to the $\mathcal{C}_{\text{LPDD}}$ outputs								
Combiner	D_1	#SO	D_1^{der}	#SO	D_1^{2der}	#SO	ALL	#SO
Mean	92.7 (0.4)	—	92.9 (0.4)	—	91.8 (0.3)	—	94.5 (0.2)	—
Prod	95.7 (0.9)	—	95.7 (1.0)	—	95.7 (0.5)	—	98.7 (0.6)	—
Voting	95.7 (0.4)	—	96.8 (0.2)	—	97.9 (0.1)	—	99.3 (0.1)	—
LPDD	89.3 (0.4)	5.9	91.5 (0.4)	5.9	94.6 (0.2)	5.9	96.6 (0.3)	13.2
5-means	90.5 (0.3)	—	93.0 (0.3)	—	92.7 (0.3)	—	97.2 (0.1)	—
Parzen	92.1 (0.3)	—	94.4 (0.3)	—	94.9 (0.3)	—	98.2 (0.1)	—
Fixed (IIb) and trained (IIc) combiners applied to the $\mathcal{C}^{\text{out}}_{\text{LPDD}}$ outputs								
Combiner	D_1	#SO	D_1^{der}	#SO	D_1^{2der}	#SO	ALL	#SO
Mean	93.7 (0.4)	—	93.6 (0.5)	—	95.6 (0.4)	—	98.8 (0.3)	—
Prod	95.4 (0.8)	—	96.2 (0.9)	—	97.2 (0.5)	—	99.5 (0.6)	—
Voting	96.3 (0.4)	—	96.8 (0.2)	—	98.0 (0.1)	—	99.5 (0.1)	—
LPDD	95.7 (0.2)	6.0	96.5 (0.2)	6.0	95.8 (0.2)	6.0	99.1 (0.1)	16.3
5-means	93.8 (0.3)	—	95.1 (0.2)	—	94.2 (0.3)	—	97.3 (0.1)	—
Parzen	95.5 (0.2)	—	96.8 (0.2)	—	96.2 (0.2)	—	98.9 (0.1)	—

10.1.3 Results and discussion

The following observations can be made from experiment I; see Table 10.1. Both an OCC trained on the combined representations (Ia) and a trained or fixed combiner on the OCC outputs (Ib and

Ic) improve the AUC measure of each single OCC trained on the considered dissimilarity representations $D_1, D_1^{\text{der}}, D_1^{\text{2der}}, D_{\text{SAM}}$ and D_{BH}. Concerning the combined representations (Ia), the element-wise average and product combiners perform better than the min and max operators. The 3-NNDD seems to give the best results; they are somewhat better than the ones obtained from the GMDD and the LPDD trained on $D_{\text{comb}}(T, R)$. However, in a testing stage, both the 3-NNDD and the GMDD rely on computing dissimilarities to all 514 objects of the representation set R, while the LPDD is based on maximum 16 support objects (see $\#SO$ in Table 10.1; the support objects are determined during training). Hence, the LPDD can be recommended from the computational efficiency point of view.

The fixed and trained combiners on the OCC outputs perform well. In fact, the best overall results for the base OCCs considered (the 3-NNDD, the GMDD and the LPDD) are reached for the fixed voting combiner. However, combiners require more computations; first individual OCCs are trained on each representation and then, the final combiner is applied. Yet, if some outliers are available for training the LPDD $\mathcal{C}_{\text{LPDD}}^{\text{out}}$, then the testing stage becomes inexpensive as it relies on the computation of the dissimilarities to 27 objects (the sum of the support objects found for each representation separately).

Concerning the experiment II, where different measurement data set are considered, the following observations can be made from the analysis of Tables 10.2 and 10.3. Again, both an OCC trained on the combined representations (IIa) by the average and product, and a fixed (IIb) or trained (IIc) combiner on the OCC outputs significantly improve the AUC performance (by more than 10%) of each single OCC (compare to the results in Table 10.2). Since the spectra derived from various wavelengths describe different information, an OCC built on their combined representation (where a single measure is used to derive DRs over six measurement data sets) allows for reaching a somewhat better AUC performance than an OCC built on the combined representation (where various dissimilarity measures are used to define the DRs) considered for a single wavelength. This consistent behavior can be observed by comparing the results of IIa in Table 10.3 and Ia in Table 10.1.

The fixed voting rule applied to the OCCs outputs (IIb) gives mostly the overall best results (an exception holds for the dissimilarity representation D_1 and the LPDDs as base classifiers). The trained combiners (IIc) on the LPDD outputs are somewhat worse (possibly due to overtraining) than the fixed voting combiner, however, they are similar to the results of the mean combiner. From the computational point of view, either an LPDD trained on the combined dissimilarity representation (IIa) or a fixed voting combiner on the LPDD outputs (IIb) should be preferred.

By using all six measurement data sets and three dissimilarity measures (so 18 representations in total), all the combining procedures yield a nearly perfect performance, i.e. mostly 99.5% or more. Such results are presented in the column denoted as 'ALL' in Table 10.3.

10.1.4 Summary and conclusions

Here we study approaches of detecting one-class phenomena based on a set of training examples, performed in an unknown or ill-defined context of alternative phenomena. Since a proximity of an object to a class is essential for such a detection, dissimilarity representations (DRs) can be used as the ones which focus on the object-to-target dissimilarities. When considering a number of different dissimilarity measures, the problem can be described more accurately by combining various representations. Three different one-class classifiers (OCCs) are used: the NNDD (based on the nearest neighbor information), the GNMD (a generalized mean classifier in an underlying pseudo-Euclidean space) and the LPDD (a hyperplane in the corresponding dissimilarity space), which offers a sparse solution. The additional advantage of using an LPDD is that a sparse solution is obtained, which means that in a testing stage, dissimilarities to a few objects need to be computed

to make a decision.

Dissimilarity representations directly encode evidences for objects which lie in close or far neighborhoods of the target objects. Hence, they can naturally be combined (after a proper scaling) into one representation, e.g. by an element-wise averaging. This is beneficial, since only one OCC can be trained, ultimately. From our study on the detection of diseased mucosa in oral cavity, it follows that dissimilarity representations combined either by average or product have a larger discriminative power than any single one. We also conclude that by combining information of representations derived for spectra of different excitation wavelengths is somewhat more beneficial than by using only one fixed wavelength, yet different dissimilarity measures. In the former case, all the OCCs on the combined representations performed about the same, while in the latter case, the LPDD trained on the targets seemed to be worse. The fixed OCC combiners have also been applied to the outputs of single OCCs. The overall best results are reached for the majority voting rule. The trained OCC combiners, applied to the outputs of single LPDDs, performed well, yet worse than the voting rule. Concerning the computational issues, either the LPDD built on the combined representations or the voting combiner applied to the LPDD outputs are recommended. Further studies on new problems need to be conducted in the future.

10.2 Combining in standard two-class classification problems

Selecting a good dissimilarity measure becomes an issue for the classification problem at hand. When considering a number of different possibilities for building a dissimilarity representation, there might be no convincing arguments to prefer one measure over another. Therefore, an interesting question is whether combining dissimilarity representations is beneficial. As in the one-class classification, two combining possibilities are investigated here. In the first case, the base classifiers (the NLC or the NN rule) are found on each dissimilarity representation and then combined into one decision rule. If the representations differ in character, a more powerful decision rule may be constructed by their combining. In the second case, instead of combining classifiers, representations are combined to create a new representation for which only one classifier has to be trained. Our experiments are conducted on a few dissimilarity representations derived for the NIST handwritten digit set. They demonstrate that when the dissimilarity representations are of different nature, a much better classification performance can be reached by their combination than by the use of individual representations only.

10.2.1 Combining strategies

To construct a decision rule on dissimilarities, the training set T of the cardinality N and the representation set R of the cardinality n will be used. In the learning process, a classifier is built on the $N \times n$ dissimilarity matrix $D(T, R)$. The information on a test set T_{te} of t new objects is given by their dissimilarities to R, i.e. as an $t \times n$ matrix $D(T_{te}, R)$. Two classifiers are used: the NLC (normal density based linear classifier) in a dissimilarity space and the 1-NN rule directly applied to the dissimilarities.

Assume that we are given the representation set R and κ different dissimilarity representations $D^{(1)}(T, R)$, $D^{(2)}(T, R)$, ..., $D^{(\kappa)}(T, R)$. Our idea is to combine base classifiers constructed on distinct representations. It is important to emphasize that the dissimilarity representations should have different character, otherwise they convey similar information and not much can be gained by their fusion.

Two cases are here considered. In the first one, a single NLC is trained in each dissimilarity space $D^{(i)}(T, R)$, $i = 1, 2, \ldots, \kappa$ separately and then all of them are combined. In the second case, the 1-NN rule is also included. The 1-NN rule and the NLC differ in their decision-making process and

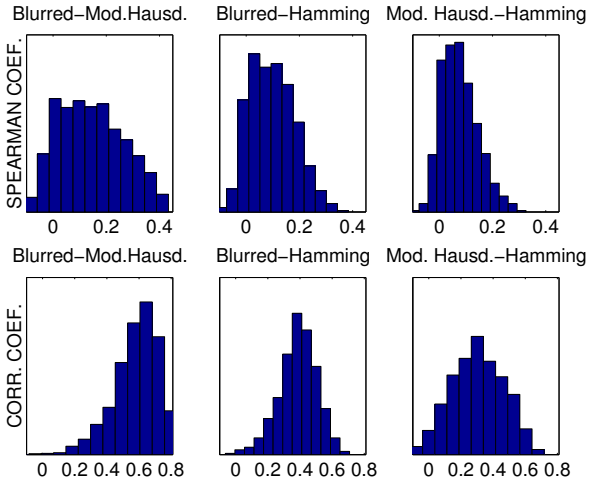


Fig. 10.1: Spearman coefficients (top) and traditional correlation coefficients (bottom) used for pairwise comparisons of the dissimilarity representations.

their assignments. The 1-NN method operates on the dissimilarity information in a rank-based way, while the NLC approaches it in a feature-based way. Although for small representation sets, the recognition accuracy of the 1-NN method is often worse than of the NLC in a dissimilarity space [293, 296, 301], still better results may be obtained when both types of classifiers are included in the combining scheme. In our approach, we will limit ourselves to the fixed rules operating on posterior probabilities. For the NLC, the posterior probabilities are based on normal density estimates, while for the 1-NN method, they are estimated from distances to the nearest neighbor of each class [111].

Another approach to learning from a number of distinct dissimilarity representations is to combine them into a new one and then train a single classifier. As a result, a more powerful representation may be obtained, allowing for a better discrimination. The first method for creating a new representation relies on building an extended representation D_{ext}, in a matrix notation given by:

$$D_{\text{ext}}(T,R) = [D^{(1)}(T,R) \quad D^{(2)}(T,R) \quad \dots \quad D^{(\kappa)}(T,R)]. \tag{10.5}$$

It means that a single object is now characterized by κn dissimilarities from κ various representations, but still related to the same representation objects. The requirement of having the same prototypes is not crucial, however, for the sake of simplicity, we will keep R fixed.

In the second method, all distances of different representations are first scaled appropriately by a non-decreasing functions f_τ, i.e. $D_s^{(\tau)}(T,R) = f_\tau(D^{(\tau)}(T,R))$, $\tau = 1,\dots,\kappa$, to guarantee that they all take values in a similar range. This is necessary, since otherwise the dissimilarity values coming from different representations could not be directly compared. The combined representation D_{comb} is then created, e.g. by computing their weighted average $D_{\text{avr}}(T,R) = \sum_{\tau=1}^{\kappa} \alpha_\tau D_s^{(\tau)}(T,R)$ or any other way as presented in (10.4). Some other possibilities for building a combined kernel for the support vector classifier are discussed in [79, 277].

10.2.2 Experiments on the handwritten digit set

To investigate the combining procedure, a two-class classification problem between the NIST handwritten digits 3 and 8 [420], originally represented as 128×128 binary images is considered; see also section A.2. Three dissimilarity measures are used: Hamming, modified-Hausdorff and 'blurred' Euclidean, resulting in the following representations: D_H, D_{MH} and D_B, correspondingly. The Hamming distance counts the number of pixels which disagree. The non-metric modified-Hausdorff distance, Def. 5.6, is found useful for template matching purposes [93]; To design D_B, images are

first blurred (smoothed) with a Gaussian kernel of the standard deviation of 8 pixels. Then the Euclidean distance is computed between such blurred versions. Such a smoothing process is meant to make the distances be more robust against small tilting, shifting and change in thickness. The resulting distances are called 'blurred' Euclidean.

Each of the dissimilarity measures uses the image information in a particular way: binary information, contours or blurring. From the process of the construction, it follows that our dissimilarity representations differ in properties. To prove, however, their different characteristics, the Spearman rank correlation coefficient is used to rank the distances computed to each prototype. For two variables $\mathcal{X}$ and $\mathcal{Y}$, the Spearman rank correlation[3] is computed as:

$$R_s(\mathcal{X},\mathcal{Y}) = 1 - 6\,\frac{\sum_i (r_i(\mathcal{X}) - r_i(\mathcal{Y}))^2}{N(N^2-1)}, \tag{10.6}$$

where N is the number of values in both variables and R_i is the i-th rank. Basically, we want to show that the ranks differ between the representations. Therefore, for each pair of the representations, D_H-D_{MH}, D_{MH}-D_B and D_B-D_H, the Spearman coefficients between the dissimilarity ranks to all the representation objects are computed. For instance, for the pair of D_B and D_{MH}, the Spearman coefficients are computed between $D_B(\cdot, p_i)$ and $D_{MH}(\cdot, p_i)$ for every $p_i \in R$. Histograms of their distributions are shown in Fig. 10.1. The coefficients vary between -0.05 and 0.4, where most of them are smaller than 0.3, which implies that the ranks significantly differ. This suggests that the 1-NN rule will behave differently on each representation.

The traditional Pearson correlation coefficient is used to check whether the dissimilarity spaces of the individual representations (and, therefore, linear classifiers built there) are different (high positive values indicate a linear correlation). Such correlation values are higher than those given by the Spearman rates, since now the vectors of dissimilarities are considered, which cannot completely vary from one representation to another. On average, the correlations are found to be (see Fig. 10.1: 0.56 between the blurred and modified Hausdorff distance representations, 0.39 between the blurred and Hamming representations and 0.28 between the modified Hausdorff and Hamming representations. In the end, most coefficients are smaller than 0.7, thereby, they indicate only weak linear dependencies. Consequently, we can say that our dissimilarity representations differ in character.

The experiments are performed 30 times and the results are averaged. In a single experiment, the data, consisting of 1000 objects per class, are randomly split into two equally-sized sets: the design set L and the test set T_{te}. Both L and T_{te} contain 500 examples per class (so 1000 objects in total). The test set is kept constant, while L serves for the selection of training sets with various sizes. These are $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$ and $\mathcal{T}_4 = L$ of the following cardinalities per class: 50, 100, 300 and 500, respectively. For each training set, the experiments are conducted for an increasing representation set R. Here, for simplicity, R is chosen to be a random subset of the training set, where both classes are equally represented. In each run, every training dissimilarity representation $D^{(\tau)}$ is scaled by the averaged dissimilarity d_τ, which also serves for scaling the test dissimilarity representation. This is necessary to guarantee that the dissimilarities express similar values.

10.2.3 Results

Training sets of different sizes are considered to investigate the influence of the training size on our combining approaches. The results are presented in Fig. 10.2 and 10.3. All plots show curves of the generalization (test) error averaged over 30 runs. Each error curve is a function of the cardinality

[3] In fact, the Spearman rank correlation is the classical Pearson correlation coefficient $R(\mathcal{X},\mathcal{Y}) = \frac{cov(\mathcal{X},\mathcal{Y})}{\sqrt{var(\mathcal{X})\,var(\mathcal{Y})}}$ when the variables are converted to the ranks [229]. A simpler formula is used for the computation.

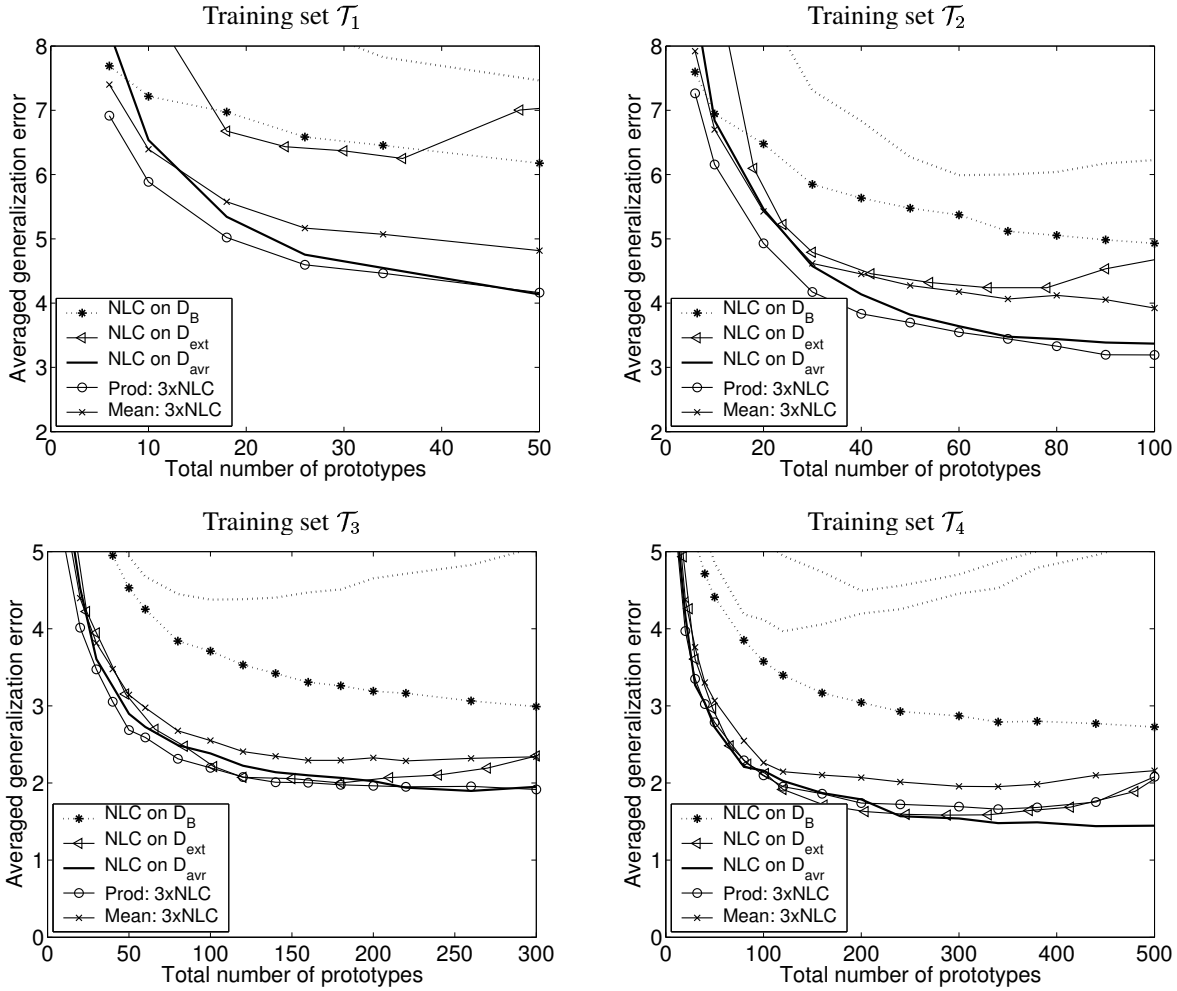


Fig. 10.2: Combining for the NIST-38 problem: the averaged classification error (in %) of the combined NLC (by product or mean) and of a single NLC on the combined representations (D_{avr} or D_{ext}) as a function of the total number of prototypes. Three dissimilarity representations are combined: D_H, D_{MH} and D_B. The results of the NLC trained on single DRs are plotted in dots. If there are less than three such curves in a plot, it means that the errors are larger than the presented scales. The best performance of the NLC is achieved for D_B. The standard deviations of the presented results are: $\mathcal{T}_1$: 0.23% on average and maximum 0.58%, $\mathcal{T}_2$: 0.19% on average and maximum 0.56%, $\mathcal{T}_3$: 0.20% on average and maximum 0.44% and $\mathcal{T}_4$: 0.09% on average and maximum 0.51%. Note the scale differences.

of the representation set R, where R is a random subset of T, not larger than half of the training size. Since our goal is to improve the performance of the NLC and 1-NN by combining, all the results are presented with respect to their performance on the single representations. Considering single classifiers, it appears that the NLC consistently outperforms the 1-NN rule for the training sets $\mathcal{T}_1$ - $\mathcal{T}_4$. The best results of a single NLC are reached on the blurred Euclidean dissimilarity representation.

Fig. 10.2 presents the generalization errors for the NLC in dissimilarity spaces. It shows the error curves obtained for three individual NLCs combined by the mean and product rules and the error curves of a single NLC operating on a combined dissimilarity representation constructed from D_B, D_{MH} and D_H. Two cases are here considered for the latter: an extended representation D_{ext}, (10.5), and the average representation D_{avr} with equal weights (other combined representations, as mentioned in (10.4) give worse results). To keep to total number of prototypes the same, if D_{avr} is defined on $|R|$ objects, the extended representation D_{ext} is in fact based on $|R|/3$ different objects to guarantee the same total number of prototypes, i.e. the dimensionality of the dissimilarity space.

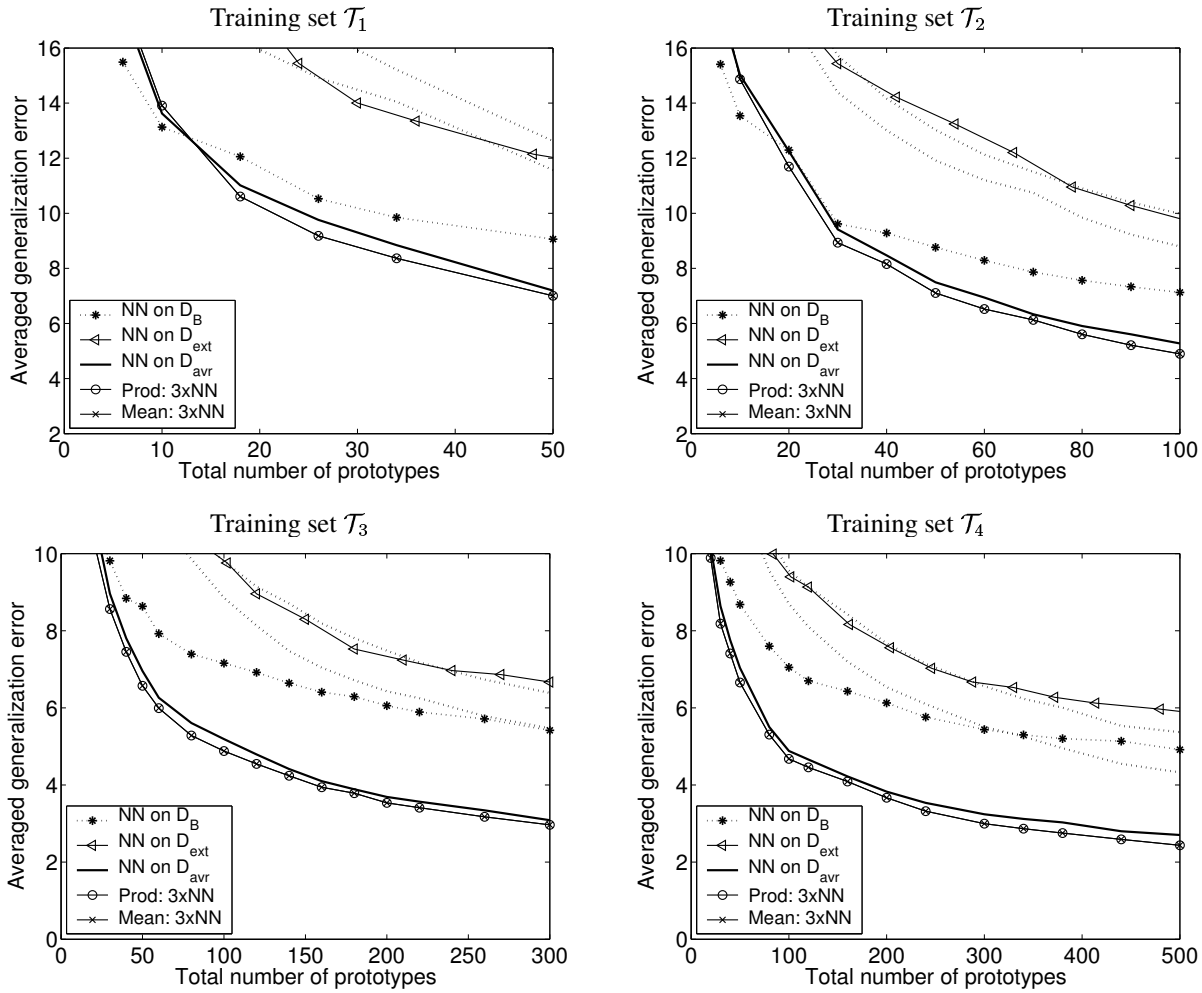


Fig. 10.3: Combining for the NIST-38 problem: the averaged classification error (in %) of the combined 1-NN rules (by product or mean) and of a single NLC on the combined representations (D_{avr} or D_{ext}) as a function of the total number of prototypes. Three dissimilarity representations are combined: D_H, D_{MH} and D_B. The results of the 1-NN trained on single DRs are plotted in dots. If there are less than three such curves in a plot, it means that the errors are larger than the presented scales. The best performance of the 1-NN is mostly achieved for D_B. The standard deviations of the presented results are: $\mathcal{T}_1$: 0.38% on average and maximum 1.18%, $\mathcal{T}_2$: 0.31% on average and maximum 1.72%, $\mathcal{T}_3$: 0.21% on average and maximum 1.08% and $\mathcal{T}_4$: 0.19% on average and maximum 1.22%. The largest standard deviations appear for the mean and product combiners. Note the scale differences.

Hence, it is more important for D_{ext} than to D_{avr} to have good prototypes selected.

From Fig. 10.2, we can conclude that the product combiner is better than the mean combiner. (Other fixed combiners have been also considered, but they were not better than the product combiner.) Also for smaller training sets ($\mathcal{T}_1$ and $\mathcal{T}_2$) and smaller representation sets, the product combiner is the best. For larger training sets ($\mathcal{T}_3$ and $\mathcal{T}_4$) and larger representation sets, a single NLC on D_{avr} performs similarly or better than the product combiner. The performance of the NLC on D_{ext} seems to suffer either from little variability among the prototypes or from not sufficiently discriminative prototypes when the training set is small. It, however, improves for larger training sets. In the latter case, for an appropriate number of prototypes, it may be as good as the product combiner.

Fig. 10.3 presents the generalization errors for the 1-NN rule. obtained for combining three individual 1-NN classifiers by the mean and product rules and the error curves of a single 1-NN built on a combined dissimilarity representation. Operating on posterior probabilities is motivated by the intention of combining both the NLC and the 1-NN method further on. Although the estimation

of these probabilities is rather crude for the 1-NN method, it still allows for an improvement of the combined rules. In all cases, the combination by the mean, or product operation gives much better results than each individual 1-NN rule. The larger, both training and representation sets, the more indicative gain in accuracy.

When a classifier ensemble consist of three NLCs and three 1-NN rules trained on D_B, D_{MH} and D_H, the product combiner is still somewhat better than the mean combiner for smaller training sets, however, they behave similarly for larger training sets. The overall results are nearly the same as presented for the product combiner in Fig. 10.2, therefore, we judge that no new plots are needed.

In summary, the mean and product combining rules perform significantly better than the individual 1-NN and NLC constructed on dissimilarity representations. In general, the dissimilarity representations tend to be independent and, therefore, the product rule based on the NLCs is expected to give better results than the mean rule [386]. Consequently, the product combiner is preferred. For the 1-NN rule, the posterior probabilities are very rough estimates from distances to the nearest neighbor and do not depend on the dimensionality of the problem. Therefore, both combiners perform about the same.

10.2.4 Conclusions

Combining a number of dissimilarity representations may be of interest when there is no clear preference for a particular one. It can be beneficial when DRs emphasize different data characteristics. This is illustrated by a two-class recognition problem between the NIST digits 3 and 8 for three dissimilarity representations: Hamming D_H, modified Hausdorff D_{MH} and blurred Euclidean D_B.

We have analyzed two possibilities of combining such information, either by combining classifiers or by combining representations themselves. In the first approach, individual classifiers are found for each representation separately and then they are combined into one rule. Our experiments show that the product combining rule works well, especially for larger representation sets (with respect to the training size). This might be explained by not very high correlations between dissimilarity spaces (especially for smaller representation sets), hence possible independence between the NLCs constructed there. Adding the 1-NN rules to the classifier ensemble improves somewhat the mean combiner, but not the product combiner.

In the second approach, dissimilarity representations are combined into a new one on which a single NLC is constructed. They are scaled so that their mean values become equal and then averaged out, resulting in the representation D_{avr}. The NLC on D_{avr} significantly outperforms the individual NLCs. As a reference, the extended representation D_{ext} is also considered, (10.5). The NLC on such a representation reaches a similar performance as on D_{avr}, but for larger training sets. In general, we conclude that for this problem the product combiner of three NLCs is recommended for small training sets, while the single NLC trained on D_{avr} is suggested for larger training sets.

10.3 Classifier projection space - a tool for investigating the classifier diversity

In this section some standard classifier ensembles designed in feature spaces are considered. The base classifiers are used to build a conceptual dissimilarity representation describing classifier pairwise diversities. Such a representation serves for the construction of a classifier projection space (CPS), based on an (approximate) embedding of the classifier diversities, which is a tool for analyzing the differences between base classifiers and it can be used as an argument for the selection of some combining rules. The rationale behind is explained below.

When a classification problem is too complex to be solved by training a single (advanced) classifier,

the problem may be divided into subproblems. They can be solved one per time by training simpler base classifiers on subsets or variations of the problem. In the next stage, these base classifiers are combined. Many strategies are possible for creating subproblems as well as for constructing combiners [237]. Base classifiers are expected to be different since they should deal with different subproblems or operate on different variations of the original problem. It is not useful to store and use sets of classifiers that perform almost identically. If they differ somewhat, as a result of estimation errors, averaging their outputs may be worthwhile. If they differ considerably, e.g. by approaching the problem in independent ways, the product of their estimated posterior probabilities may be a good rule [217]. Having significantly different base classifiers in a collection is important since this gives raise to essentially different solutions. The concept of diversity is, thereby, crucial [233]. There are various ways to describe the diversity, usually producing a single number attributed to the whole collection of base classifiers. Here, we will use it differently.

What we are looking for is a method of combining base classifiers that is not sensitive to their defects resulting from the way their collection is constituted. We want to use the fact that we deal with classifiers and not with arbitrary functions of the original features. To achieve that, we propose to study the collection of classifier pairwise differences, an $n \times n$ conceptual dissimilarity matrix D, before combining them into an output combiner. The dissimilarity value may be based on one of the diversity measures [233], like the disagreement [196]. Such a matrix D can be embedded into a space $\mathbb{R}^m$, $m < n$, in a (non-)linear way. The classifiers are then represented as a set of n points in $\mathbb{R}^m$ such that their pairwise Euclidean distances preserve the original dissimilarities given by D. It is only possible when D is Euclidean, so there might be a need for an approximate embedding, where a space of a lower, fixed dimensionality is determined for an optimal approximation of D. We call this a Classifier Projection Space (CPS).

If the CPS is two-dimensional, it can be visualized. Then, the collection of base classifiers, various combiners and, if desired, also other decision rules can be presented in a single 2D plot. Here we will choose the classical scaling, section 3.3.1 and Sammon mapping S_0, section 3.4.2, as the methods to construct the CPS. Yet, other techniques described in chapter 6 can be used as well.

10.3.1 Construction and the use of the Classifier Projection Space

Let us assume n classifiers trained on a training set. The CPS will be constructed based on the evaluation (test) set. For each pair of classifiers, their diversity value is determined, by using an evaluation set. This gives an $n \times n$ symmetric diversity matrix D. To take into account the original characteristics of the base classifier outputs, a suitable diversity measure should be chosen to establish the basic difference between classifiers. Studying the relations between classifiers in the CPS allows us for gaining a better understanding than by using the mean diversity only. The latter might be irrelevant e.g. for an ensemble consisting of both similar and diverse classifiers, where their contributions might average out.

The joint output of two classifiers, $\mathcal{C}_i$ and $\mathcal{C}_j$ can be related by counting the number of occurrences of correct (1) or wrong (0) classification. Then the counters used for binary features as described in section 5.1 can be adopted appropriately such that a is the number of correct classifications for both $\mathcal{C}_i$ and $\mathcal{C}_j$, etc. This requires the knowledge of correct labels, which might not be available, e.g. for a test set. This can be avoided when the classifier assignments are compared. Many known (dis)similarity measures can be used; examples are given in section 5.1; see also [72, 233]. Here, we will consider a simple diversity measure, the

		$\mathcal{C}_j$	
		class 1	class 2
$\mathcal{C}_i$	class 1	a_{ij}	b_{ij}
	class 2	c_{ij}	d_{ij}

Fig. 10.4: $\mathcal{C}_i$ versus $\mathcal{C}_j$: the counters.

Table 10.4: Disagreement values, $D \cdot 100$, between classifiers built on the morphological features of the MFEAT set; since D is symmetric, only the upper part is presented.

	NMSC	NLC	NUC	NQC	1-NN	k-NN	Parzen	SVC-1	SVC-2	DT	ANN20	ANN50
NMC	47.1	47.3	43.4	50.3	53.5	30.9	24.1	63.1	71.4	50.4	77.5	72.8
NMSC	-	13.7	43.3	30.2	54.0	46.9	46.8	54.7	59.9	21.9	71.5	69.1
NLC	-	-	48.5	24.1	53.9	49.0	48.4	53.0	58.0	24.3	72.1	69.1
NUC	-	-	-	53.8	64.8	54.5	50.5	55.5	76.8	54.7	72.1	75.2
NQC	-	-	-	-	53.8	39.5	39.9	50.3	65.5	31.5	67.5	57.1
1-NN	-	-	-	-	-	48.5	49.5	65.5	78.7	53.9	77.7	77.0
k-NN	-	-	-	-	-	-	7.5	56.5	75.5	48.0	68.1	72.2
Parzen	-	-	-	-	-	-	-	56.7	73.2	48.1	68.8	71.2
SVC-1	-	-	-	-	-	-	-	-	79.1	54.2	36.7	89.9
SVC-2	-	-	-	-	-	-	-	-	-	65.0	84.2	86.7
DT	-	-	-	-	-	-	-	-	-	-	70.1	71.4
ANN20	-	-	-	-	-	-	-	-	-	-	-	100.0

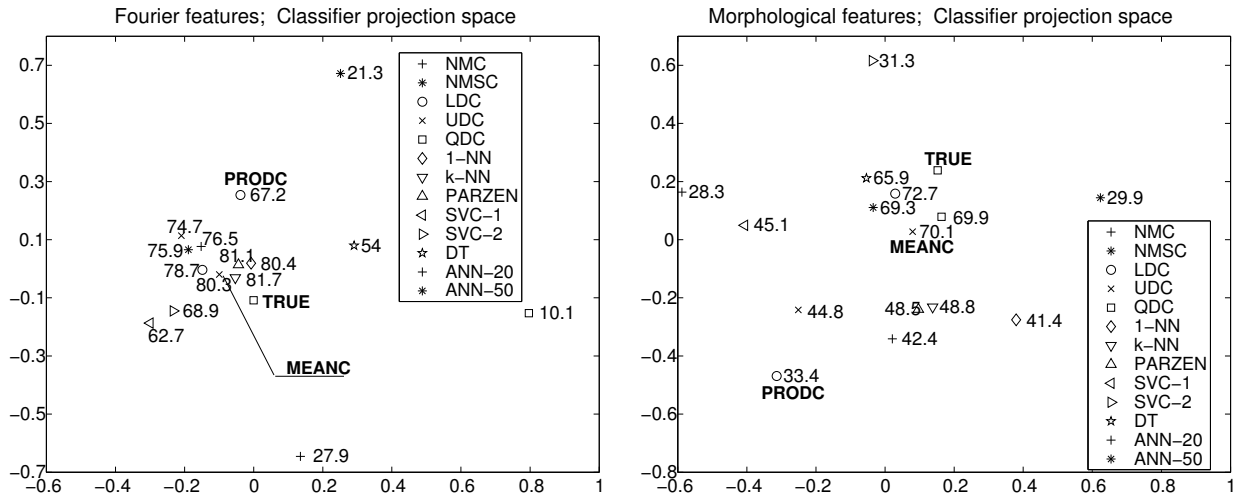


Fig. 10.5: Two-dimensional CPS for the MFEAT data: Fourier features (left) and morphological features (right). Points correspond to the classifiers; numbers refer to their accuracy. The 'perfect' classifier (true labels) is marked as TRUE. Remember that the axes cannot be interpreted themselves.

disagreement [196], which for $\mathcal{C}_i$ and $\mathcal{C}_j$ and a two-class problem is defined as (see Fig. 10.4):

$$D_{ij} := D(\mathcal{C}_i, \mathcal{C}_j) = \frac{b_{ij} + c_{ij}}{a_{ij} + b_{ij} + c_{ij} + d_{ij}}, \qquad i, j = 1, \ldots, n, \tag{10.7}$$

which is in fact the simple matching; see Table 5.1. Given the complete diversity matrix D, reflecting the relations between classifiers, the CPS is found by an approximate (non-)linear projection, a variant of Multidimensional Scaling, section 3.4.2.

Below some examples of the use of the CPS are presented.

Fixed combiners. To present a two-dimensional CPS, the ten-class MFEAT digit dataset [269] is considered; see also section A.2. For our presentation, Fourier (74D) and morphological (6D) feature sets are chosen with a training set consisting of 50 randomly chosen objects per class. The following classifiers are considered: the nearest (scaled) mean classifier, the NM(S)C, normal density based linear (the NLC), uncorrelated quadratic (the NUC) and quadratic (the NQC) classifiers, the 1-NN and k-NN rules, Parzen classifier, linear or quadratic support vector classifier, the SVC-1 or the SVC-2, decision tree, DT and feed-forward neural network with 20 or 50 hidden units, the ANN20 or the ANN50. For each feature set, the disagreement matrix between all classifiers and the two combiners, the mean (MEANC) and the product (PRODC) rules, is derived from formula

(10.7); see also Table 10.4. This is done for a test set of 150 objects per class. The diversity matrix served then for a construction of a two-dimensional CPS by the MDS procedure, described in section 3.4.2. Such examples of the CPS can be seen in Fig. 10.5. Remember that the points correspond to classifiers. The Euclidean distances between them approximate the original pairwise disagreement values. The hypothetical perfect classifier, i.e. given by the original labels, marked as TRUE, is also projected. The numbers in the plots indicate the accuracy reached on a test set.

In both cases, we can observe that the mean combiner is better than the product combiner. The latter, apparently deteriorates with respect to some, although diverse, but very badly performing classifiers. The mean rule seems to reflect the averaged variability of the most compact cloud (of classifier points). Note also that diversity might not be always correlated with accuracy. See, for instance, the right plot in Fig. 10.5, where the NMSC is more similar (less diverse) to the hypothetical classifier than ANN20, although the accuracy of the latter is higher.

Bagging, boosting and the random subspace method. Many combining techniques can be used to improve the performance of weak classifiers. Examples are bagging [42], boosting [135] or the random subspace method (RSM) [196, 367]. They modify the training set by sampling the training objects (bagging), by weighting them (boosting) or by sampling data features (the RSM). Next, they build classifiers on these modified training sets and combine them into a final decision. Bagging is useful for linear classifiers constructed when the training size is about the data dimensionality. Boosting is effective for classifiers of low-complexity built on large training sets [367]. The RSM is beneficial for small training sets of a relatively large dimensionality, or for data with redundant features (where the discrimination power is spread over many features) [367].

To study the relations within these ensembles, the 34-dimensional, two-class ionosphere data [31] is considered; see also section A.2. The NMC is used for constructing the ensembles of 50 classifiers. The training is done on the sets $\mathcal{T}_1$ and $\mathcal{T}_2$ consisting of randomly chosen $N_1 = 100$ and $N_2 = 17$ objects per class, respectively. This is done to observe a different behavior of base classifiers. The following combining rules are used: majority voting (maj), weighted majority voting (wmaj), mean, product (prod), minimum (min), maximum (max), decision templates (dtempl) and naive Bayes (NB). The test set consists of 151 objects. For each of the mentioned ensemble, the disagreement matrix between the base classifiers and the combiners is derived, which serves further for obtaining the CPS (by the classical scaling). The hypothetical, perfect classifier, representing true labels (marked as TRUE) has been added, as well; see Fig. 10.6.

To understand better the relation between the diversity and accuracy of the classifiers, while maintaining the clarity of presentation, another plots have been made; see Fig. 10.7. They show a one-dimensional CPS (representing the relative difference in diversity) versus classifier accuracy. So, the differences between classifiers in the horizontal and vertical directions correspond to the change in diversity and accuracy, respectively.

The following conclusions can be made from the analysis of Fig. 10.6 and 10.7. First of all, in the CPS, the classifiers obtained by bagging and the RSM are grouped around the single (original) NMC, creating mostly a compact cloud. The variability relations between the bagged and RSM classifiers might be very small. On the contrary, the boosted classifiers do not form a single cloud. In terms of both diversity and accuracy, from the set of 50 classifiers, they are reduced to $9 - 14$ different ones (depending on the training set). A group of $5 - 8$ poor classifiers is then separated from the others, as well as from the bagged and RSM classifiers. Secondly, for a small training set $\mathcal{T}_2$, Fig. 10.7, right, the RSM and bagging create classifiers that behave similarly in variability, since the classifier clouds in the one-dimensional CPS are of the same spread. For a larger training set $\mathcal{T}_1$, Fig. 10.7, left, the diversity for the RSM classifiers is larger than for the bagging case. Thirdly, the classifiers in all ensembles, even in boosting, seem to be constructed in a random order with respect

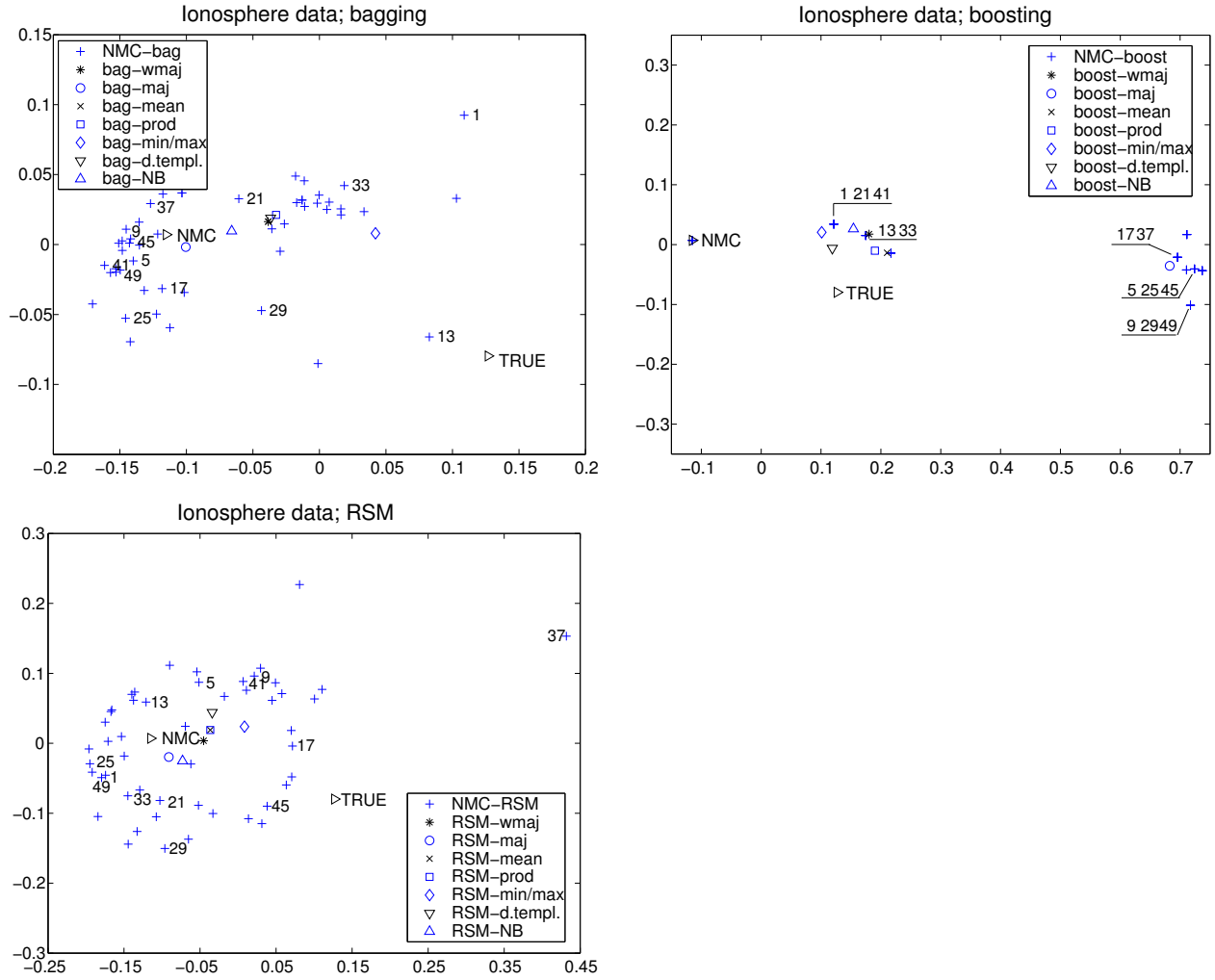


Fig. 10.6: Two-dimensional CPS for the ionosphere data trained with $\mathcal{T}_1$. The numbers correspond to the order in which classifiers are created. To maintain the clarity of presentation, only some classifiers are marked. Note the scale differences.

to the diversity and accuracy.

Concerning the combiners studied here, the minimum rule (equivalent to the maximum rule for a two-class problem) achieves, in most cases, the highest accuracy. It is even better than the weighted majority, used for the boosting construction. For a small sample size problem, Fig. 10.7, right plot, most of the combining rules for bagging and the RSM are alike, both in diversity and accuracy. A much larger variability is observed for boosting; a collection of diverse both classifiers and combiners is here obtained. Finally, a striking observation is that nearly all classifiers, as well as their combiners, are placed in the CPS at one side (i.e. not around) of the perfect classifier (this was less apparent for the MFEAT data; compare to Fig. 10.5).

Image categorization problem. In the problem of image database retrieval, images can be represented by single feature vectors or by clouds of points. Usually, given a query image Q, represented by a vector, images in the database are ranked according to their similarity to Q. This similarity is measured e.g. by the normalized inner product. A cloud of points offers a more flexible representation, but it may suffer from the overlap between cloud representations, even for very distinct images. Recently, a novel approach has been investigated for describing clouds of points based on the support vector data description (SVDD) [390], which is a boundary descriptor (an OCC) in a feature space describing the domain of such a cloud. For each image I_j in the database, represented

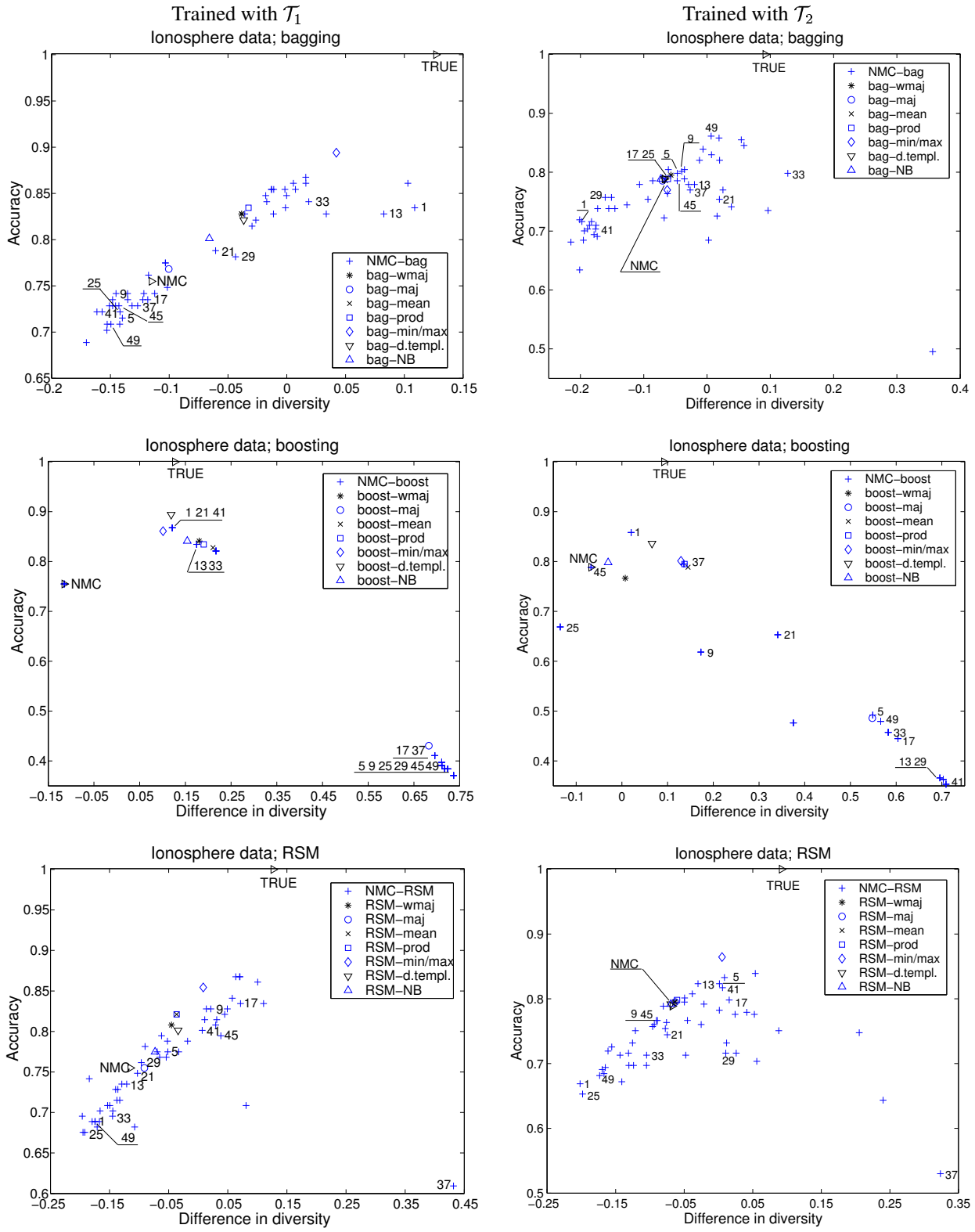


Fig. 10.7: Accuracy versus one-dimensional CPS for the ionosphere data. The numbers correspond to the order in which classifiers are created. To maintain the clarity of presentation, only some classifiers are marked. Note the scale differences.

as a set of points in a feature space, an SVDD $\mathcal{C}^j_{\text{SVDD}}$ is trained. The query image Q is represented as a set of points in the same space. The fraction $r_j(Q)$ of the query points rejected by the $\mathcal{C}^j_{\text{SVDD}}$ is a measure of a dissimilarity between the query image and the descriptor of the image I_j. A low value is expected to indicate that I_j is similar to Q. The retrieval is based on computing the fractions

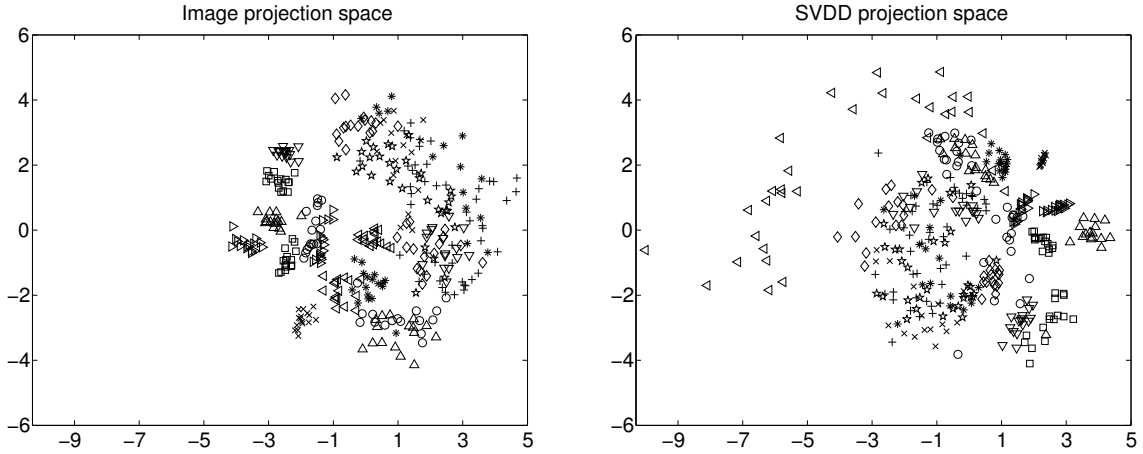


Fig. 10.8: Spatial representations: image projection space of D_I (left) and the the SVDD classifier projection space of D_{occ} (right). See text for details. Different marks refer to different classes.

$r_j(Q)$ for all images in the database, ranking them and returning the images corresponding to the lowest ranks. We have found out that a single SVDD may suffer if the clouds of points between different images are highly overlapping (this happens if the features derived from images are not well discriminating the classes). However, combining of the SVDD descriptors improves the retrieval precision; see [236] for details.

In our experiment, performed on a database of texture images, 23 different images are given. Each original image is cut into 16 128×128 non-overlapping pieces; see also appendix A.2. These correspond to a single class. Such pieces are mostly homogeneous and represent one type of a texture. The images are, one by one, considered as queries, and the 16 best ranked images are taken into account. The retrieval precision is computed using all $N = 368$ images. The details are in [236].

Each image is represented as a combined profile. This means that the image I_j is represented as $p(I_j) = [r_1(I_j)\, r_2(I_j) \ldots r_N(I_j)]$, which is in fact a conceptual dissimilarity representation, such that $r_k(I_j)$ expresses a dissimilarity between an image (a set of points) and a model (a boundary description of an image). In the standard approach, to retrieve images the most similar to Q, one will find the smallest $r_k(Q)$. In our approach, a dissimilarity between the profile of the query Q, $p(Q) = [r_1(Q)\, r_2(Q) \ldots r_N(Q)]$ and the profiles $p(I_j)$ of other images is considered. For instance, a Euclidean distance can be chosen. This approach is novel as it combines the image profiles of image one-class classifiers into a conceptual dissimilarity representation. The retrieval is then based on ranking the distances $d(p(Q), p(I_j))$ and finding the images of the smallest ranks.

In order to see all relations between the images, a distance matrix D_I consisting of the Euclidean distances between the image profiles $d(p(I_i), p(I_j))$ can be computed. The resulting spatial representation of D_I becomes then an image projection space; see Fig. 10.8, left plot. On the other hand, we can build the CPS, now based on the differences between the SVDD classifiers. To do that, one need a SVDD-profile, which for the $\mathcal{C}^j_{\text{SVDD}}$ - the boundary description of image I_j is given as $p_{occ}(\mathcal{C}^j_{\text{SVDD}}) = [r_j(I_1)\, r_j(I_2), \ldots, r_j(I_N)]$. Then, for instance a Euclidean distance matrix D_{occ} consisting of the distances between the SVDD profiles $d(p_{occ}(\mathcal{C}^i_{\text{SVDD}}), p_{occ}(\mathcal{C}^j_{\text{SVDD}}))$ can be found. A spatial representation of D_{occ} is a (one-class) classifier projection space. See Fig. 10.8, right plot. Remember that in this case, the classifiers correspond directly to the images. Comparing the two graphs, we see that the image space maintains in this case a better separation, which was confirmed by our good retrieval precision [236].

A more profound study on combining image representations for image classification and retrieval can be found in [235].

10.4 Summary

In this chapter, two different combining approaches to learning from dissimilarity representations have been investigated for the purpose of novelty detection problems and classification problems. When dissimilarity representations differ in character, combining either individual classifiers constructed on each single of them separately or by creating a new representation can be beneficial. In our experiments, we have shown that when distinct representations are combined into one representation, as a result, a representation possessing a better discriminative power can be obtained. This does not only improve the classifier, but it is also of interest because of the computational aspect. Fixed combiners, such as majority voting, can also be advantageous, especially in the case of one-class classifiers.

Additionally, a new way of representing classifiers is proposed. The classifier projection space (CPS), based on (approximate) embedding of the diversities between the classifiers, offers a possibility to study the classifier differences. This may increase the understanding of the recognition problem at hand and, thereby, offers an analyst a tool based on which she can decide on the architecture of an entire combining system. The notion of the CPS extends further to a spatial representation of conceptual dissimilarities (dissimilarities between classifiers or objects and models), which can be useful for understanding of an image retrieval problem, for instance. Conceptual dissimilarity representations resulting from combining one-class classifiers or weak models can be useful for retrieval [235, 236].

11. Conclusions

We shall not cease from exploration
And the end of all our exploring
Will be to arrive where we started
And know the place for the first time.

"FOUR QUARTETS: LITTLE GIDDING", T.S. ELIOT

The notion of proximity is fundamental in learning from a set of examples. Depending on the function it serves, a relative proximity or a conceptual proximity can be distinguished. The former describes a relation between pairs of objects, while the latter relates objects (or concepts) to a concept, such as a Gaussian model of a class. Objects are often bound together by relative proximity (quantifying their degree of commonality) to form a class. This is the necessary condition on which the compactness hypothesis relies, justifying the use of a learning algorithm. In a learning phase, a concept of a class is modeled. Any decision concerning the assignment of an object to a class is grounded in the conceptual proximity. This is the basic principle in pattern recognition.

Pattern analysis usually starts from measurements describing a set of objects. Such measurements are further preprocessed to derive a suitable description. This is a representation that can be built based on two distinctive principles: statistical or structural. Both make use of some kind of basic characteristics. In the statistical framework, these are the features, i.e. object attributes encoded as numerical variables. They are assumed to be discriminative for the object classes. A set of features constitutes a feature vector space, where each object is represented as a point. Additional structures such as inner product, norm and Euclidean distance are usually imposed to enrich this vector space. Learning is then inherently connected to the mathematical methods that can be used in this space. Although any flexible discrimination function can be designed, it will at most discover what can be inferred from the statistics of a set of features. The structural organization that an object possesses, such as connectivity of shape elements, is not incorporated in the representation.

In the structural approach, the basic descriptors are primitives, i.e. structural elements, such as strokes, corners or stems of words, encoded as syntactic units for the construction of objects. This approach is advisable for problems with objects which contain an inherent, identifiable, structure or organization e.g. shapes, spectra, images or texts[1]. There is some underlying factor in the objects, such as order, time, hierarchy or functional relationships (as between the words in sentences) that describes the inter-relationships between the morphological primitives. In the structural approach, it is assumed that there exists sufficient and suitably formulated problem knowledge, often developed and encoded with the assistance of an expert, such that a structural description of objects and classes can be constructed. Learning then relies on defining syntactic grammars or a way of comparing objects, usually in a matching process. In principle, specific criteria are used for that purpose, so the whole process is domain-specific.

In summary, the strength of the structural approach lies in encoding domain knowledge and relationships within an object, capturing its internal structural organization. The strength of the statistical approach lies in a well-developed mathematical theory of vector spaces. These approaches are complementary, hence their integration should compensate for their drawbacks, while conserving their

[1] Currently, the majority of learning tasks is concerned with this type of data. So, there is a need for designing good learning strategies, possibly incorporating both statistical and structural approaches, as they are complementary.

advantages. Some attempts in this direction have been made. For example one can associate the statistical information with structural elements to resolve some ambiguities [137]. Other possibilities include the construction of classifiers in both frameworks and combining their decisions. Such strategies are, however, hybrid. Looking at the properties of both frameworks, the unification should be reached at the representation level. In a chain of events, first a description based on structural information is derived, which is then encoded to obtain a numerical representation, which can be used in statistical learning. A natural candidate is a *proximity representation*, developed by us. This is a relative representation, in which each object is described by a set of proximities to so-called representation objects. A *conceptual proximity representation* can also be constructed which measures proximity of objects to classes or the decision boundaries induced by classifiers.

Proximity representations bridge the gap between the statistical and structural approaches to pattern recognition. This is the central motivation for this work. To limit the scope of the study, proximity is modeled as a dissimilarity, to focus on the class and object differences. This is not an essential restriction. Since similarity and dissimilarity are intimately connected, many issues discussed here can be applied to similarities after suitable adaptations.

The main goal of this thesis is to provide some foundation and to develop (statistical) learning methodologies for dissimilarity representations. The reason for the statistical framework is the necessity of establishing a learning framework for a further development of structure-aware dissimilarity measures.

The proposed dissimilarity representation is a dissimilarity matrix $D(T, R)$, where R is a set of representation objects, also called prototypes, and T is a set of training objects. The dissimilarity measure does not need to be a metric, but not any measure is acceptable. It should be meaningful to the problem and fulfill at least the compactness hypothesis, stating that similar objects are close in their representations.

Contributions

To develop learning methodologies, appropriate frameworks for the interpretation of dissimilarity representations have to be considered. Since dissimilarities quantitatively express the relative differences between pairs of objects, while learning algorithms usually optimize a kind of an error in the context of a chosen numerical model, one will deal with numerical representations of the problems. The numbers have, therefore, a particular meaning within the frame of specified assumptions and models. Spaces with different characteristics lead to different interpretations of the dissimilarity data, hence to different learning algorithms.

Chapter 2 briefly introduces topological, (indefinite) inner product, norm and metric spaces. Although most of the material presented there is not new, Kreĭn spaces are not usually treated in the standard works. Our major contribution is to present the relations between the spaces and the development of the Kreĭn space, later discussed in the form of a pseudo-Euclidean space of a finite embedding. The introduction of these spaces prepares the way for a mathematical framework for handling arbitrary dissimilarity data.

Metric dissimilarities have advantageous properties, since many numerical methods operate in metric spaces, or more specifically in Euclidean spaces. In chapter 3, dissimilarities are further characterized with respect to Euclidean and metric properties. Further on, a linear pseudo-Euclidean embedding is studied, as well as nonlinear multidimensional scaling. This prepares the ground for one of the learning approaches defined in chapter 4.

Three main frameworks have been proposed for learning on dissimilarity representations, which rely on the following interpretation of dissimilarities:

1. as relations between the objects based on dissimilarity-ball neighborhoods,
2. in an embedded space, where the original dissimilarities are preserved, found by a linear pseudo-Euclidean embedding,
3. in a dissimilarity space, where each dimension is a dissimilarity to a particular object.

These three approaches are discussed in chapter 4, where the learning strategies are introduced. A natural question that arises now is how these strategies differ from the standard learning techniques in feature spaces. If one relies on (Euclidean) distances in a feature-based representation, the methods applied on such distances refer to a topological space. The difference lies in the accompanying feature space and the metric distances. The use of embedded and dissimilarity spaces is novel. However, it might be seen as a generalization framework of the support vector machines (SVM). An SVM can be seen as a linear classifier in some high-dimensional space defined by the (conditionally) positive definite kernel. In our approach, a linear classifier in the dissimilarity space can be interpreted as a quadratic (or linear) classifier in a high-dimensional Kreĭn space. Since one deals with finite samples, such a Kreĭn space simplifies to a finite-dimensional pseudo-Euclidean embedded space. Basically, the SVM is a mathematically elegant, but specific procedure in our framework.

Basic dissimilarity measures and a brief overview of measures used in practical applications have been discussed in chapter 5.

Chapters 6 – 10 constitute the experimental part of this thesis, in which dissimilarity representations are practically analyzed. A systematic approach is presented to such an analysis, hence the most basic questions concerning the data understanding are handled first.

Chapter 6 investigates a number of well-known visualization techniques and their usefulness for dissimilarity data. The conclusion is that multidimensional scaling techniques and Isomap provide useful insights into the relations in the data.

Chapter 7 focuses further on methods that help in data exploration. Three main issues are investigated concerning both structure and complexity in the dissimilarity representation: clustering techniques, intrinsic dimensionality and sampling. A number of clustering methods in the three interpretation frameworks is presented. Preliminary results of the clustering in dissimilarity spaces are promising. Additionally, a statistical estimate of the intrinsic dimensionality from a Euclidean distance representation of a hyper-spherical Gaussian sample is derived. Finally, some criteria are proposed and examined that can be used in quantifying whether a representation set contains a sufficient number of objects to describe a class. The most useful criteria are the ones based on the number of most significant eigenvalues in either in PCA-dissimilarity space or pseudo-Euclidean embedding and the mean relative rank criterion. A more detailed study on the sampling issues has been performed, where additionally the skewness criterion is found indicative [104].

Chapter 8 moves on to the construction of one-class classifiers (OCCs) on dissimilarity representations. Currently existing OCCs are built either on features in traditional feature spaces or on Euclidean distances derived there. Two new OCCs, one in embedded space and one in dissimilarity space, are proposed and successfully applied to some practical problems. Non-metric dissimilarity measures seem to work well for noisy data in such domain description problems. Usually, only metric measures are used.

Chapter 9 is concerned with classification issues. Dissimilarity measures with different properties (Euclidean, non-Euclidean metric and non-metric) are analyzed for this purpose. Experiments demonstrate that simple linear or quadratic classifiers constructed in dissimilarity or embedded spaces may significantly outperform the k-NN rule for smaller representation sets, irrespective of whether the dissimilarity is metric or not. We also investigated some ways, as discussed in chapter 3,

of transforming the dissimilarity measure to make it (more) Euclidean (hence more metric) for the purpose of discrimination. We have found that the imposed Euclidean behavior cannot guarantee a better performance. It is more important that the measure itself describes compact classes than its strict Euclidean or metric properties.

Various prototype selection criteria are proposed and studied for both embedded and dissimilarity spaces, indicating that systematic procedures (making use of the label information) are beneficial, especially for a smaller number of prototypes. For very small representation sets a supervised selection based on the cross-validation error of a classifier or a forward feature selection method also based on the classification error are the best. In general, for all representation set sizes, the k-centres clustering finds good prototypes, especially for multi-modal data. In dissimilarity spaces, the representation set selected by a sparse linear programming gives a good discrimination. The drawback is the lack of control over the number of selected prototypes. That is why the k-centres selection, followed by the sparse LP may offer a better result. In embedded spaces, except for the k-centres procedure, alternatively, the prototypes selected as the ones which yield the average approximation error can be chosen. Additionally, we have observed that for representation sets consisting of more than 20% of the training objects, a random selection is beneficial. Some considerations on the framework of a zero-error recognition have also been shared.

Combining information originating from different sources or combining individual learning strategies can be effective for designing a good pattern recognition system. Some of these issues are discussed in chapter 10. Combining is a natural way of integrating the statistical and structural representations into one framework. Some ways are proposed of combining dissimilarity representations into a new one on which a single final classifier can be trained. In our experiments on two-class and one-class classification problems, we found that dissimilarity representations combined by either a (weighted) average or product have a larger discriminative power than any single one. Classifiers built on such combined representations outperformed the best classifier (of the same type) constructed on single representations. This is especially useful if the final classifier works in a reduced dissimilarity space, as offered by the linear programming data description (LPDD) for one-class classification tasks.

Additionally, we have observed that classifiers, first trained on single representations and then combined, work well. Especially, the product rule combiner seems to be good for small representation sets in two-class classification problems, while majority voting may be advantageous for one-class classifiers.

In brief, this thesis develops a general framework for learning from dissimilarity representations.

Open questions

This thesis can serve as a foundation for continuing research into learning from dissimilarity representations. The aim is to renew the pattern recognition area by the integration of structural and statistical approaches. At the fundamental level, the topics of interest are described below.

1. We think that neighborhood-based pretopological spaces are important for a further development of pattern recognition. They allow one to use weaker type of relations between objects (without additional structures of an inner product or a norm), hence novel types of relational classifiers could be potentially constructed. These should be domain-based, in contrast to probability-based, decision functions. Although they might not be able (at this time) to compete with the advanced techniques of inner product spaces, they might stimulate new ways of thinking.
2. It seems that metric or Euclidean properties of a dissimilarity measure are less important than their discriminative properties; see sections 9.2.3 and 9.4.1. Although some intuition has

been developed and non-metric and non-Euclidean behavior is characterized, it is important to study these properties fundamentally in relation to the topological, embedded and dissimilarity spaces. New types of measures could be developed, especially in the structural approach, and applied in a dissimilarity-based framework, without imposing metric constraints.

3. An understanding is needed of the topological relations between the three spaces: pretopological, embedded and dissimilarity spaces. Non-decreasing nonlinear transformations of the dissimilarity measure change the topological properties of the embedded and dissimilarity spaces, while they do not affect the dissimilarity-ball neighborhoods. Our results suggest that these concave transformations, like sigmoidal ones, can be beneficial for discrimination, since they diminish the effect of possible outliers (see section 9.4). A more thorough investigation is needed.
4. The possibility of zero-error dissimilarity-based classifiers has been introduced. Ultimately, it is related to the compactness hypothesis and a true representation, section 4.1.1. They both put constraints on a dissimilarity measure which should be such that not only similar objects similar are close in their representations, but also the other way around. This issue has to be studied more theoretically. For instance, for shapes in images, this would include a study on robustness of a measure against object position and orientation, small perturbations and occlusions.
5. The design of morphological (structure-aware) dissimilarity measures, both general and specific for the problem at hand, is an open issue. This would require the definition of a suite of structure detectors, general enough for the data types such as images, time-signals, spectra etc. The intriguing question is not only how data type specific detectors should be found, but more importantly, how a measure can be learned from a given set of examples. Some inspiration can be found in [153, 160–162].
6. In general, some foundation for learning from dissimilarities has been laid down, but much more should be done. Research effort should be devoted to the further development of the proposed framework, aiming at integration of both statistical and structural approaches.

On the methodological level, topics of investigations include the following issues.

1. The use of dissimilarity neighborhoods is very popular in clustering, so many algorithms have been developed so far. Preliminary results on the use of embedded and dissimilarity spaces give promising results. Some theoretically well-founded methods can be developed.
2. Given a training set, a number of methods for the selection of a representation set appropriate for learning in dissimilarity and embedded spaces have been proposed (see sections 9.2 and 9.3). The methods should be studied further in a number of applications. The next step relies on designing new prototypes at the level of measurements. This means that new prototypes encompassing the information on a number of original objects are created and used for learning. This would mean that e.g. the information on a set of spectra, where each spectrum describes a particular case, could be captured by their most representative spectrum, which becomes a member of the representation set. One could expect that if domain-based ways are used to derive new prototypes, the resulting representation set can be powerful.
3. We mostly used linear and quadratic classifiers in the embedded and dissimilarity spaces. They may suffer from the curse of dimensionality [208] if large representation sets create spaces of a high dimensionality. The use of decision trees, appropriately reformulated for dissimilarities, might be an alternative in this case. This is open for investigation.
4. In the area of combining, some a priori knowledge, e.g. on labels, could be incorporated in the combined dissimilarity representation and in the final classifier. Inspiration can be found in the work of Muñoz and Martin de Diego [79, 277]. It can also be advantageous to combine

representations derived by employing both statistical and structural approaches.

Another intriguing point of interest is to combine the three learning frameworks, to benefit from the strength of each of the interpretation spaces. The k-NN rule is locally sensitive, while a linear (or nonlinear) classifier in the embedded or dissimilarity spaces is globally sensitive, as it relies on all representation objects. How to combine such information is a point for research.

5. This thesis is mostly concerned with inductive learning principles. The next step for an investigation is transductive learning [403], which might be considered together with the issue of combining the local and global approaches to the dissimilarities. Additionally, new research areas are open for study: learning from unlabeled data (partly related to the clustering issue) and active learning.
6. New applications, especially from structural pattern recognition, should be considered.

In conclusion, the use of proximity representations opens a new possibility for integrating both statistical and structural approaches to learning from a set of examples.

Practical considerations

In all experiments performed on various dissimilarity data sets, the conclusion is that both dissimilarity and embedded spaces defined on dissimilarity representations $D(T, R)$ offer a good compromise between learning accuracy (precision) and computational effort, often better than the nearest-neighbor methods directly applied to the dissimilarities. The best approach is problem-dependent.

Practical advice and useful suggestions can be formulated. First of all, one needs to understand the problem and the data. One may start from the observations of the distribution of the dissimilarities in the form of a histogram and by deriving simple statistics as the mean, standard deviation, modes, kurtosis, skewness etc. Visualization techniques, as described in chapter 6 should be used to get further insight into the dissimilarity data. Low-dimensional spatial maps of dissimilarities offer a way to inspect the relations between the objects. Classical scaling results and PCA-dissimilarity space should be studied first. Later on, Sammon mapping S_0 and Isomap can be used. To analyze the hierarchical organization of the objects, an ultrametric dissimilarity tree can be constructed by single-linkage clustering or a minimum spanning tree. If the examples are labeled, the study of the intensity image of the dissimilarity relations is recommended to analyze the discrimination properties between classes, existence of outliers or additional clusters.

Before moving to task-specific suggestions, first some general recommendations are given:

1. If dissimilarity values are very large, e.g. the average dissimilarity is larger than 100, use a linear scaling to bound them to a reasonably small interval such as $[0, 1]$ or $[0, 10]$. This is necessary to avoid numerical problems.
2. If objects contain an identifiable, structure or organization, make use of a structural approach to derive the dissimilarity representation.
3. If the classes (or expected clusters) have different spreads (e.g. one is compact, while the other is widely spread), start your analysis from $D^{*2}(T, T)$ instead of $D(T, T)$. The square dissimilarities emphasize the class differences even more.
4. If the dissimilarities take only a limited number of different values (the measure is not continuous), the embedding approach is preferred.
5. If different sources are available (e.g. by using different excitation wavelengths to measure various sets of respond spectra), make use of them. If a number of dissimilarity measures different in characteristics can be defined, make use of them as well. Different representations can be defined and combined later on.

Clustering. Remember that the clustering task is a subjective one, since the data can be partitioned differently depending on what is taken into account. Moreover, one cannot discover structures, which are not encoded in the dissimilarities. For instance, if the dissimilarity measure is poor, compact clusters will not be found.

Hypothesize K clusters in the dissimilarity data $D(T,T)$. To label the objects, do the following:

1. Analyze a dendogram on the dissimilarity data obtained by a hierarchical clustering method to detect specific clusters. Cut the dendogram at a specified level to determine K clusters.
2. Analyze the k-centers result.
3. Determine the spaces:
   (a) Embedded space. Find the dimensionality of the embedded space $\mathcal{E}$ based on the number of most significant eigenvalues in the pseudo-Euclidean embedding of $D(T,T)$. If T is large, select a subset R of T by the k-centres algorithm, where e.g. k equals 20% of the data instances, and use $D(R,R)$ to define the embedding. Project the remaining examples to $\mathcal{E}$ and use all of them.
   (b) Dissimilarity space. Select the dimensionality of the PCA-dissimilarity space by the amount of preserved variance. If T is large, make use of $D(T,R)$, where R is a representation set chosen by the k-centres method.

   Apply NLC-clustering or NQC-clustering in the embedded and dissimilarity spaces to find K clusters. Repeat these M times with different initializations. Use the goodness-of-clustering measure J_{GOC} to decide which partitioning is the best.
4. Inspect various clustering results by visual judgment:
   (a) Low-dimensional spatial maps obtained by classical scaling or the Sammon map.
   (b) Intensity images of $D(T,T)$ as suggested in section 7.1.2.

Specify a criterion regarding cluster separability and cluster compactness, e.g. by formula (7.1), to judge the clustering results and select the best one. If the number of clusters is unknown, some criterion should be defined to evaluate the results for a various number of clusters. In probabilistic approaches to clustering, likelihood-ratio measures can be used. For hierarchical approaches, a criterion suggested in [132] can be exploited.

One-class classification. Suggestions for approaching one-class classification problems:

1. Consider an adequate non-metric dissimilarity as a representation of the problem, if the original measurements, such as images or spectra, are noisy.
2. Analyze the distribution of all dissimilarities in the target class. If there exists a long tail of large dissimilarities (the skewness is highly positive), apply a concave and bounding transformation, such as $f_p(d) = d^p$, $p < 1$, or $f_{\text{sigm}}(d) = 2/(1 + e^{-d/s}) - 1$, to all the dissimilarities. Choose the transformation parameters p or s based on a validation set or by some stability criterion for an OCC [388]. Do not apply such transformations to neighborhood-based OCCs, since they are not useful.
3. If the target set is small or the computational aspect is not important, use a neighborhood-based descriptor such as the k-nearest neighbor data description. If outlier examples are available, use a linear programming data description (LPDD or LPDD-II) in a dissimilarity space. Alternatively, if the distance representation is (nearly) Euclidean (all eigenvalues in the linear embedding are nonnegative), consider a support vector data description on the positive definite Gaussian kernel $K = e^{-D^{*2}/\sigma^2}$ [386, 390].
4. Use a weak classifier, e.g. a generalized nearest mean data description for poor dissimilarity representations.

5. To emphasize a number of properties of the initial representation (such as raw measurements), define a few suitable *different* dissimilarity measures and derive representations. Combine these representations by a weighted average and train a single LPDD or, alternatively, use the majority voting rule to combine a number of LPDD outputs trained on single representations.

Classification. Before any classification experiment, get insight into the data to learn about possible outliers, modality of the classes and their spread (e.g. find their average within-class and between-class dissimilarities). Determine outliers either by using OCC methods or by detecting objects with very large dissimilarities to other objects. Removing outliers is more important for the embedding approach than for the dissimilarity space approach. General suggestions are:

- If the chosen dissimilarity measure is based on sums (built from a number of components of similar variances), use normal density-based linear or quadratic classifiers.
- Estimate m, the number of significant eigenvalues in the linear embedding. If m is large and the 'eigenvalue curve' does not approach zero reasonably fast, focus on the k-NN method and the dissimilarity space approach. Otherwise, consider all three frameworks.
- If a number of different dissimilarity measures, emphasizing different characteristics in the initial data (images, spectra, graphs, etc) can be designed for the problem, derive the representations, scale them appropriately and combine them by a weighted average.

Assume a single (given, optimized or combined) dissimilarity representation $D(L, L)$, where L is a learning set. To find the best pattern recognition approach, perform M times, for instance $M = 50$, a 90%-10% hold-out experiment[2], i.e. split randomly all objects from L into the training set T and the test set T_{te} such that T consist of 90% of the examples and the remaining 10% are assigned to T_{te}.

Consider the dissimilarity space and neighborhood-based approaches. In each step:

1. Determine the following representation sets as subsets of T:
   - R_{LP} by applying sparse linear programming (LP) to $D(T, T)$. Use formulation (4.15) with $\gamma = 1$ or, alternatively, formulation (4.16) with μ being a (rough) estimate of the generalization error (e.g. estimated as the 1-NN error). The latter formulation should be more useful, when the classification problem is difficult.
   - R_{EC} by using a 1-NN editing-condensing algorithm [86] on $D(T, T)$.
   - R_{k-LP} by using the k-centres procedure to preselect a set R_* of K objects, e.g. consisting of 20% of the training examples, and then applying the sparse LP formulation (4.15) to $D(T, R_*)$.
2. Train NLC, NQC (use regularized versions if needed) and the standard non-sparse LP machine (4.14) in the dissimilarity spaces $D(T, R_{LP})$, $D(T, R_{EC})$ and $D(T, R_{k-LP})$. Find the test classification errors.
3. Make use of the same representations sets as above to compute the 1-NN error on the test set, as well as the 1-NN and the k-NN error using all training objects (optimize k on the training set by the leave-one-out procedure).
4. If T is of small or moderate size or the computational aspect is not important, train the NLC, the NQC and the standard non-sparse LP machine (4.14) in the PCA-dissimilarity space (perform the PCA on $D(T, T)$ and select the dimensionality corresponding e.g. to 95% of the preserved variance). Compute the test errors.
5. Additionally, if $D(T, T)$ is (nearly) Euclidean, consider a support vector machine [74, 352] on the positive definite Gaussian kernel $K = e^{-D^{*2}/\sigma^2}$.

In the embedding space approach, in each step:

[2] Alternatively, consider a 5- or 10-fold crossvalidation experiment repeated e.g. 20 times.

1. Determine the dimensionality m of the approximate linear embedding of $D(T,T)$ (as the number of significant eigenvalues).
2. Find the following representation sets of $m+1$ objects:
   - R_* by applying the k-centers algorithm to $D(T,T)$.
   - R_{APE} by selecting prototypes which yield the smallest average approximation error.
   - R_P by selecting pivot object as in the FastMap technique.

   Use the same procedures as above to select $2m+1$ objects. This leads to six representation sets in total, three sets for $m+1$ objects and three sets for $2m+1$ objects.
3. For each selected representation set R above, use $D(R,R)$ to find the m-dimensional embedded space $\mathcal{E}$. Project the remaining objects $T \backslash R$ to this space and train a linear or quadratic classifier there. Project T_{te} to $\mathcal{E}$ and test the classifiers.
4. Determine also the embedding in m-dimensional space based on the complete data $D(T,T)$. Apply the same discrimination functions as above.

Perform the same experiment as above on a sigmoidal transformation of the dissimilarities $f_{\text{sigm}}(D^{*2}(T,T))$. Average the classification errors of all approaches. Choose a decision rule and a representation set as a trade-off between performance and computational effort for the evaluation of new objects. Additionally, if very small representation sets (of few objects) need to be selected, then, make use of the forward feature selection method with the criterion based on the classification error.

APPENDIX

A. Data sets

All information is imperfect. We have to treat it with humility.
JACOB BRONOWSKI

To avoid multiple descriptions of the data sets used in our study, they are introduced here. They are examples of data with different characteristics. Therefore, they should be representative for a number of learning problems dealing with dissimilarity representations (DR). Some of the dissimilarity data matrices are visualized as intensity images, where each pixel corresponds to a dissimilarity value between a pair of objects. The darker the pixel, the smaller the dissimilarity, hence the black line on the diagonal describes zero values. Additionally, the eigenvalues of the pseudo-Euclidean linear embedding, discussed in section 3.3, may be presented. The usage of the data sets described in this chapter is summarized in Table A.1.

Table A.1: Data sets used in the thesis.

Data	Usage
Ringnorm	Clustering: chapter 7
Hypercube	Visualization: chapter 6
Banana	Illustration and visualization: chapters 3, 4 and 6
Polygon	Classification: chapter 9
Convex polygon	Classification: chapter 9
Ionosphere	Combining: chapter 10
Wine	Classification: chapter 9
Ecoli	Classification: chapter 9
MFEAT	Combining: chapter 10
Pump vibration	Visualization: chapter 6
Cat-cortex	Clustering: chapter 7
Protein	Clustering: chapter 7
Ball-bearing	One-class classification: chapter 8
Heart disease	One-class classification: chapter 8
Diseased mucosa	One-class classification and combining: chapters 8 and 10
Geophysical spectra	Classification: chapter 9
ProDom	Classification: chapter 9
NIST digit	Exploration and classification: chapters 7 and 9
NIST-38 digit	Classification and combining: chapters 9 and 10
Zongker digit	Visualization and classification: chapters 6 and 9
Pen-based digit	Classification: chapter 9
Newsgroups	Visualization and clustering: chapters 6 and 7
Texture	Combining: chapter 10

A.1 Artificial data sets

We will consider a number of artificial data sets describing two-class discrimination problems. Gaussian data refer to normally distributed classes. Studying artificial data are useful, since we can control their parameters or properties, such as the initial dimensionality and class overlap. Therefore, some insight can be gained while different dissimilarity measures are used for the representation.

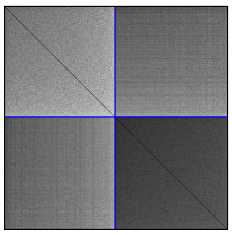


Fig. A.1: Euclidean DR for the ringnorm data.

Ringnorm. This is an implementation of Breiman's ringnorm example [43], taken from DELVE [81]. The data consist of two classes in a 20-dimensional space. Each class is drawn from a multivariate normal distribution. The first class has a zero mean $\mathbf{m}_1 = \mathbf{0}$ and the covariance matrix of $C = 4 \cdot I$. The second class has

the mean $\mathbf{m}_2 = 2/sqrt(20)\,\mathbf{1}$ and the identity covariance matrix. Breiman reports the theoretical expected misclassification rate of 1.3%. A Euclidean distance is used for the representation; see also Fig. A.1. This data set is used in section 7.1.2 for the illustration of clustering approaches.

Hypercube data. This data set consists of 600 points generated according to a uniform distribution and equally confined in two hypercubes in a 100-dimensional space. The leftmost corner of both hypercubes is set to the origin. The edge lengths of the hypercubes are 0.5 and 1, correspondingly. This means that the first hypercube contains the other one and, in fact, the sampling density in the small hypercube is larger than outside it. The Euclidean distance representation has been considered for these data, which will give an indication of a clear cluster corresponding to the points of the small hypercube. Due to the coarse sampling of the points outside this hypercube, their distances become relatively larger. Moreover, in such a space, they tend to lie close to the boundary. Note that this is the well-known effect of the curse-of-dimensionality. The volume of the small hypercube with respect to the large one is $(0.5/1)^{100} \approx 7.9 \cdot 10^{-31}$. Not surprisingly, the points in the small hypercube are close, while others are remote. In order to realize that the data points are uniform in both hypercubes, one would need, 10^{100} sampled points, for instance. This is not feasible, so for any coarse sampling, we should perceive two clusters: one compact and the other spread out. This fact can also be clearly observed while studying the corresponding dissimilarity matrix, see Fig. A.2. This data set is used in chapter 6 for visualization.

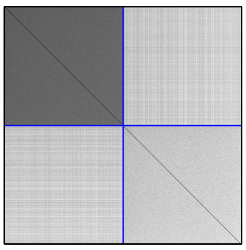


Fig. A.2: Euclidean DR for the hypercube data.

Banana data. This data set consists of two banana-shaped classes in a 2-dimensional space. It is mainly used for illustration purposes when a number of different dissimilarity measures is considered. See Fig. A.3 for an illustration.

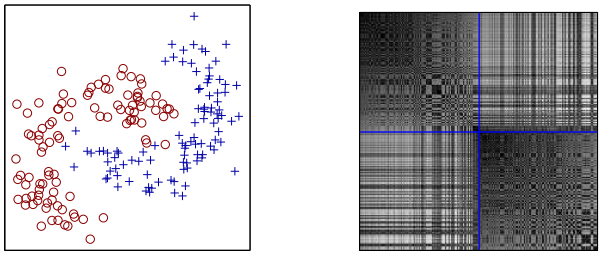


Fig. A.3: Banana data (left) and its Euclidean distance representation.

Polygon data. The data consist of two classes of polygons: convex quadrilaterals and irregular heptagons, randomly generated. See Fig. A.4 for some examples. The polygons are first scaled and then the metric Hausdorff distances, Def. 5.3, and non-metric modified Hausdorff distances, Def. 5.6, are computed between their vertices. In total, 2000 objects per class are available. The intensity plots of the derived dissimilarity representations are presented in Fig. A.5. This data set is used in chapter 9 for classification.

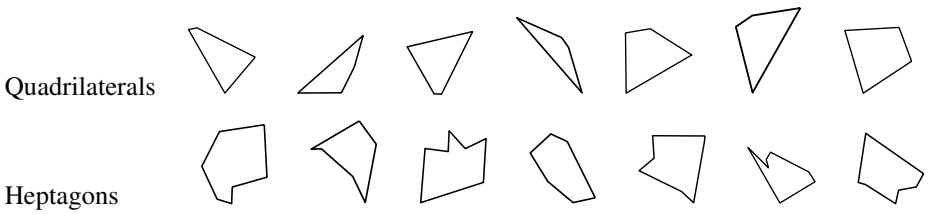


Fig. A.4: Polygon data: examples of quadrilaterals convex and irregular heptagons.

Convex polygon data. The data consist of convex pentagons and heptagons. For the generation of a polygon, p vertices (5 for pentagons and 7 for heptagons) are first regularly positioned on the unit circle such that the Euclidean distances between two consecutive vertices are equal. Next, two-dimensional noise is added to each vertex to perturb the polygons. Similarly as for the polygon data above, the Hausdorff and modified-Hausdorff distance representations are considered. Some examples are shown in Fig. A.6. The data set is used in chapter 9 for building zero-error classifiers.

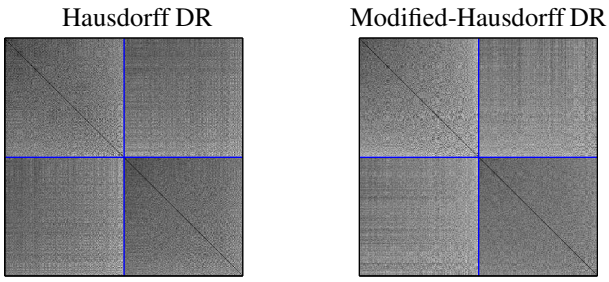


Fig. A.5: Dissimilarity representations for the polygon data.

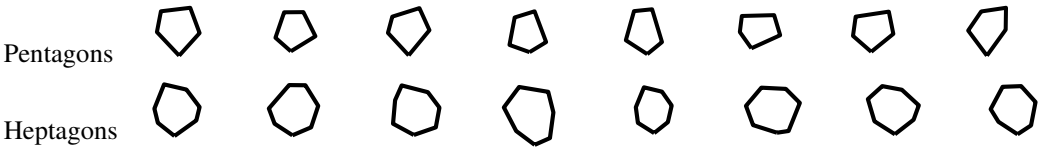


Fig. A.6: Convex polygon data: examples of pentagons and heptagons.

A.2 Real-world data sets

Our goal is to show the usefulness of dissimilarity representations for novelty detection and classification problems. To be representative, real data sets will have various characteristics. There are examples, in which raw data are collected by a sensors and represented in a digitized form by spectra, shapes, or images. There are also cases, in which the original feature-based data are of mixed types or lie in a high-dimensional space.

Ionosphere data. This radar data, coming from UCI Repository [31], was collected by a system of 16 high-frequency antennas with a total transmitted power of about 6.4 kW in Goose Bay in Labrador. The targets were free electrons in the ionosphere. Positive examples are those for which the evidence of the structure in the ionosphere was shown. Negative examples refer to the cases where nothing was returned, thus the signals went through the ionosphere. The received signals are preprocessed by using an autocorrelation function with the arguments being the time of a pulse and the pulse number. For 17 pulse numbers present, each instance in this data is described by two attributes per pulse number, corresponding to the complex values obtained from the complex electromagnetic signal. Hence, the data is described by 34 features. The positive class consists of 225 examples and the negative class posses 126 examples, yielding 351 examples, in total. This data set is used in section 10.3 for the illustration of the classifier projection space being a spatial representation of classifier diversities in an ensemble of classifiers.

Wine data. The *Wine* data come from Machine Learning Repository [31] and describe three types of wines described by 13 features. In each experiment, when the data are split into the training and test sets, the features are standardized as they have different ranges. A Euclidean distance is chosen for the representation.

Ecoli data. The data come from Machine Learning Repository [31] and describe eight protein localization sites. Since the number of examples in all these classes is not sufficient for a prototype selection study, three largest localization sites are selected as a sub-problem. These localization classes are: cytoplasm (143 examples), inner membrane without signal sequence (77 examples) and perisplasm (52 examples). Since the features are some type of scores between 0 and 1, they are not normalized. Five numerical attributes are taken into account to derive the l_1 and $l_{0.8}$ distance representations, denoted as *Ecoli-p1* and *Ecoli-p08*, respectively. Remember that the l_p distance between two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ is computed $d_p(\mathbf{x}_i, \mathbf{x}_j) = (\sum_{z=1}^{m} |x_{iz} - x_{jz}|^p)^{1/p}$ and it is metric for $p \geq 1$.

MFEAT data. This data set consists of sets of features derived for handwritten numerals '0'-'9' extracted from a collection of Dutch utility maps. 200 patterns per class have been digitized in binary images. The digits are represented by six feature sets, as used in [210]. Here two feature sets are used: *Fourier* describing 76 Fourier coefficients of the character shapes and *morphological* describing six morphological features; see [31]. This data set is used in section 10.3 for the illustration of the classifier projection space being a spatial representation of classifier diversities in an ensemble of classifiers.

Pump vibration data. Pump vibration was measured with three accelerometers mounted on a submersible pump which operated in three states: normal, presence of imbalance and presence of bearing failure. Moreover, the bearing failure was measured at three different operating speeds. The data consist of 500 observations with 256 spectral features of the acceleration spectrum (see [247]). It is known [427] that the data has a low intrinsic dimensionality and that it probably lies in a nonlinear subspace of a 256-dimensional space. The city block distance representation has been considered for this set, as it can be observed in Fig. A.7. The data are used in chapter 6 for visualization.

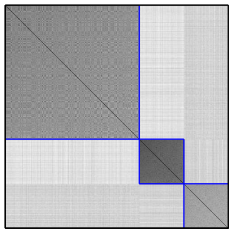


Fig. A.7: City block DR for the pump data.

Cat-cortex data. The cat-cortex data set is provided as a 65×65 dissimilarity matrix describing the connection strengths between 65 cortical areas of a cat. It was collected by Scannell [336] and used for clasification in [172, 173] and for clustering in [85]. The data set is obtained from [84]. The dissimilarity values are measured on the ordinal scale and take the following values: 1 for a strong and dense connection, 2 for an intermediate connection, 3 for a weak connection and 4 for an absent or unreported connection [172]. Concerning the cortex functions, four regions can be distinguished: auditory (A), frontolimbic (F), somatosensory (S) and visual (V). The class cardinalities are 10, 19, 18 and 18, respectively. The above mentioned classes can be identified in Fig. A.8, left. One may also observe that the classes are not homogeneous and that there is a confusion between the frontolimbic class and other classes. As indicated by the negative eigenvalues of the pseudo-Euclidean embedding, Fig. A.8, right, the dissimilarity data are highly non-Euclidean. This data set is used in section 7.1.2 for the illustration of clustering approaches.

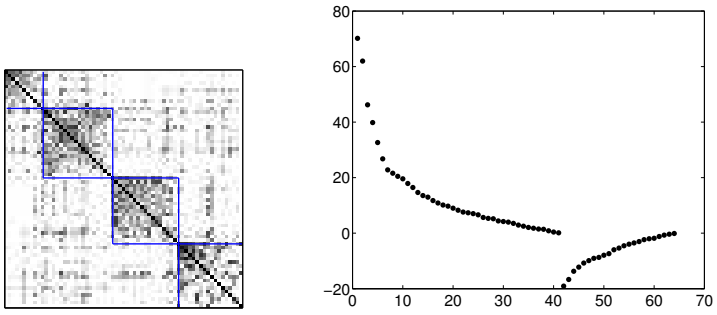


Fig. A.8: Cat cortex dissimilarity matrix (left), where the visible clusters (denoted by rectangles) are presented in the following order: A, F, S, and V and the eigenvalues in the pseudo-Euclidean embedding (right). See text for details.

Protein data. The protein data are provided as a 213×213 dissimilarity matrix comparing the protein sequences based on the concept of an evolutionary distance. It was used for clasification in [172] and for clustering in [85]. The data set is obtained from [84]. The proteins are originally assigned to four classes of globins: heterogeneous globin (G), hemoglobin-α (HA), hemoglobin-β (HB) and myoglobin (M). The class cardinalities are 30, 72, 72 and 39, respectively. The above mentioned classes can be identified in Fig. A.8 (left), however the globin class is very weak. Not surprisingly, the hemoglobin classes are similar, while the myoglobin class is distinct. One may also observe that the classes are not homogeneous and that there is a confusion between the frontolimbic class and other classes. As indicated on the right in Fig. A.9, the dissimilarity data are nearly Euclidean. This data set is used in section 7.1.2 for the clustering approaches.

Ball-bearing data. Fault detection is an important problem in machine diagnostics. A detection of four types of fault in ball-bearing cages is considered, a data set [124], as used in [53]. Each data item consists of 2048 samples of acceleration taken with a Bruel and Kjaer vibration analyzer. After preprocessing with a discrete Fast Fourier Transform, each signal is characterized by 32 attributes. There are five categories: normal behavior, NB, corresponding to measurements made from new ball-bearings and four types of anomalies A_1 –A_4: the outer race completely broken (A_1), broken cage with one loose element (A_2), damaged cage with four loose elements (A_3) and a badly worn ball-bearing with no evident damage (A_4); see Fig. A.10 for some examples. The data representation is based on Euclidean, city block and $l_{0.8}$ distances together with their power and sigmoidal transformations. This data set is used in chapter 8 for training one-class classifiers.

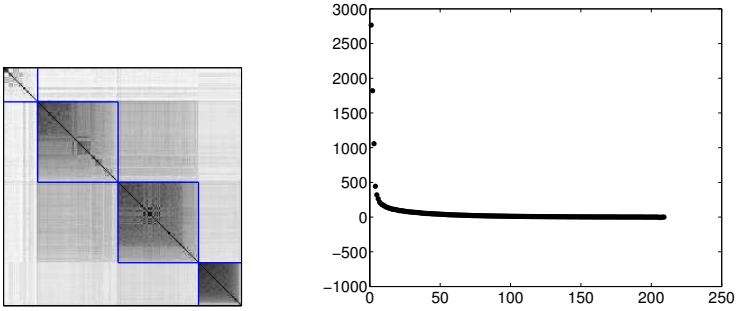


Fig. A.9: Protein dissimilarity data (left), where the visible globin clusters (denoted by rectangles) are presented in the following order: G, HA, HB, and M, and the eigenvalues in the pseudo-Euclidean embedding (right). See text for details.

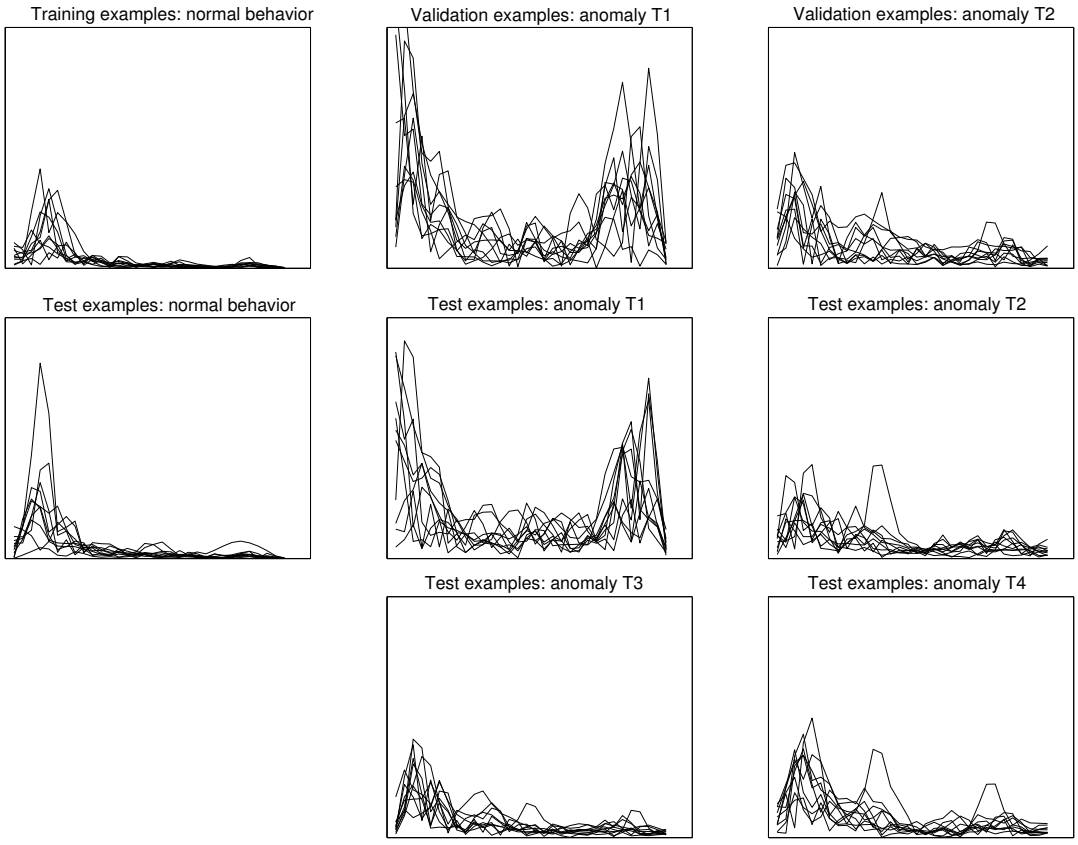


Fig. A.10: Examples of the pre-processed acceleration samples from the ball-bearing data.

Heart disease data. The data come from the UCI Machine Learning Repository [31]. The goal is to detect the presence of heart disease in the patient. There are 303 cases, where 139 correspond to ill patients. This database contains 75 attributes, but all published experiments refer to using a subset of 13 of them, so we use them as well. The attributes are: age, sex $(1/0)$, chest pain type $(1-4)$, resting blood pressure, serum cholesterol, fasting blood sugar >120 mg/dl $(1/0)$, resting electrocardiograph results, maximum heart rate achieved, exercise induced angina $(1/0)$, the slope of the peak exercise ST segment, ST depression induced by exercise relative to rest $(1-3)$, number of major vessels colored by fluoroscopy $(0-3)$ and heart condition (3 - normal, 6 - fixed defect, 7 - reversable defect). Hence, the data consist of mixed types: continuous, dichotomous and categorical variables. There are also several missing values. Gower's dissimilarity, as defined in (5.5), has been chosen for the representation. See also Fig. A.11. This data set is used in section 8.3.3 in the one-class classification problem.

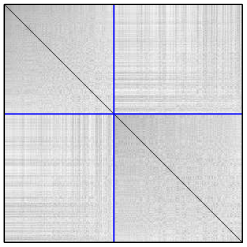


Fig. A.11: Gower's DR for the heart data.

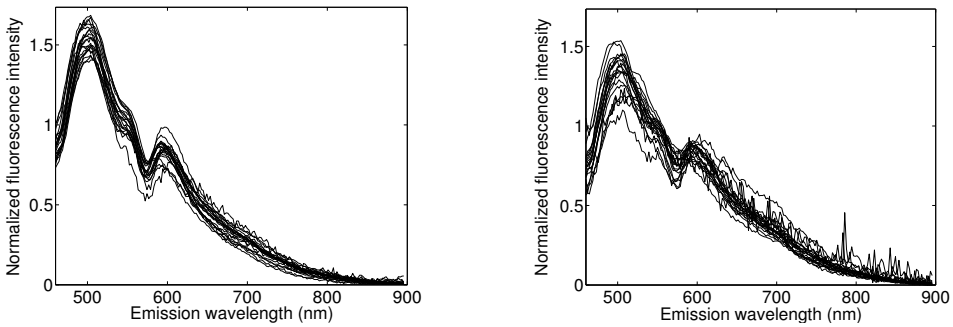


Fig. A.12: Examples of normalized autofluorescence spectra for healthy (left) and diseased (right) patients.

Diseased mucosa in oral cavity. The data consist of the autofluorescence spectra acquired from healthy and diseased mucosa in the oral cavity; see [368, 406] for details. Autofluorescence spectra were collected from 97 volunteers with no clinically observable lesions of the oral mucosa and 137 patients having lesions in oral cavity. The measurements were taken at 11 different anatomical locations using seven different excitation wavelengths 350, 365, 385, 405, 420, 435 and 450 nm. We will, however, concentrate on the wavelength of 365nm, since the corresponding spectra have the smallest number of outliers. After preprocessing [406], each spectrum consists of 199 bins (pixels/wavelengths). In total, 857 spectra representing healthy tissue and 112 spectra representing diseased tissue were obtained. Two normalization techniques have been used here: identical area, i.e. the bins are scaled such that their sum is 100, or standard normal variate (SNV) transformation where each spectrum is standardized to have a zero mean and a unit standard deviation; see Fig. A.12 for some examples.

A number of dissimilarity measures has been considered for normalized spectra. First, the city block distances between first order Gaussian-smoothed ($\sigma = 3$ samples) derivatives of the spectra are computed. The zero-crossings of the derivatives indicate the peaks and valleys of the spectra, so they are informative. Moreover, the distances between smoothed derivatives contain some information of the order of bins. In this way, the property of a continuity of a spectrum is somewhat taken into account. Next, a spherical geodesic distance, Def. 3.47, is also considered, called also a *spectral angle mapper*, since it is popular to measure the similarity between the spectra. The spectra (when properly scaled) can also be treated as histograms-like distributions, which allows us to compare them by divergence measures, section 5.2. This data set is used in chapters 8 and 10 for training and combining one-class classifiers.

Geophysical spectra. The geophysical spectra data set describes two classes. Both classes are geologically heterogeneous, hence multi-modal. Each class is represented by 500 examples. The objects are described by large wavelength spectra, since (hyper-)spectra are popular in remote sensing [239]. Since the data are confidential we cannot provide more details. The spectra are first normalized to a unit area and then two dissimilarity representations are derived. The first one relies on the spectral angle mapper distance (SAM) [239] defined for the spectra $\mathbf{s}_i$ and $\mathbf{s}_j$ as $d_{SAM}(\mathbf{s}_i, \mathbf{s}_j) = \arccos(\mathbf{s}_i^T \mathbf{x}_j / ||\mathbf{s}_j||_2 \, ||\mathbf{s}_j||_2)$ (which is in fact a spherical distance; see Def. 3.47). The second dissimilarity is based on the l_1 distance between the Gaussian smoothed (with $\sigma = 2$ bins) first order derivatives of the spectra [286, 287]. Since by the use of the first derivative, the shape of the spectra is somewhat taken into account, we will refer to this measure as to the shape dissimilarity. Hence, the geophysical data are denoted as *GeoSam* and *GeoShape*, respectively.

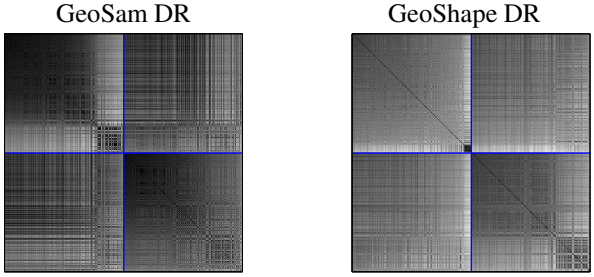


Fig. A.13: Dissimilarity representations for the geophysical spectra.

ProDom. ProDom is a comprehensive set of protein domain families [68]. A *ProDom* subset of 2604 protein domain sequences from the ProDom set [68] was selected by Roth [319]. These are chosen based on a high similarity to at least one sequence contained in the first four folds of the SCOP database. The pairwise structural alignments are computed by Roth [319]. Each SCOP sequence belongs to a group, as labeled by the experts [279]. We use the same four-class problem in our investigations. Originally, a structural similarities s_{ij} are derived, from which the dissimilarities are derived as $d_{ij} = (s_{ii} + s_{jj} - 2s_{ij})^{1/2}$ for $i \neq j$. $D = (d_{ij})$ is slightly non-Euclidean and slightly non-metric.

NIST digit data. This data set describes 2000 handwritten digits from the NIST database [420], each represented by 128×128 binary images; see Fig. A.14 for some examples. Each digit class is represented by 200 examples. Two dissimilarity measures are considered here: Euclidean on the blurred images and modified-Hausdorff, Def. 5.6 on the digit contours. When needed, the images are blurred by the use of the Gaussian function with a standard deviation of 8 pixels. The motivation for such a preprocessing is to avoid sharp edges of the digits and, thereby, make the distances robust to small tilts or variable thickness. This set is used in chapter 9 for the classification task.

Fig. A.14: Examples of the NIST digits, resampled to 16×16 pixels.

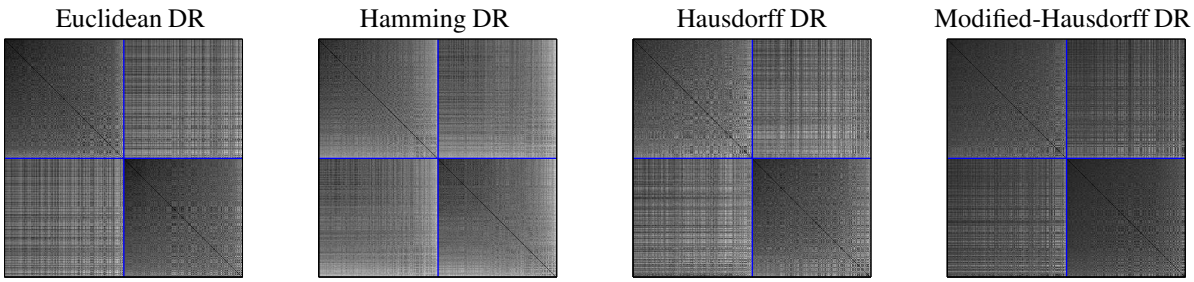


Fig. A.15: Dissimilarity representations for the NIST-38 digit data.

NIST-38 digit data. Within the collection of the NIST digit, a two-class problem is also separately considered, represented by the digits '3' and '8'. Here, each digit class consists of 1000 examples. Four dissimilarity measures are considered: Hamming (section 5.3) Euclidean on the blurred (Gaussian-smoothed) images, Hausdorff (Def. 5.3) and modified-Hausdorff (Def. 5.6) on the digit contours. This set is used in chapter 9 for a simulation of a missing value problem and in chapter 10 for combining strategies.

Zongker digit data. The data describes the NIST digits [420], originally given as 128×128 binary images. Here, the similarity measure, based on deformable template matching, as defined by Zongker and Jain [207], is used. Let $S = (s_{ij})$ denote the similarities. The symmetric dissimilarities $D = (d_{ij})$ are computed as follows: $d_{ij} = (s_{ii} + s_{jj} - s_{ij} - sji)^{1/2}$ for $i \neq j$ and $d_{ii} = 0$, since the data are slightly asymmetric. Note that the latter can be obtained in a traditional way as well, Theorem 3.38, second item, if first the corresponding similarities s_{ij} and s_{ji} are averaged out. Since the original S and its averaged out version S_{avr} are not positive-definite, then D is non-Euclidean. Moreover, D is non-metric, since the triangle inequality does not hold. Since $s_{ij} \in [0, 1]$, in some other cases, we will also distinguish the dissimilarities derived for the averaged similarities as $D = (1 - S_{avr}).^{1/2}$. These are also non-metric.

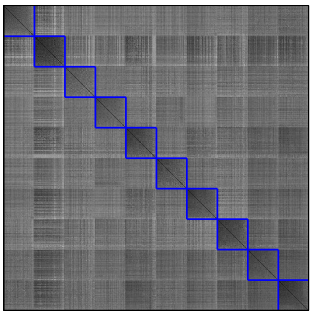


Fig. A.16: Non-metric DR for the Zongker data.

To have an impression of the non-Euclidean aspect of both dissimilarities, an indication can be given by the estimated ratio of $|\lambda_{min}|/\lambda_{max} \in [0.31, 0.38]$, that is in the pseudo-Euclidean embedding process this is the ratio of the largest in magnitude negative eigenvalue to the largest positive one. The overall contribution of negative eigenvalues in terms of the generalized average variance, see section 3.3.4, is about 35%. These numbers imply a significant 'deviation' from Euclidean behavior. This data set is used in chapter 6 for

visualization and in chapter 9 for discrimination. Dr Douglas Zongker and prof. Anil Jain are acknowledged for providing the template-matching dissimilarities on the NIST digits.

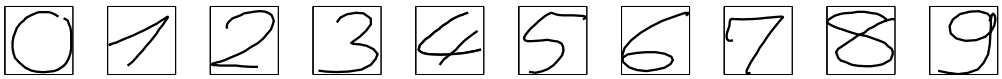


Fig. A.17: Examples of the pen-based handwritten digits.

Pen-based handwritten digit data. This data set comes from the UCI Machine Learning Repository [31] and was created by Alpaydin and Alimoglu. They used a pressure sensitive tablet with an integrated LCD display and a cordless stylus. Samples hand-written by a number of subjects are described by the x and y coordinates within 500×500 pixel box. Hence, each digit is presented as a sequence of points in a 2-dimensional space. First, the data are resampled such that the distances between any consecutive pair of points equal some chosen Δ. Then, from the transformed sequence $s = (x_1, y_1) \dots (x_m, y_m)$, a string $z = \mathbf{z}_1 \dots \mathbf{z}_m$ is derived such that $\mathbf{z}_i$ is the vector pointing from (x_i, y_i) to (x_{i+1}, y_{i+1}). Each digit is then represented by a string. The distance between the strings is an edit distance with a fixed insertion and deletion costs, $c_{\text{ins}} = c_{\text{del}} = \Delta$ and with some substitution cost c_{sub}. Two different substitution costs are considered as an angle between the vectors and the Euclidean distance between the vectors. Different definitions of c_{sub} lead to different distance measures, hence different dissimilarity representations called Pen-angle and Pen-dist, respectively; see also [47, 48].

Here, we only consider a part of the pen-digits data, consisting of 3488 digit examples originally assigned as the 'test' data on the UCI Repository Web-page (actually, all but first samples of each test class are used). The digits are unevenly represented with the class cardinalities varying between 334 and 363. For some examples of original pen-digits data can be seen in Fig. A.17. This data set is used in chapter 9 for classification. We are grateful to prof. Horst Bunke and Simon Günter for providing the edit-distance data.

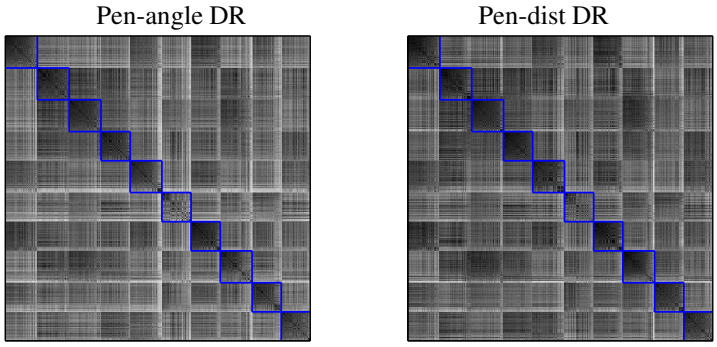


Fig. A.18: Edit-distance representations for the pen-digit data.

Newsgroups data. This is a small part of the so-called *20Newsgroups* data [282], as considered by Roweis [283]. The original data set is a collection of approximately 20000 messages, partitioned (nearly) evenly across 20 different newsgroups. Each newsgroup corresponds to a different topic. Some of the newsgroups are very closely related to each other, while others differ substantially. The full list, partitioned according to the subject matter is: comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, misc.forsale, talk.politics.misc, talk.politics.guns, talk.politics.mideast, talk.religion.misc, alt.atheism and soc.religion.christian. The small subset used here consist of all the 'comp.*', 'rec.*', 'sci.*' and 'talk.*' groups combined into four classes. Each message is then described by an occurrence for 100 words across 16242 postings. Hence, the messages are described by occurrence vectors in a 100-dimensional space.

The non-metric correlation-based dissimilarity measures D_{cor} and D_{cor2}, defined in Table 5.2 are used to construct the *News-cor* and *News-cor2* dissimilarity representations, respectively. Since the occurrence vectors can be treated as describing the event only (a particular keyword has appeared or not), they might be then simplified to binary variables for which some measures can be defined. Also the Jaccard, dice, simple

matching and Hamman measures were investigated; see Table 5.1. However, since the used keywords are not representative, these measures were found very poor. Therefore, we skipped them from the analysis. This data set is used in chapter 6 for visualization and in chapter 7 for illustration of some clustering approaches.

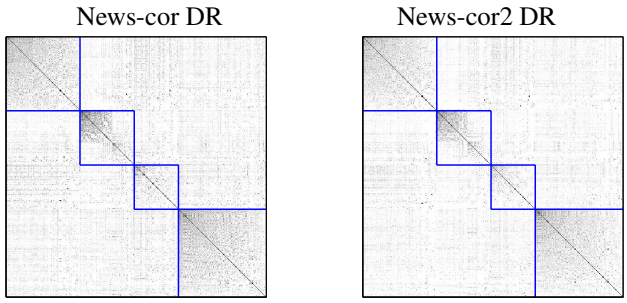


Fig. A.19: Dissimilarity representations for the newsgroup data.

Texture data. These data are created from 23 large images obtained from MIT Media Lab [394] and used as illustration for an image database retrieval problem. Each original image is cut into 16 128×128 non-overlapping pieces. These represent a single class. Therefore, our database consists of 23 classes and 368 images. These images are mostly homogeneous and represent one type of a texture. Each image is described by the responses (in terms of magnitudes) of ten Gabor filters. They are chosen by a backward feature selection from a set of 48 Gabor filters defined by different smoothing, frequency and direction parameters; see also [236].

Bibliography

[1] R. Agarwala, V. Bafna, M. Farach, M. Paterson, and M. Thorup. On the approximability of numerical taxonomy (fitting distances by tree metrics). *SIAM Journal on Computing*, 28(3):1073–1085, 1999.

[2] D.W. Aha, D. Kibler, and M.K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[3] D. Alpay, A. Dijksma, J. Rovnak, and H. de Snoo. *Schur Functions, Operator Colligations, and Reproducing Kernel Pontryagin Spaces*. Birkhäuser Verlag, Basel-Boston-Berlin, 1997.

[4] A.G. Arkadiev and E.M. Braverman. *Teaching a computer pattern recognition*. Nauka, 1964.

[5] C.G. Atkeson, A.W. Moore, and S. Schaal. Locally weighted learning. *AI Review*, 11:11–73, 1997.

[6] P. Avesani, E. Blanzieri, and F. Ricci. Advanced metrics for class-driven similarity search. In *International Workshop on Database and Expert Systems Applications*, pages 223–227, Italy, 1999.

[7] H. Ayad, O. Basir, and M. Kamel. A probabilistic model using information theoretic measures for cluster ensembles. In F. Roli, J. Kittler, and T. Windeatt, editors, *Multiple Classifier Systems, LNCS*, volume 3077, pages 144–153, 2004.

[8] J.P. Barthélemy and A. Guénoche. *Trees and Proximity Representations*. Chichester: Wiley, 1991.

[9] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Distance metric between 3D models and 2D images for recognition and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(4):465–470, 1996.

[10] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[11] R. Basri and D. Jacobs. Constancy and similarity. *Computer Vision and Image Understanding*, 65(3):447–449, 1997.

[12] F.B. Baulieu. A classification of presence/absence based dissimilarity coefficients. *Algebra Universalis*, 20: 351–367, 1989.

[13] F.B. Baulieu. Two variant axiom systems for presence/absence based dissimilarity coefficients. *Journal of Classification*, 14:159–170, 1997.

[14] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems*, volume 14, pages 585–591. The MIT Press, 2002.

[15] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[16] S. Belongie and J. Malik. Matching with shape context. In *IEEE Workshop on Content-based Access of Image and Video Libraries*, pages 20–26, 2000.

[17] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(24):509–522, 2002.

[18] C.H. Bennett, P. Gacs, M. Li, P.M.B. Vitányi, and W. Zurek. Information distance. *IEEE Transactions on Information Theory*, IT-44(4):1407–1423, 1998.

[19] K.P. Bennett and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–24, 1992.

[20] K.P. Bennett and O.L. Mangasarian. Combining support vector and mathematical programming methods for induction. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*, pages 307– 326. MIT Press, Cambridge, MA, 1999.

[21] S. Berchtold, B. Ertl, D.A. Keim, H.-P. Kriegel, and T. Seidl. Fast nearest neighbor search in high-dimensional spaces. In *International Conference on Data Engineering*, Orlando, Florida, 1998.

[22] C. Berg, J.P.R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag, 1984.

[23] J.C. Bezdek and R.J. Hathaway. VAT: A tool for visual assessment of (cluster) tendency. In *International Joint Conference on Neural Networks*, pages 2225–2230, Piscataway, NJ, 2002. IEEE Press.

[24] J.C. Bezdek, J.M. Keller, R. Krishnapuram, and N.R. Pal. *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Boston, 1999.

[25] J.C. Bezdek and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man and Cybernetics*, 28(3):301–315, 1998.

[26] A. Białynicki-Birula. *Algebra liniowa z geometrią*. PWN, Warszawa, 1976.

[27] J. Bilmes. A gentle tutorial on the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. Technical Report ICSI-TR-97-021, Signal, Speech, and Language Interpretation Laboratory, University of Washington, 1997.

[28] A. Birkholc. *Analiza matematyczna. Funkcje wielu zmiennych*. PWN, Warszawa, 1986.

[29] C.M. Bishop, M. Svensén, and C.K. Williams. GTM: a principled alternative to the self-organizing map. In C. Von der Malsburg, C. Von Seelen, J.C. Vorbrggen, and B. Sendhoff, editors, *International Conference on Artificial Neural Networks*, pages 165–170, Berlin, 1996. Springer-Verlag.

[30] C.M. Bishop, M. Svensén, and C.K.I Williams. Developments of the generative topographic mapping. *Neurocomputing*, 21:203–224, 1998.

[31] C.L. Blake and C.J. Merz. UCI repository of machine learning databases. University of California, Irvine, Department of Information and Computer Sciences, 1998. `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[32] L.M. Blumenthal. Remarks concerning the Euclidean four-point property. *Ergebnisse eines Math. Koll.*, 7:8–10, 1936.

[33] L.M. Blumenthal. *Theory and Applications of Distance Geometry*. Oxford University Press, Amen House, London, 1953.

[34] J. Bognár. *Indefinite Inner Product Spaces*. Springer-Verlag, Berlin Heidelberg New York, 1974.

[35] M.M. Bonsangue, F. van Breugel, and J.J.M.M. Rutten. Generalized metric spaces: Completion, topology and powerdomains via the Yoneda embedding. *Theoretical Computer Science*, 193:1–51, 1998.

[36] A. Bookstein, S.T. Klein, and T. Raita. Fuzzy Hamming distance: A new dissimilarity measure. In A. Amir and G.M. Landau, editors, *CPM 2001: LNCS 2089*, pages 86–97, 2001.

[37] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer-Verlag, New York, 1997.

[38] G. Borgefors. Distance transformation in digital images. *Compure Vision, Graphics and Image Processing*, 34: 344–371, 1986.

[39] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52:46–52, 1985.

[40] A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[41] P.S. Bradley, O.L. Mangasarian, and W.N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10:209–217, 1998.

[42] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[43] L. Breiman. Bias, variance, and arcing classifiers. Technical Report 460, Statistic Department, University of California, April 1996.

[44] J. Bretagnolle, D. Dacunha Castelle, and J.L. Krivine. Lois stables et espaces l^p. *Ann. Inst. Henri Poincaré*, II (3):231–259, 1966.

[45] J.M. Buhmann and T. Hofmann. A maximum entropy approach to pairwise data clustering. In *International Conference on Pattern Recognition*, volume II, pages 207–212, Jerusalem, Israel, 1994.

[46] J.M. Buhmann and T. Hofmann. Hierarchical pairwise data clustering by mean-field annealing. In *International Conference on Artificial Neural Networks*, pages 197–202, 1995.

[47] H. Bunke, S. Günter, and X. Jiang. Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. In *International Conference on Advances in Pattern Recognition: Springer LNCS 2013*, pages 1–11, 2001.

[48] H. Bunke, X. Jiang, K. Abegglen, and A. Kandel. On the weighted mean of a pair of strings. *Pattern Analysis and Applications*, 5(1):23–30, 2002.

[49] H. Bunke and A. Sanfeliu, editors. *Syntactic and Structural Pattern Recognition Theory and Applications*. World Scientific, 1990.

[50] H. Bunke and K. Shearer. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8):689–694, 1997.

[51] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.

[52] C.J.C. Burges. Geometry and invariance in kernel based methods. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*. MIT Press, 1998.

[53] C. Campbell and K.P. Bennett. A linear programming approach to novelty detection. In *Advances in Neural Information Processing Systems*, pages 395–401, 2000.

[54] F.A.T. de Carvalho. *New Approaches in Classification and Data Analysis*, chapter Proximity coefficients between Boolean symbolic objects, pages 387–394. Springer-Verlag, 1994.

[55] F.A.T. de Carvalho. *Data Science, Classification and Related Methods*, chapter Extension based proximities between constrained Boolean symbolic objects, pages 370–378. Springer-Verlag, 1998.

[56] A. Cayley. On the theorem in the geometry of position. *Cambridge Mathematical Journal*, II:267–271, 1841.

[57] S.H. Cha and S.N. Srihari. Distance between histograms of angular measurements and its application to handwritten character similarity. In *International Conference on Pattern Recognition*, volume 2, pages 21–24, 2000.

[58] Y. Chabrillac and J.-P. Crouzeix. Definiteness and semidefiniteness of quadratic forms revisited. *Linear Algebra and its Applications*, 63(4):283–292, 1984.

[59] T.Y.T. Chan and L. Goldfarb. Primitive pattern learning. *Pattern Recognition*, 25(8):883–889, 1992.
[60] B.B. Chaudhuri and A. Rosenfeld. On a metric distance between fuzzy sets. *Pattern Recognition Letters*, 17: 1157–1160, 1996.
[61] B.B. Chaudhuri and A. Rosenfeld. On a metric distance between fuzzy sets. *Information Sciences*, 118:159–171, 1999.
[62] E. Čech. *Topological Spaces*. Wiley, London, 1966.
[63] Y. Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.
[64] V. Chepoi and B. Fichet. l_∞-approximation via subdominants. *J. Mathematical Psychology*, 44:600–616, 2000.
[65] D. Cho and D.J. Miller. A Low-complexity Multidimensional Scaling Method Based on Clustering. *concept paper*, 2002.
[66] J. Cohen and M. Farach. Numerical taxonomy on data: Experimental results. *Journal of Computational Biology*, 4(4), 1997.
[67] T. Constantinescu and A. Gheondea. Representations of Hermitian kernels by means of Krein spaces II. invariant kernels. *Communications in Mathematical Physics*, 216:409–430, 2001.
[68] F. Corpet, F. Servant, J. Gouzy, and D. Kahn. Prodom and prodom-cg: tools for protein domain analysis and whole genome comparisons. *Nucleid Acids Research*, 28:267–269, 2000.
[69] L. da Fontoura Costa and Jr. Cesar, R.M. *Shape Analysis and Classification*. CRC Press, Boca Raton, 2001.
[70] P. Courrieu. Straight monotonic embedding of data sets in Euclidean spaces. *Neural Networks*, 15:1185–1196, 2002.
[71] T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
[72] T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1995.
[73] T.F. Cox and M.A.A. Cox. A General Weighted Two-Way Dissimilarity Coefficient. *Journal of Classification*, 17:101–121, 2000.
[74] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, UK, 2000.
[75] F. Critchley and B. Fichet. On (Super-)Spherical Distance Matrices and Two Results from Schoenberg. *Linear Algebra and its Applications*, 251:145–165, 1997.
[76] I. Csiszár. Information-type measures of divergence of probability distributions and indirect observations. *Studia Scientiarium Mathematicarum Hungarica*, 2:299–318, 1967.
[77] B.V. Dasarthy, editor. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, CA, 1991.
[78] B.V. Dasarthy. Minimal consistent set (MCS) identification for optimal nearest neighbor decision systems design. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(3):511–517, 1994.
[79] I.M. de Diego, J.M. Moguerza, and A. Muñoz. Combining kernel information for support vector classification. In F. Roli, J. Kittler, and T. Windeatt, editors, *Multiple Classifier Systems, LNCS*, volume 3077, pages 102–111. 2004.
[80] L. Debnath and P. Mikusinski. *Introduction to Hilbert Spaces with Applications*. Academic Press, San Diego, 1990.
[81] DELVE. Data for evaluating learning in valid experiments. University of Toronto, Department of Computer Science. `http://www.cs.toronto.edu/~delve/`.
[82] P. Demartines and J. Hérault. Curvilinear component annalysis: A self-organizing neural network for nonlinear mapping of data sets. *IEEE Transations on Neural Networks*, 8(1):148–154, 1997.
[83] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society*, Series B, 39(1):1–38, 1977.
[84] T. Denœux and et al. Belief functions and pattern recognition: Matlab software. `http://www.hds.utc.fr/~tdenoeux/software.htm`.
[85] T. Denoeux and M.-H. Masson. Evclus: Evidential clustering of proximity data. *IEEE Transations on Systems, Man and Cybernetics*, 34(1):95–109, 2004.
[86] P.A. Devijver and J. Kittler. *Pattern recognition: A statistical approach*. Prentice/Hall, London, 1982.
[87] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.
[88] M. Deza and M. Laurent. Applications of cut polyhedra. *Journal of Computational and Applied Mathematics*, 55(2):217 – 247, 1994.
[89] C. Domeniconi, J. Peng, and D. Gunopulos. Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1281–1285, 2002.
[90] P. Domingos. A unified bias-variance decomposition and its applications. In *International Conference on Machine Learning*, pages 231–238. Morgan Kaufmann, 2000.
[91] P. Domingos. A unified bias-variance decomposition for zero-one and squared loss. In *International Conference*

on Artificial Intelligence, pages 564–569, Austin, Texas, 2000. AAAI Press.

[92] M.A. Dritschel and J. Rovnyak. Operators on indefinite inner product spaces. *Lectures on Operator Theory and its Applications, Fields Institute Monographs*, pages 141–232, 1996.

[93] M.P. Dubuisson and A.K. Jain. Modified Hausdorff distance for object matching. In *International Conference on Pattern Recognition*, volume 1, pages 566–568, 1994.

[94] W. Duch. Similarity based methods: a general framework for classification, approximation and association. *Control and Cybernetics*, 29(4):937–968, 2000.

[95] W. Duch, R. Adamczak, and G.H.F. Diercksen. Classification, association and pattern completion using neural similarity based methods. *Applied Mathematics and Computer Science*, 10(4):101–120, 2000.

[96] W. Duch, A. Naud, and R. Adamczak. A framework for similarity-based methods. In *Polish Conference on Theory and Applications of Artificial Intelligence*, pages 33–60, Łód'z, 1998.

[97] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2nd edition, 2001.

[98] R.P.W. Duin. Compactness and complexity of pattern recognition problems. In *International Symposium on Pattern Recognition 'In Memoriam Pierre Devijver'*, pages 124–128, Royal Military Academy, Brussels, 1999.

[99] R.P.W. Duin. Classifiers in almost empty spaces. In *International Conference on Pattern Recognition*, volume 2, pages 1–7, Barcelona, Spain, 2000.

[100] R.P.W. Duin. The combining classifier: To train or not to train? In *International Conference on Pattern Recognition*, volume II, pages 765–770, Quebec City, Canada, 2002.

[101] R.P.W. Duin, P. Juszczak, D. de Ridder, P. Paclík, E. Pękalska, and D.M.J. Tax, 2004. `http://prtools.org`. PR-Tools, a Matlab toolbox for pattern recognition.

[102] R.P.W. Duin and E. Pękalska. Complexity of dissimilarity based pattern classes. In *Scandinavian Conference on Image Analysis*, Bergen, Norway, 2001.

[103] R.P.W. Duin and E. Pękalska. Possibilities of zero-error recognition by dissimilarity representations. In J.M. Inesta and L. Mico, editors, *Pattern Recognition in Information Systems*, Allicante, Spain, 2002.

[104] R.P.W. Duin and E. Pękalska. Object representation, sample size and data set complexity. submitted, 2004.

[105] R.P.W. Duin, E. Pękalska, P. Paclík, and D.M.J. Tax. The dissimilarity representation, a basis for domain based pattern recognition? In L. Goldfarb, editor, *Pattern representation and the future of pattern recognition, ICPR 2004 Workshop Proceedings*, pages 43–56, Cambridge, United Kingdom, 2004.

[106] R.P.W. Duin, E. Pękalska, and D. de Ridder. Relational Discriminant Analysis. *Pattern Recognition Letters*, 20 (11-13):1175–1181, 1999.

[107] R.P.W. Duin, E. Pękalska, M. Skurichina, D. de Veld, H.J.C.M. Sterenborg, M.J.H. Witjes, and L.N. Roodenburg. Combined classifiers for dissimilarity based representations of auto-fluorescence spectra applied to lesion recognition. *IEEE Transactions on Systems, Man and Cybernetics*, accepted, 2005.

[108] R.P.W. Duin, D. de Ridder, and D.M.J. Tax. Experiments with object based discriminant functions; a featureless approach to pattern recognition. *Pattern Recognition Letters*, 18(11-13):1159–1166, 1997.

[109] R.P.W. Duin, D. de Ridder, and D.M.J. Tax. Featureless pattern classification. *Kybernetika*, 34(4):399–404, 1998.

[110] R.P.W. Duin, F. Roli, and D. de Ridder. A note on core research issues for statistical pattern recognition. *Pattern Recognition Letters*, 23(4):493–499, 2002.

[111] R.P.W. Duin and D.M.J. Tax. Classifier conditional posterior probabilities. In *Advances in Pattern Recognition, LNCS*, volume 1451, pages 611–619, Sydney, 1998. Joint IAPR International Workshops on SSPR and SPR.

[112] N. Dunford and J.T. Schwarz. *Linear operators. Part I: general theory*. Interscience Publishers, Inc., New York, 1958.

[113] S. Edelman. *Representation and Recognition in Vision*. MIT Press, Cambridge, 1999.

[114] S. Edelman, S. Cutzu, and S. Duvdevani-Bar. Similarity to reference shapes as a basis for shape representation. *Cognitive Science Conference*, 1996.

[115] S. Edelman, S. Cutzu, and S. Duvdevani-Bar. Representation is representation of similarities. *Behavioral and Brain Sciences*, 21:449–498, 1998.

[116] S. Edelman and S. Duvdevani-Bar. Similarity, connectionism, and the problem of representation in vision. *Neural Computation*, 9:701–720, 1997.

[117] T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2): 109–133, 1997.

[118] F. Esposito, D. Malerba, V. Tamma, H.H. Bock, and F.A. Lisi. *Analysis of Symbolic Data*, chapter Similarity and Dissimilarity. Springer-Verlag, 2000.

[119] B.S. Everitt, S. Landau, and M. Leese. *Cluster Analysis*. 4th edition, Arnold, London, 2001.

[120] B.S. Everitt and S. Rabe-Hesketh. *The Analysis of Proximity Data*. Arnold, London, 1997.

[121] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

[122] C. Faloutsos and K.-I. Lin. Fastmap: A fast algorithm for indexing, data-mining and visualization of tradditional

and multimedia datasets. In *ACM SIGMOD, International Conference on Management of Data*, pages 163–174, California, 1995.

[123] M. Farach, S. Kannan, and T. Warnow. A robust model for finding optimal evolutionary trees. *Algorithmica*, 13: 155–179, 1995.

[124] Fault data. `http://www.sidanet.org`.

[125] G.M. Fichtenholz. *Rachunek różniczkowy i całkowy*. Państwowe Wydawnictwo Naukowe, Warszawa, 1997.

[126] B. Fischer, T. Thomas Zöller, and J.M. Buhmann. Path based pairwise data clustering with application to texture segmentation. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 235–250, 2001.

[127] fish. Fish contours. University of Surrey. `http://www.ee.surrey.ac.uk/Personal/F.Mokhtarian/`.

[128] C. Fraley and A.E. Raftery. How many clusters? Which clustering method? Answers via model-based cluster analysis. *Computer Journal*, (41):578–588, 1998.

[129] A. Fred and A.K. Jain. Data clustering using evidence accumulation. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *International Conference on Pattern Recognition*, pages 276–280, Quebec City, Canada, 2002.

[130] A. Fred and A.K. Jain. Robust data clustering. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 442 –451, Madison - Wisconsin, USA, 2002.

[131] A. Fred and A.K. Jain. Evidence accumulation clustering based on the k-means algorithm. In *Structural, Syntactic, and Statistical Pattern Recognition, LNCS vol. 2396*, volume II, pages 128–133. Springer-Verlag, 2003.

[132] A. Fred and J. Leitão. A new cluster isolation criterion based on dissimilarity increments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(8):944–958, 2003.

[133] H. Freeman and J.M. Glass. On the encoding of arbitrary geometric configurations. *IRE Transactions*, EC-10 (2):260–268, September 1961.

[134] C. Frélicot and H. Emptoz. A pretopological approach for pattern classification with reject options. In A. Amin, D. Dori, P. Pudil, and H. Freeman, editors, *Joint IAPR International Workshops on SSPR and SPR, LNCS*, volume 1451, pages 707–715. Springer, 1998.

[135] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proc. of the 13th International Conference*, pages 148–156, 1996.

[136] J.H. Friedman. Flexible metric nearest neighbor classification. Technical Report 113, Stanford University Statistics Department, 1994.

[137] K.S. Fu. *Syntactic Pattern Recognition and Applications*. Pretice-Hall, 1982.

[138] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[139] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Molecular Biology and Evolution*, 14:685–695, 1997.

[140] O. Gascuel. Data model and classification by trees: The minimum variance reduction (MVR) method. *Journal of Classification*, 17:67–99, 2000.

[141] G.C. Gastl and Hammer P.C. Extended topology. neighboorhoods and convergents. In *Colloquium on Convexity 1965*, pages 104–116, Copenhagen, 1967. Kobenhavns Univ. Matematiske Inst.

[142] D.M. Gavrila. Pedestrian detection from a moving vehicle. In *European Conference on Computer Vision*, Dublin, Ireland, 2000.

[143] D.M. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *IEEE International Conference on Computer Vision*, Kerkyra, 1999.

[144] Y. Gdalyahu and D. Weinshall. Flexible syntactic matching of curves and its application to automatic hierarchical classification of silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(12), 1999.

[145] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[146] A. Gibbs and F. Su. On choosing and bounding probability metrics. *International Statistical Review*, 70(3): 419–435, 2002.

[147] F. Girosi. An equivalence between sparse approximation and support vector machines. *Neural Computation*, 10 (6):1455–1480, 1998.

[148] S. Gniłka. On extended topologies. i: Closure operators. *Commentationes Mathematicae*, 34:81–94, 1994.

[149] S. Gniłka. On extended topologies. ii: Compactness, quasi-metrizability, symmetry. *Commentationes Mathematicae*, 35:147–162, 1995.

[150] S. Gniłka. On continuity in extended topologies. *Annales Societatis Mathematicae Polonae. Seria I. Commentationes Mathematicae*, 37:99–108, 1997.

[151] L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17:575–582, 1984.

[152] L. Goldfarb. A new approach to pattern recognition. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 241–402. Elsevier Science Publishers BV, 1985.

[153] L. Goldfarb. On the foundations of intelligent processes – I. An evolving model for pattern recognition. *Pattern*

Recognition, 23(6):595–616, 1990.

[154] L. Goldfarb. What is distance and why do we need the metric model for pattern learning? *Pattern Recognition*, 25(4):431–438, 1992.

[155] L. Goldfarb, J. Abela, V.C. Bhavsar, and V.N. Kamat. Transformation systems are more economical and informative class descriptions than formal grammars. In *International Conference on Pattern Recognition*, volume II, pages 660–664, The Netherlands, 1992.

[156] L. Goldfarb, J. Abela, V.C. Bhavsar, and V.N. Kamat. Can a vector space based learning model discover inductive class generalization in a symbolic environment? *Pattern Recognition Letters*, 16(7):719–726, 1995.

[157] L. Goldfarb and S. Deshpande. What is a symbolic measurement process? In *Systems, Man and Cybernetics*, volume 5, pages 4139–4145, Orlando, Florida, 1997.

[158] L. Goldfarb, S. Deshpande, and V.C. Bhavsar. Inductive theory of vision. Technical Report TR96-108, University of New Brunswick, Fredericton, Canada, 1996.

[159] L. Goldfarb, D. Gay, O. Golubitsky, and D. Korkin. What is a structural representation? second version. Technical Report TR04-165, University of New Brunswick, Fredericton, Canada, 2004.

[160] L. Goldfarb and O. Golubitsky. What is a structural measurement process? Technical Report TR01-147, University of New Brunswick, Fredericton, Canada, 2001.

[161] L. Goldfarb, O. Golubitsky, and D. Korkin. What is a structural representation? Technical Report TR00-137, University of New Brunswick, Fredericton, Canada, 2000.

[162] L. Goldfarb, O. Golubitsky, and D. Korkin. What is a structural representation in chemistry? Towards a unified framework for CADD. Technical Report TR00-138, University of New Brunswick, Fredericton, Canada, 2000.

[163] L. Goldfarb and R. Verma. Hybrid associative memories and metric data models. In *SPIE, Digital and Optical Shape Representation and Pattern Recognition*, volume 938, 1988.

[164] R.L. Goldstone. Similarity, interactive activation, and mapping. *Journal of Experimental Psychology*, 20:3–28, 1994.

[165] R.L. Goldstone. Hanging together: A connectionist model of similarity. In J. Grainger and A.M. Jacobs, editors, *Localized Connectionist Approaches to Human Cognition*, pages 283–325. NJ: Lawrence Erlbaum Associates, Mahwah, 1998.

[166] R.L. Goldstone. Similarity. In R.A. Wilson and F.C. Keil, editors, *MIT encyclopedia of the cognitive sciences*, pages 763–765. MA: MIT Press, Cambridge, 1999.

[167] A.D. Gordon. *Clustering and Classification*, chapter Hierarchical Classification, pages 65–122. London: World Scientific, 1996.

[168] K.C. Gowda and E. Diday. Unsupervised learning through symbolic clustering. *Pattern Recognition Letters*, 12: 259–264, 1991.

[169] J. Gower and G. Ross. Minimum spanning trees and single linkage cluster analysis. *Applied Statistics*, 18:54–64, 1969.

[170] J.C. Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 27:25–33, 1971.

[171] J.C. Gower. Metric and Euclidean Properties of Dissimilarity Coefficients. *Journal of Classification*, 3:5–48, 1986.

[172] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In *Advances in Neural Information System Processing 11*, pages 438–444, 1999.

[173] T. Graepel, R. Herbrich, B. Schölkopf, A. Smola, P. Bartlett, K.-R. Müller, K. Obermayer, and R. Williamson. Classification on proximity data with LP-machines. In *International Conference on Artificial Neural Networks*, pages 304–309, 1999.

[174] U. Grenander. *Pattern synthesis: Lectures in pattern theory, volume 1*. Springer-Verlag, 1976.

[175] U. Grenander. *Pattern analysis: Lectures in pattern theory, volume 2*. Springer-Verlag, 1978.

[176] U. Grenander. *Regular structures: Lectures in pattern theory, volume 3*. Springer-Verlag, 1981.

[177] W. Greub. *Linear Algebra*. Springer-Verlag, 1975.

[178] A.D. Griffiths and D.G. Bridge. Towards a theory of optimal similarity measures. In *Workshop on Case-Based Reasoning*, United Kingdom, 1997.

[179] P.J. Grother, G.T. Candela, and J.L. Blue. Fast implementations of nearest-neighbor classifiers. *Pattern Recognition*, 30(3):459–465, 1997.

[180] A. Guérin-Dugué, P. Teissier, G. Delso Gafaro, and J. Hérault. Curvilinear component analysis for high-dimensional data representation: II. examples of additional mapping constraints in specific applications. In *Conference on Artificial and Natural Neural Networks, LNCS 1607*, pages 635–644, Spain, 1999.

[181] P. Guo, C.L.P. Chen, and M.R. Lyu. Cluster number selection for a small set of samples using the bayesian ying-yang model. *IEEE Transactions on Neural Networks*, 13(3):757–763, 2002.

[182] B. Haasdonk. Feature space interpretation of svms with non positive definite kernels. *Internal report 1/03, Department of Pattern Recognition and Image Processing, Freiburg University, submitted to a journal*, 2003.

[183] M. Hagedoorn and R.C. Veltkamp. Reliable and efficient pattern matching using an affine invariant metric.

International Journal of Computer Vision, 31(2-3):103–115, 1999.
[184] M. Halkidi, Y. Batistakis, and M. Vazirgiannis. On clustering validation techniques. *Intelligent Information Systems Journal*, 17(2-3):107–145, 2001.
[185] P.R. Halmos. *Measure Theory*. Springer-Verlag, New York, 1974.
[186] D.J. Hand. *Construction and Assesment of Classification Rules*. John Wiley & Sons, Chchester, England, 1997.
[187] P.E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14:515–516, 1968.
[188] J. Hartigan. *Clustering Algorithms*. Wiley, New York, NY, 1975.
[189] T. Hastie and W. Stuetzle. Principal curves. *Journal of the American Statistical Association*, 84:502–516, 1989.
[190] T. Hastie and R. Tibshirani. Discriminant adaptive nearest neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(6):607–616, 1996.
[191] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer Verlag, New York Berlin Heidelberg, 2001.
[192] R.J. Hathaway and J.C. Bezdek. Visual cluster validity (VCV) for prototype generator clustering models. *Pattern Recognition Letters*, 24(9-10):1563–1569, 2003.
[193] W.J. Heiser. A generalized majorization method for least squares multidimensional scaling of pseudodistances that may be negative. *Psychometrica*, 56:7–27, 1991.
[194] J. Hérault, C. Jausions-Picaud, and A. Guérin-Dugué. Curvilinear component analysis for high dimensional data representation: I. theoretical aspects and practical use in the presence of noise. In *Conference on Artificial and Natural Neural Networks, LNCS 1607*, pages 625–634, Spain, 1999.
[195] P. Hjort, P. Lisonék, S. Markvorsen, and C. Thomassen. Finite metric spaces of strictly negative type. *Linear Algebra and Its Applications*, 270:255–273, 1998.
[196] T.K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
[197] T. Hofmann and J.M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(1):1–14, 1997.
[198] F. Höppner, F. Klawonn, R. Kruse, and T. A. Runkler. *Fuzzy Cluster Analysis*. Chichester, England, 1999.
[199] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University, Oxford, 1991.
[200] N.P. Hughes and D. Lowe. Artefactual structure from least squares multidimensional scaling. In *Advances in Neural Information Processing Systems (NIPS 2002)*. MIT Press, 2003.
[201] D.P. Huttenlocher, G.A. Klanderman, and J.R. William. Comparing images using the Hausdorff distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.
[202] M. Ichino and H. Yaguchi. General Minkowski metrics for mixed features type data analysis. *IEEE Transaction on System, Man and Cybernetics*, 24:698–708, 1994.
[203] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Annual Symposium on Foundations of Computer Science*, pages 10–33, Las Vegas, Nevada, 2001.
[204] I.S. Iohvidov, M.G. Kreĭn, and H. Langer. *Introduction to the Spectral Theory of Operators in Spaces with an Indefinite Metric*. Akademie-Verlag, Berlin, 1982.
[205] Isomap - homepage. `http://isomap.stanford.edu/`.
[206] D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with Non-Metric Distances: Image Retrieval and Class Representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.
[207] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.
[208] A. K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics*, volume 2, pages 835–855. North-Holland, Amsterdam, 1987.
[209] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.
[210] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.
[211] A.K. Jain, M.N. Murthy, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
[212] N. Jardine and R. Sibson. *Mathematical Taxonomy*. Wiley, London, 1971.
[213] J.B. Kelly. Hypermetric spaces. In L.M. Kelly, editor, *The Geometry of Metric and Linear Spaces, Lecture Notes in Mathematics*, volume 490, pages 17–31. Springer, 1970.
[214] E. Khalimsky. Topological structures in computer science. *Journal of Applied Mathematics and Simulation*, 1: 25–40, 1987.
[215] E. Khalimsky, R. Kopperman, and P.R. Meyer. Computer graphics and connected topologies on finite ordered sets. *Topology and its Applications*, 36:1–17, 1990.
[216] J. Kim and T. Warnow. Tutorial on phylogenetic tree estimation. In *Intelligent Systems for Molecular Biology*, Heidelberg, Germany, 1999.

[217] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
[218] E. Klein and A. Thompson. *Theory of Correspondences*. John Wiley & Sons, New York, 1984.
[219] T. Kohonen. *Self-organizing maps*. Springer-Verlag, Heidelberg, Germany, 1995.
[220] T. Kohonen. *Self-organizing maps*. Springer-Verlag, Heidelberg, Germany, 2000, 3rd edition.
[221] T.Y. Kong, R. Kopperman, and P.R. Meyer. A topological approach to digital topology. *American Mathematical Monthly*, 98:901–917, 1991.
[222] Kong2. Special issue on digital topology. *Topology and its Applications*, 46, 1992.
[223] D. Korkin and L. Goldfarb. Multiple genome rearrangement: a general approach via the evolutionary genome graph. *Bioinformatics*, 18:303–311, 2002.
[224] G. Köthe. *Topological vector spaces I*. Springer-Verlag, Berlin, Heidelberg, New York, 1969.
[225] E. Kreyszig. *Introductory Functional Ananlysis with Applications*. John Wiley & Sons, New York, 1978.
[226] J.B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29:1–27, 1964.
[227] J.B. Kruskal. Multidimensional scaling and other methods for discovering structure. In *Statistical methods for digital computers*, pages 296–339. John Wiley & Sons, New York, 1977.
[228] J.B. Kruskal and M. Wish. *Multidimensional scaling*. Sage Publications, Newbury Park, CA, 1978.
[229] W. Krysicki, J. Bartos, W. Dyczka, K. Królikowska, and M. Wasilewski. *Rachunek prawdopodobieństwa i statystyka matematyczna w zadaniach, część I i II*. Państwowe Wydawnictwo Naukowe, Warszawa, 1995.
[230] L.I. Kuncheva. *Combining Pattern Classifiers. Methods and Algorithms*. Wiley, 2004.
[231] L.I. Kuncheva, M. Skurichina, and R.P.W. Duin. An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion*, 3(2):245–258, 2002.
[232] L.I. Kuncheva and C.J. Whitaker. Using diversity with three variants of boosting: aggressive, conservative and inverse. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 2364, pages 81–90. Springer-Verlag, 2002.
[233] L.I. Kuncheva and C.J. Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181 – 207, 2003.
[234] S. Kurcyusz. *Matematyczne podstawy teorii optymalizacji*. PWN, Warszawa, 1982.
[235] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pękalska, and P. Paclik. A study on combining image representations for image classification and retrieval. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(5): 867–890, 2004.
[236] C. Lai, D.M.J. Tax, E. Pękalska, R.P.W. Duin, and P. Paclík. On combining one-class classifiers for image database retrieval. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 2364, pages 212–221. Springer-Verlag, 2002.
[237] L. Lam. Classifier combinations: implementation and theoretical issues. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 1857, pages 78–86, 2000.
[238] G.N. Lance and W.T. Williams. A general theory of classificatory sorting strategies. I. hierarchical systems. *Computer Journal*, (9):373–380, 1967.
[239] D. Landgrebe. *Signal theory methods in multispectral remote sensing*. John Wiley & Sons, 2003.
[240] F. Lebourgeois and H. Emptoz. Pretopological approach for supervised learning. In *International Conference on Pattern Recognition*, pages 256–260, Los Alamitos, CA, 1996. IAPR, IEEE Computer Society Press.
[241] J.A. Lee, A. Lendasse, N. Donckers, and M. Verleysen. A robust nonlinear projection method. In *European Symposium on Artificial Neural Networks*, pages 13–20, Bruges, Belgium, 2000.
[242] J.A. Lee, A. Lendasse, and M. Verleysen. Curvilinear distance analysis versus isomap. In *European Symposium on Artificial Neural Networks*, pages 185–192, Bruges, Belgium, 2002.
[243] A.J. Lemin. Isometric embeddings of isosceles (non-archimedean) spaces in Euclidean spaces. *Soviet Math. Dokl.*, 32(3):740–744, 1985.
[244] V.I. Levenshtein. Binary codes capable of correcting delations, insertions and reversals. *Soviet Phys. Dokl.*, 6: 707–710, 1966.
[245] E. Levina and P.J. Bickel. The earth mover's distance is the Mallows distance: Some insights from statistics. In *International Conference on Computer Vision*, Vancouver, Canada, 2001.
[246] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi. The similarity metric. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 863–872, Baltimore, Maryland, USA, 2003.
[247] R. Ligteringen, R.P.W. Duin, E.E.E. Frietman, and A. Ypma. Machine diagnostics by neural networks, experimental setup. In P.P. Jonker H.E. Bal, H. Corporaal and J.F.M. Tonino, editors, *Annual Conference of the Advanced School for Computing and Imaging*, pages 185–190, The Netherlands, 1997.
[248] D. Lin. An information-theoretic definition of similarity. In *International Conference on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.
[249] T.-L. Liu and D. Geiger. Approximate tree matching and shape similarity. In *International Conference on*

Computer Vision, pages 456–462, Greece, 1999.
[250] Locally Linear Embedding - homepage. http://www.cs.toronto.edu/~roweis/lle/.
[251] D.G. Lowe. Similarity metric learning for a variable-kernel classifier. *Neural Computation*, 7(1):72–85, 1995.
[252] M. Lee: similarity judgements. http://www.psychology.adelaide.edu.au/members/staff/michaellee.html.
[253] J.B. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, Berkeley, CA, 1967.
[254] D. Malerba, F. Esposito, V. Gioviale, and V. V. Tamma. Comparing dissimilarity measures in symbolic data analysis. In *Joint Conferences on 'New Techniques and Technologies for Statistcs' and 'Exchange of Technology and Know-how'*, pages 473–481, 2001.
[255] S.W. Malone, P. Tarazaga, and M.W. Trosset. Better initial configurations for metric multidimensional scaling. *Computational Statistics and Data Analysis*, 41:143–156, 2002.
[256] S.W. Malone and M.W. Trosset. A study of the stationary configurations of the sstress criterion for metric multidimensional scaling. Technical Report 00-06, Department of Computational and Applied Mathematics, Rice University, 2000.
[257] O.L. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24(1-2):15–23, 1999.
[258] B.F.J. Manly. *Multivariate Statistical Methods*. Chapman & Hall, Englewood Cliffs, New Jersey, 1994.
[259] C. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. , MIT Press, Cambridge, MA, 1999.
[260] J. Mao and A.K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Transactions on Neural Networks*, 6:296–317, 1995.
[261] T. Martinez and K. Schulten. Topology representing networks. *Neural Networks*, 7:507–523, 1994.
[262] A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):926 – 932, 1993.
[263] Mathworl. http://mathworld.wolfram.com.
[264] J. Matousek. *Lectures on Discrete Geometry*. Springer GTM Series, 2002.
[265] G.J. McLachlan and K.E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.
[266] MCS00. Multiple Classifier Systems, LNCS, 2000.
[267] MCS02. Multiple Classifier Systems, LNCS, 2002.
[268] K. Menger. New foundation of Euclidean geometry. *American Journal of Mathematics*, 53:721–745, 1931.
[269] MFEAT. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/.
[270] L. Micó and J. Oncina. Comparison of fast nearest neighbour classifiers for handwritten character recognition. *Pattern Recognition Letters*, 19(3-4):351–356, 1998.
[271] L. Micó, J. Oncina, and R.C. Carrasco. A fast branch & bound nearest neighbour classifier in metric spaces. *Pattern Recognition Letters*, 17(7):731–739, 1996.
[272] M.F. Møler. A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 6:525–533, 1993.
[273] F. Moreno-Seco, L. Micó, and J. Oncina. A modification of the LAESA algorithm for approximated k-nn clasification. *Pattern Recognition Letters*, 24(1-3):47–53, 2003.
[274] G. Mori, S. Belongie, and H. Malik. Shape contexts enable efficient retrieval of similar shapes. In *Computer Vision and Pattern Recognition*, volume 1, pages 723–730. 2001.
[275] V.V. Mottl, S.D. Dvoenko, O.S. Seredin, C.A. Kulikowski, and I.B. Muchnik. Featureless pattern recognition in an imaginary Hilbert space and its application to protein fold classification. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 322–336, Leipzig, 2001.
[276] V.V. Mottl, S.D. Dvoenko, O.S. Seredin, C.A. Kulikowski, and I.B. Muchnik. Featureless regularized recognition of protein fold classes in a hilbert space of pairwise alignment scores as inner products of amino acid sequences. *Pattern Recognition and Image Analysis, Advances in Mathematical Theory and Applications*, 11(3):597–615, 2001.
[277] A. Muñoz, I.M. de Diego, and J.M. Moguerza. Support vector machine classifiers for asymmetric proximities. In *International Conference on Artificial Neural Networks*, pages 217–224, Istanbul, Turkey, 2003.
[278] J.R. Munkres. *Topology*. Prentice-Hall, Englewood Cliffs, New Jersey, 2000, second edition.
[279] A.G. Murzin, S.E. Brenner, T. Hubbard, and C. Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.
[280] M. Nadler and E.P. Smith. *Pattern recognition engineering*. John Willey & Sons Inc., New York, Chichester, Brisgbane, Toronto, Singapore, 1993.
[281] G. Navarro. A guided tour to approximate string matching. *ACM computing surveys*, 33(1):31–88, 2001.
[282] Newsgroups data. http://www.ai.mit.edu/~jrennie/20Newsgroups/.
[283] Newsgroups data: a subset. http://www.cs.toronto.edu/~roweis/data.html.

[284] A.Y. Ng, M.I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, volume 14. The MIT Press, 2002.

[285] B. Noble and J.W. Daniel. *Applied linear algebra*. Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

[286] P. Paclík and R.P.W. R.P.W. Duin. Classifying spectral data using relational representation. In *Spectral Imaging Workshop*, Graz, Austria, 2003.

[287] P. Paclík and R.P.W. R.P.W. Duin. Dissimilarity-based classification of spectra: computational issues. *Real Time Imaging*, 9(4):237–244, 2003.

[288] R. Paredes and E. Vidal. A class-dependent weighted dissimilarity measure for nearest neighbor classification problems. *Pattern Recognition Letters*, 21(12):1027–1036, 2000.

[289] E. Pękalska. *Dealing with the data flood. Mining data, text and multimedia*, chapter Introduction to Multidimensional Scaling. STT/Beweton, The Hague, The Netherlands, 2002.

[290] E. Pękalska and R.P.W. Duin. Classifiers for dissimilarity-based pattern recognition. In *International Conference on Pattern Recognition*, volume 2, pages 12–16, Barcelona, Spain, 2000.

[291] E. Pękalska and R.P.W. Duin. Automatic pattern recognition by similarity representations. *Electronic Letters*, 37(3):159–160, 2001.

[292] E. Pękalska and R.P.W. Duin. On combining dissimilarity representations. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 2096, pages 359–368. Springer Verlag, 2001.

[293] E. Pękalska and R.P.W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, 2002.

[294] E. Pękalska and R.P.W. Duin. Prototype selection for finding efficient representations of dissimilarity data. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *International Conference on Pattern Recognition*, volume 3, pages 37–40, Quebec City, Canada, 2002.

[295] E. Pękalska and R.P.W. Duin. Spatial representation of dissimilarity data via lower-complexity linear and nonlinear mappings. In T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, and de D. Ridder, editors, *International Workshop on SPR + SSPR, LNCS*, volume 2396, pages 470–478. Springer-Verlag, 2002.

[296] E. Pękalska, R.P.W. Duin, S. Günter, and H. Bunke. On not making dissimilarities euclidean. In T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, and de D. Ridder, editors, *Joint IAPR International Workshops on SSPR and SPR, LNCS*, pages 1145–1154. Springer-Verlag, 2004.

[297] E. Pękalska, R.P.W. Duin, M. Kraaijveld, and D. De Ridder. An overview of Multidimensional Scaling techniques with application to Shell data. Technical Report TN-97-036-1, Pattern Recognition Group, Delft University of Technology, The Netherlands, 1998.

[298] E. Pękalska, R.P.W. Duin, M. Kraaijveld, and D. De Ridder. Multidimensional Scaling: Applications to Shell data. Technical Report TN-97-036-3, Pattern Recognition Group, Delft University of Technology, The Netherlands, 1998.

[299] E. Pękalska, R.P.W. Duin, M. Kraaijveld, and D. De Ridder. Multidimensional Scaling: Theoretical Aspects. Technical Report TN-97-036-2, Pattern Recognition Group, Delft University of Technology, The Netherlands, 1998.

[300] E. Pękalska, R.P.W. Duin, and P. Paclík. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, accepted, 2005.

[301] E. Pękalska, P. Paclík, and R.P.W. Duin. A Generalized Kernel Approach to Dissimilarity Based Classification. *Journal of Machine Learning Research*, 2(2):175–211, 2002.

[302] E. Pękalska, D. de Ridder, R.P.W. Duin, and M.A. Kraaijveld. A new method of generalizing Sammon mapping with application to algorithm speed-up. In *Conference of the Advanced School for Computing and Imaging*, pages 221–228, Heijen, The Netherlands, 1999.

[303] E. Pękalska, M. Skurichina, and R.P.W. Duin. Combining Fisher Linear Discriminants for Dissimilarity Representations. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 1857, pages 117–126. Springer Verlag, 2000.

[304] E. Pękalska, M. Skurichina, and R.P.W. Duin. A Discussion on the Classifier Projection Space for Classifier Combining. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 2364, pages 137–148. Springer Verlag, 2002.

[305] E. Pękalska, M. Skurichina, and R.P.W. Duin. Combining Dissimilarity Representations in One-class Classifier Problems. In F. Roli, J. Kittler, and T. Windeatt, editors, *Multiple Classifier Systems, LNCS*, volume 3077, pages 122–133. Springer Verlag, 2004.

[306] E. Pękalska, D.M.J. Tax, and R.P.W. Duin. One-class LP classifier for dissimilarity representations. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 761–768. MIT Press, Cambridge, MA, 2003.

[307] E. Persoon and K.-S. Fu. Shape Discrimination Using Fourier Descriptors. In *2nd International Conference on Pattern Recognition*, pages 126–130, Copehagen, Danmark, 1974.

[308] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University

Press, Cambridge, 1992.

[309] J. Puzicha, T. Hofmann, and J.M. Buhmann. Non-parametric similarity measures for unsupervised texture segmentation and image retrieval. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 267–272, San Juan, 1997.

[310] J. Puzicha, T. Hofmann, and J.M. Buhmann. A theory of proximity based clustering: Structure detection by optimization. *Pattern Recognition*, 33(4):617–634, 1999.

[311] J. Puzicha, Y. Rubner, C. Tomasi, and J.M. Buhmann. Empirical evaluation of dissimilarity measures for color and texture. In *IEEE International Conference on Computer Vision*, pages 1165–1173, 1999.

[312] S.G. Pyatkov. *Operator Theory. Nonclassical problems*. VSP, Utecht, Boston, Köln, Tokyo, 2002.

[313] V. Ramasubramanian and K.K. Paliwal. Fast nearest-neighbor search algorithms based on approximation-elimination search. *Pattern Recognition*, 33(96):1497–151, 2000.

[314] S. Raudys and R.P.W. Duin. On expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix. *Pattern Recognition Letters*, 19(5-6):385–392, 1998.

[315] D. Ridder, E. Pękalska, and R.P.W. Duin. The economics of classification: Error vs. complexity. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *International Conference on Pattern Recognition*, volume 3, pages 244–247, Quebec City, Canada, 2002.

[316] D. de Ridder and R.P.W. Duin. Sammon's mapping using neural networks: a comparison. *Pattern Recognition Letters*, 18(11-13), 1997.

[317] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.

[318] Robinson's notes. `http://www.math.nwu.edu/~clark/310/2001/metric.pdf`.

[319] V. Roth, J. Laub, J.M. Buhmann, and K.-R. Müller. Going metric: Denoising pairwise data. In *Advances in Neural Information Processing Systems*, pages 841–856. MIT Press, 2003.

[320] J. Rovnyak. Methods of Krein space operator theory. *Toeplitz lectures*, given at Tel Aviv, 1999.

[321] S. Roweis, L. Saul, and G. Hinton. Global coordination of local linear models. *Advances in Neural Information Processing Systems*, 14, 2002.

[322] S.T. Roweis and L.K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290: 2323–2326, 2000.

[323] R.P.W. Duin - personal communication.

[324] Y. Rubner. *Texture Metrics*. PhD thesis, Stanford University, 1999.

[325] Y. Rubner, C. Tomasi, and L.J. Guibas. The earth mover's distance as a metric for image retrieval. Technical Report STAN-CS-TN-98-86, Department of Computer Science, Stanford University, 1998.

[326] Y. Rubner, C. Tomasi, and L.J. Guibas. A metric for distributions with applications to image databases. In *IEEE International Conference on Computer Vision*, pages 59–66, Bombay, India, 1998.

[327] V.A. Sadovnichij. *Theory of Operators*. Consultants Bureau, New York and London, 1991.

[328] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4:406–425, 1987.

[329] J.W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18: 401–409, 1969.

[330] J.S. Sánchez, F. Pla, and F.J. Ferri. Prototype selection for the nearest neighbor rule through proximity graphs. *Pattern Recognition Letters*, 18:507–513, 1997.

[331] J.S. Sánchez, F. Pla, and F.J. Ferri. Improving the k-ncn classification rule through heuristic modifications. *Pattern Recognition Letters*, 19:1165–1170, 1998.

[332] S. Santini and R. Jain. Gabor space and the development of preattentive similarity. In *International Conference on Pattern Recognition*, Vienna, Austria, 1996.

[333] S. Santini and R. Jain. Image databases are not databases with images. In A. Del Bimbo, editor, *International Conference on Image Analysis and Processing*, Florence, Italy, 1997.

[334] S. Santini and R. Jain. Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(9):871–883, 1999.

[335] L.K. Saul. Think globally, fit locally: Unsupervised learning of low-dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[336] J.W. Scannell, C. Blakemore, and M.P. Young. Analysis of connectivity in the cat cerebral cortex. *Journal of Neuroscience*, 15:1463–1483, 1995.

[337] R. Schaback. Native hilbert spaces for radial basis functions i. In M.D. Buhman, D.H. Mache, M. Felten, and Müller M.W., editors, *International Series of Numerical Mathematics*, volume 132, pages 255–282. Birkhäuser-Verlag, 1999.

[338] R. Schaback. A unified theory of radial basis functions (native hilbert spaces for radial basis functions ii),. *Journal of Computational and Applied Mathematics*, 121(1-2):165–177, 2000.

[339] R. Schaback and H. Wendland. Approximation by positive definite kernels. In M.D. Buhman and D.H. Mache, editors, *Advanved Problems in Constructive Approximation, International Series in Numerical Mathematics*,

volume 142, pages 203–221. Birkhäuser-Verlag, 2001.

[340] A. Schenker, M. Last, H. Bunke, and A. Kandel. Comparison of distance measures for graph-based clustering of documents. In *Graph Based Representations in Pattern Recognition, LNCS*, volume 2726, pages 202–213. Springer, 2003.

[341] I.J. Schoenberg. Remarks to maurice fréchet's article 'sur la definition axiomatique... d'une classe d'espace distancies vectoriellement applicable sur l'espace de hilbert. *Annals of Mathematics*, 36(3):724–732, 1935.

[342] I.J. Schoenberg. On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in hilbert space. *Annals of Mathematics*, 38:787–797, 1937.

[343] I.J. Schoenberg. Metric spaces and completely monotone functions. *Annals of Mathematics*, 39:811–841, 1938.

[344] I.J. Schoenberg. Metric spaces and positive definite functions. *Transactions on American Mathematical Society*, 44:522–536, 1938.

[345] B. Schölkopf. *Support vector learning*. PhD thesis, Verlag, Munich, 1997.

[346] B. Schölkopf. The kernel trick for distances. In *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2000.

[347] B. Schölkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In *International Conference on Artificial Neural Networks*, 1996.

[348] B. Schölkopf, S. Mika, C.J. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. Smola. Input space vs. feature space in kernel-based methods. *IEEE Transations on Neural Networks*, 1999.

[349] B. Schölkopf, J.C. Platt, A.J. Smola, and R.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[350] B. Schölkopf, Williamson R.C., A.J. Smola, J. Shawe-Taylor, and J.C. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems*, 2000.

[351] B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, 1999.

[352] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.

[353] B. Schölkopf, A.J. Smola, and K.-R. Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, 1997.

[354] B. Schölkopf, A.J. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5), July 1 1998.

[355] B. Schölkopf, A.J. Smola, R. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

[356] B. Schölkopf, K.K. Sung, C. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik. Comparing support vector machines with gaussian kernels to radial basis function classifiers. *IEEE Transations on Signal Processing*, 1997.

[357] T.B. Sebastian and B.B. Kimia. Curves vs skeletons in object recognition. In *International Conference on Image Processing*, Thessaloniki, Greece, 2001.

[358] T.B. Sebastian and B.B. Kimia. Curves vs skeletons in object recognition. *Signal Processing*, to appear, 2003.

[359] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing shock graphs. In *International Conference on Computer Vision*, pages 755–762, 2001.

[360] T.B. Sebastian, P.N. Klein, and B.B. Kimia. Shock-based indexing into large shape databases. In *European Conference on Computer Vision*, volume 3, pages 731–746, 2002.

[361] T.B. Sebastian, P.N. Klein, and B.B. Kimia. On aligning curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):116–125, 2003.

[362] D. Sharvit, J. Chan, H. Tek, and B.B. Kimia. Symmetry-based indexing of image databases. *Journal of Visual Communication and Image Representation*, 9(4):366–380, 1998.

[363] W. Sierpiński. *General Topology*. University of Toronto Press, Toronto, 1952.

[364] SIGIR, Special Interest Group on Information Retrieval. `http://www.sigir.org/`.

[365] P. Simard, Y. Le Cun, and J. Denker. Efficient pattern recognition using a new transformation distance. In *Advances in Neural Information Processing Systems*, pages 50–58, Canada, 1993.

[366] P. Simard, Y. Le Cun, J. Denker, and B. Victorri. *Transformation Invariance in Pattern Recognition – Tangent Distance and Tangent Propagation*, volume 1524, pages 239–274. Springer, Heidelberg, 1998.

[367] M. Skurichina. *Stabilizing Weak Classifiers*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2001.

[368] M. Skurichina and R.P.W. Duin. Combining different normalizations in lesion diagnostics. In O. Kaynak, E. Alpaydin, E. Oja, and L. Xu, editors, *Artificial Neural Networks and Information Processing, Supplementary Proceedings ICANN/ICONIP*, pages 227–230, Istanbul, Turkey, 2003.

[369] A.J. Smola, T.T. Friess, and B. Schölkopf. Semiparametric support vector and linear programming machines. In M.J. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processings Systems 11*, pages 585–591, Cambridge,MA, 1999. MIT Press.

[370] P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy*. W.H. Freeman, San Francisko, California, 1973.
[371] G. de Soete. Additive tree representations of incomplete dissimilarity data. *Quality and Quantity*, 18:387–393, 1984.
[372] G. de Soete. A least squares algorithm for fitting an ultrametric tree to a dissimilarity matrix. *Pattern Recognition Letters*, 2:133–137, 1984.
[373] G. de Soete. Ultrametric tree representations of incomplete dissimilarity data. *Journal of Classification*, 1: 235–242, 1984.
[374] G. de Soete and J.D. Caroll. *Clustering and Classification*, chapter Tree and other Network Models for Representing Dissimilarity Data, pages 157–198. London: World Scientific, 1996.
[375] B.M.R. Stadler and P.F. Stadler. Basic properties of filter convergence spaces. Technical report, Institute for Theoretical Chemistry and Structural Biology, University of Vienna, Austria, 2001.
[376] B.M.R. Stadler and P.F. Stadler. Higher separation axioms in generalized closure spaces. *Annales Societatis Mathematicae Polonae. Series I. Commentationes Mathematicae*, submitted, 2001.
[377] B.M.R. Stadler and P.F. Stadler. Basic properties of closure spaces. Technical report, Institute for Theoretical Chemistry and Structural Biology, University of Vienna, Austria, 2002.
[378] B.M.R. Stadler, P.F. Stadler, M. Shpak, and G.P. Wagner. Recombination spaces, metrics, and pretopologies. *Zeitschrift für Physikalische Chemie*, 216:217–234, 2002.
[379] B.M.R. Stadler, P.F. Stadler, G.P. Wagner, and W. Fontana. The topology of the possible: Formal spaces underlying patterns of evolutionary change. *Journal of Theoretical Biology*, 213(2):241–274, 2001.
[380] G.A. Stephen. *String Searching Algorithms*. World Scientific Publishing Company, 2nd edition, 1998.
[381] R. Strauss. Matlab routines. `http://www.biol.ttu.edu/Strauss/Matlab/matlab.htm`.
[382] A. Strehl. *Relationship-based Clustering and Cluster Ensembles for High-dimensional Data Mining*. PhD thesis, University of Texas at Austin, USA, 2002.
[383] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.
[384] A. Strehl and J. Ghosh. Cluster ensembles - a knowledge reuse framework for combining partitionings. In *Conference on Artificial Intelligence*, pages 93–98, Edmonton, 2002.
[385] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search*, pages 58–64, Austin, Texas, USA, 2000. AAAI.
[386] D.M.J. Tax. *One-class classification*. PhD thesis, Delft University of Technology, The Netherlands, 2001.
[387] D.M.J. Tax. DD-Tools, a Matlab toolbox for data description, outlier and novelty detection, 2003.
[388] D.M.J. Tax. A consistency-based model selection for one-class classifiers. In *International Conference on Pattern Recognition*, volume 2, page BLA, Cambridge, UK, 2004.
[389] D.M.J. Tax and R.P.W. Duin. Combining one-class classifiers. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems, LNCS*, volume 2096, pages 299–308. Springer Verlag, 2001.
[390] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine Learning*, 54(1):45–56, 2004.
[391] D.M.J. Tax, R.P.W. Duin, and J. Kittler. Combining multiple classifiers by averaging or by multiplying? *Pattern Recognition*, 33(9):1475–1485, 2000.
[392] Y. Teh and S. Roweis. Automatic alignment of local representations. *Advances in Neural Information Processing Systems*, 15, 2003.
[393] J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
[394] Texture data. `ftp://whitechapel.media.mit.edu/pub/VisTex/`.
[395] A. Thayananthan, B. Stenger, P.H.S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–133, Wisconsin, 2003.
[396] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.
[397] A. Topchy, B. Minaei, A.K. Jain, and W. Punch. Adaptive clustering ensembles. In R. Kasturi, D. Laurendeau, and C. Suen, editors, *International Conference on Pattern Recognition*, Cambridge,United Kingdom, 2004.
[398] W.S. Torgerson. *Theory and Methods of Scaling*. John Wiley & Sons, 1967.
[399] A. Torsello and E.R. Hancock. Computing approximate tree edit distance using relaxation labeling. *Pattern Recognition Letters*, 24(8):1089–1097, 2003.
[400] M.W. Trosset and R. Mathar. Optimal dilations for metric multidimensional scaling. In *Statistical Computing Section, American Statistical Association*, 2000.
[401] A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.
[402] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.
[403] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.
[404] N. Vasconcelos and M. Kunt. Content-based retrieval from image databases: Current solutions and future direc-

tions. In *International Conference on Image Processing*, Thessaloniki, Greece, 2000.

[405] N. Vasconcelos and A. Lippman. A unifying view of image similarity. In *International Conference on Pattern Recognition*, Barcelona, Spain, 2000.

[406] D.C.G. de Veld, M. Skurichina, M.J.H. Witjes, and et.al. Autofluorescence characteristics of healthy oral mucosa at different anatomical sites. *Lasers in Surgery and Medicine*, 23:367–376, 2003.

[407] R. Veltkamp. Shape matching: Similarity measures and algorithms. Technical Report UU-CS-2001-03, Utrecht University, the Netherlands, 2001.

[408] R. Veltkamp and M. Hagedoorn. State-of-the-art in shape matching. Technical Report UU-CS-1999-27, Utrecht University, the Netherlands, 1999.

[409] R. Verma. A metric approach to isolated word recognition. Master's thesis, Department of Computer Science, University of Toronto, 1991.

[410] E. Vidal, A. Marzal, and Aibar P. Fast computation of normalized edit distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(9):899 – 902, 1995.

[411] R.A. Wagner and M.J. Fisher. The string-to-string correction problem. *Journal of the Association for Computing Machinery*, 21(1):168–173, 1974.

[412] G. Wahba. Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*, pages 69–88. MIT Press, Cambridge, MA, 1999.

[413] S. Watanabe. *Pattern Recognition, Human and Mechanical*. Academic Press, New York, 1974.

[414] A.R. Webb. Multidimensional scaling by iterative majorization using radial basis functions. *Pattern Recognition*, 28(5):753–759, 1995.

[415] A.R. Webb. Radial basis functions for exploratory data analysis: An iterative majorisation approach for Minkowski distances based on multidimensional scaling. *Journal of Classification*, 14(2):249–268, 1997.

[416] Webster dictionary. `http://w-m.com`.

[417] M. Werman and D. Weinshall. Similarity and affine invariant distance between 2D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8), December 1995.

[418] C.M. Wharton, K.J. Holyoak, Downing, P.E., T.E. Lange, and T.D. Wickens. The story with reminding: Memory retrieval is influenced by analogical similarity. In *Annual Conference of the Cognitive Science Society*, pages 588–593, Blomington, 1992.

[419] S. Willard. *General Topology*. Addison-Wesley Publishing Company, 1970.

[420] C.L. Wilson and M.D. Garris. Handprinted character database 3. Technical report, National Institute of Standards and Technology, February 1992.

[421] D.R. Wilson and T.R. Martinez. Improved heterogeneous distance functions. *Journal of Artificial Intelligence Research*, 6:1–34, 1997.

[422] R.W. Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

[423] L. Younes. Computable elastic distances between shapes. *SIAM Journal on Applied Mathematics*, 58(2):565–586, 1998.

[424] L. Younes. Optimal matching between shapes via elastic deformations. *Image and Vision Computing*, 17(5-6): 381–389, 1999.

[425] G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.

[426] A. Ypma and R.P.W. Duin. Support objects for domain approximation. In *International Conference on Artificial Neural Networks*, pages 719–724, Skovde, Sweden, 1998. Springer, Berlin.

[427] A. Ypma, R. Ligteringen, E.E.E. Frietman, and R.P.W. Duin. Recognition of bearing failures using wavelets and neural networks. In *British Symposium on Applications of Time-Frequency and Time-Scale Methods*, pages 69–72, University of Warwick, Coventry, UK, 1997.

[428] C.T. Zahn and R.Z. Roskies. Fourier Descriptors for Plane Closed Curves. *IEEE Transactions*, C-21(3):269–281, 1972.

[429] S. Zhu and A. Yuille. Forms: A flexible object recognition and modeling system. *International Journal on Computer Vision*, 20(3):187–212, 1996.

Summary

I woke up and the world outside was dark
all so quiet before the dawn
opened up the door and walked outside
the ground was cold
I walked until I couldn't walk anymore
to a place I'd never been
there was something stirring in the air
in front of me, I could see
More than this

"MORE THAN THIS", PETER GABRIEL

In 2004, kernel methods [74, 352] have become popular in statistical learning. Kernels are (conditionally) positive definite (cpd) functions of two variables, which serve to encode similarities between pairs of objects. Such objects are usually represented in a feature space. In 1995, Vapnik [403] proposed an elegant formulation of the largest margin classifier. This support vector machine (SVM) was based on the reproducing property of kernels. Since then, many variants of the SVM have been applied to a wide range of learning problems.

It was recognised before the start of our project [106, 108, 109] in 1999, that the class of cpd functions is restricted. It does not accommodate a number of useful proximity measures already developed in pattern recognition and computer vision. Many existing similarity measures are not positive definite and many existing dissimilarity measures are not Euclidean[1] or even not metric. Examples are pairwise structural alignments of proteins, variants of the Hausdorff distance and normalized edit-distances. The major limitation in using such kernels is that the original formulation of the SVM relies on a quadratic optimization. This problem is guaranteed to be convex for cpd kernels, and therefore uniquely solvable by standard algorithms. Kernel matrices disobeying these requirements are usually regularized by adding a suitable constant to their diagonal.

This thesis extends the notion of a kernel to that of a proximity representation, since proximity underpins the description of a class as a group of similar objects. In such a representation, each object is described by a set of proximities to the so-called representation set R [301]. If R is chosen to be the set of training examples, then this proximity representation becomes a generalized kernel. When a suitable similarity measure is selected, a cpd kernel is obtained as a special case. Using a proximity representation, learning can be addressed in a more general way than is possible using the SVM. To focus on class and object differences, in this work proximity is modeled as dissimilarity rather than similarity. This is, however, not essential.

The main goal of this thesis is to provide a mathematical foundation and to develop learning methodologies for dissimilarity representations. The thesis is divided into a theoretical part (chapters 2–5) and an experimental part verifying the proposed methodologies (chapters 6–10).

Chapter 2 briefly describes various spaces such as (pre)topological, normed, metric and inner product spaces, as well as their interrelations. Some attention is devoted to Kreĭn spaces as indefinite inner product spaces. These are later discussed as pseudo-Euclidean spaces of finite embeddings

[1] The dissimilarity measure being Euclidean is inherently related to the corresponding kernel being positive definite, as explained in chapter 3.

of dissimilarities. The introduction of such spaces prepares a mathematical framework for handling arbitrary dissimilarity data and the design of learning procedures later on.

Chapter 3 deals with the characterization of $n \times n$ dissimilarity matrices with respect to Euclidean, l_1 and metric properties and some related transformations. It introduces useful tools for checking metric or Euclidean behavior. In particular, the issue of (approximate) linear and nonlinear embedding into Euclidean spaces, as well as into pseudo-Euclidean spaces, is discussed. This relies on finding spatial vector representations such that dissimilarities are preserved, which is possible for any symmetric dissimilarity measure. This lays the foundation for designing learning algorithms on spatial representations, as described in chapter 4. Next, the city block distance is characterized by its additivity property. This distance can be perfectly structured by an additive tree model, where the distance is realized in terms of the shortest path in the tree. Other dissimilarity measures can also be interpreted via such tree models, though only approximately. Such models of dissimilarity can help in understanding the organization of objects.

Chapter 4 starts with a brief introduction to feature-based statistical learning. Then, a more detailed description of dissimilarity representations is given. The (relative) dissimilarity representation is described by a dissimilarity matrix $D(T, R)$between the set of (training) objects T and the representation set of prototype objects, R. This chapter further focuses on possible decision functions for such representations. The three main learning approaches rely on the interpretation of dissimilarities in different spaces:

1. pretopological spaces, in which dissimilarity values are used directly to describe neighborhood relations between objects;
2. embedded spaces, usually pseudo-Euclidean, which are vector spaces determined by dissimilarity-preserving projections;
3. dissimilarity spaces, where each dimension corresponds to the dissimilarity to a particular representation object.

These methods can use any nonnegative dissimilarity measure satisfying the reflexivity condition (and, additionally, the symmetry condition if the embedded space approach is considered), provided that it is meaningful for the learning problem. Various non-Euclidean or non-metric statistical or structural dissimilarity measures already known can be used, or even designed to respond better to practical requirements. For instance, in computer vision, it is known that in the presence of partially occluded objects, non-metric measures are preferred for template matching purposes [206].

In chapter 5, various similarity and dissimilarity measures are described, together with their basic properties. Also, a brief overview of measures used in practical applications is presented.

Chapters 6 and 7 discuss fundamental questions related to exploratory data analysis, i.e. the understanding of relations within data. Chapter 6 investigates a number of well-known visualization techniques and their use on dissimilarity data. Multidimensional scaling and Isomap seem to reveal most of the data structure. Additive tree models are useful for understanding hierarchical or nested structures in dissimilarity data. However, their interpretation is limited to a moderate number of objects.

Chapter 7 investigates structure and complexity in dissimilarity representations. Clustering methods in dissimilarity spaces may be useful for problems in which at least one of the clusters is compact and others are more wide-spread. The intrinsic dimensionality can be estimated in embedded or dissimilarity spaces. The number of significant eigenvalues in a linear pseudo-Euclidean embedding, as well as in a principal component analysis applied in a dissimilarity space, give a reasonable indication. Additionally, some statistics are proposed and experimentally examined which may be used to quantify whether a representation set contains a sufficient number of objects to describe

class variability.

A possible approach to outlier detection is analyzed in chapter 8 by constructing a one-class classifier (OCC). Currently existing OCCs are built either in traditional feature spaces or on Euclidean distances derived there. Two new OCCs, one in embedded space and one in dissimilarity space, the so-called linear programming data description (LPDD), are proposed and applied to the problems of machine monitoring, lesion diagnostics and heart disease diagnostics. When the outliers do not heavily overlap with target objects, the LPDD may provide the best solution as a trade-off between performance and computational complexity. Noteworthy is the fact that the best measures in the considered noisy problems of machine monitoring and lesion diagnostics are non-metric distances.

Chapter 9 discusses classification. Various dissimilarity measures have been analyzed for this purpose. Experiments show that linear or quadratic classifiers, constructed in either dissimilarity spaces or embedded spaces, often significantly outperform the k-NN rule for small representation sets, irrespective of the measure used. Additionally, some transformations have been applied to non-Euclidean dissimilarity measures to make them (more) Euclidean (as discussed in chapter 3). This imposed Euclidean behavior is not found to result in better classification performance. It is more important that the measure itself describes separated and possibly compact classes than that it is strictly Euclidean or metric.

Various methods for representation set selection have been studied, for both the embedding and dissimilarity space approaches. When small representation sets are sought, systematic procedures work best, e.g. optimizing the classification error. For moderate representation set sizes, the k-centers algorithm is fast and works well on average. It allows one to control the number of selected prototypes, and hence the complexity of classifiers to be further constructed. In the dissimilarity space approach, support objects selected by a sparse linear programing machine form good representation sets as well, although their size cannot be influenced (except when k-centres is used beforehand). In the embedding approach, the representation set can consist of objects resulting in the largest approximation error. Finally, when the representation set should be large, random selection can be used. A randomly selected representation set consisting of 20% (or more) of the training objects works well in both dissimilarity and embedded spaces.

Some ideas on zero-error recognition are also discussed. Under some constraints (on unambiguous labeling of objects and on properties of the dissimilarity measure) the k-NN rule will work perfectly for very large training sets. As this may be infeasible, an alternative is the use of linear classifiers in dissimilarity spaces. For these, a small representation set may suffice. However, a zero-error solution cannot always be found for the test set. This depends on the dissimilarity measure and the size of the representation set in relation to the classifier chosen.

In chapter 10, it is discussed how combining different sources of information or different learning strategies may be effective for designing a good pattern recognition system. Combining is a natural way of merging statistical and structural dissimilarity representations. Methods for combining dissimilarity representations are proposed and experimentally investigated. Classifiers built on such combined representations outperform the best classifier (of the same type) constructed on single representations. Classifiers combined by fixed rules also work well. The product rule combiner seems to be especially useful for small representation sets in two-class problems, while majority voting can be applied for one class classifiers. Additionally, a way of representing a group of classifiers is proposed, in a projection space based on (approximate) embedding of pairwise diversities between classifiers. Studying classifier differences in this way may increase understanding of the recognition problem at hand.

Our study on dissimilarity representations applies to all dissimilarities, independently of the way they have been derived, e.g. from raw data or from an initial representation by features, strings or

graphs. Expert knowledge on the application can be used to formulate this initial representation and in the definition of the proximity measure. This makes the dissimilarity representations developed natural candidates for combining the strengths of structural and statistical approaches in pattern recognition and machine learning. The advantage of the structural approach lies in encoding both domain knowledge and the structure of an object. The benefit of the statistical approach lies in a well-developed mathematical theory of vector spaces. First, a description of objects in the structural framework can be found. This can then be quantized to capture the dissimilarity relations between the objects. If necessary, other structurally and statistically derived measures can be designed and combined. The final dissimilarity representation is then used in statistical learning. The results in this thesis justify the use and further exploration of dissimilarity information for pattern recognition.

Samenvatting

Perfection is achieved, not when there is nothing more to add, but when there is nothing left to take away.

ANTOINE DE SAINT-EXUPÉRY

Sinds 2004 zijn *kernel methoden* [74, 352] populair geworden in het statistisch leren. Kernels zijn (conditioneel) positief definiete (cpd) functies van twee variabelen, die gebruikt worden om overeenkomsten tussen paren van objecten te coderen. Meestal worden objecten gerepresenteerd in een kenmerkruimte. In 1995 heeft Vapnik [403] een elegante formulering voor de *largest margin classifier* voorgesteld. Deze *support vector machine* (SVM) is gebaseerd op de reproductie-eigenschap van kernels. Sindsdien worden veel varianten van de SVM toegepast op een breed scala aan leerproblemen.

Al voor de start van ons project [106, 108, 109] in 1999 werd ingezien dat de klasse van cpd functies beperkt is. Zij mist een aantal bruikbare nabijheidsmaten die al ontwikkeld waren in de patroonherkenning en *computer vision*. Veel bestaande overeenkomstmaten zijn niet positief definiet en veel bestaande ongelijkheidsmaten zijn niet Euclidisch[2] of zelfs niet metrisch. Voorbeelden zijn paarsgewijze structurele oplijning van eiwitten, varianten van de Hausdorff afstand en genormaliseerde bewerkingsafstanden. De voornaamste beperking in het gebruik van dergelijke kernels is dat de originele formulering van de SVM een kwadratische optimalisatie vereist. Dit probleem is gegarandeerd convex voor cpd kernels en daarmee uniek oplosbaar met standaard algorithmen. Kernel matrices die niet aan deze voorwaarden voldoen worden normaliter geregulariseerd door er een toepasselijke constante bij hun diagonaal op te tellen.

Dit proefschrift breidt de notie van een kernel uit naar die van een nabijheidsrepresentatie, aangezien nabijheid de beschrijving van een klasse als groep van gelijkende objecten ondersteund. In zo'n representatie wordt elk object beschreven door zijn afstanden tot de zogenaamde representatieset R [301]. Indien R wordt gekozen als de verzameling leervoorbeelden dan wordt de nabijheidsrepresentatie een gegeneraliseerde kernel. Als er een toepasselijke overeenkomstmaat is gekozen, wordt een cpd kernel verkregen als speciaal geval. Gebruikmakend van een nabijheidsrepresentatie kan het leren op een algemenere wijze worden benaderd dan mogelijk is met gebruik van een SVM. Om nadruk te leggen op klasse- en objectverschillen wordt in dit werk nabijheid gemodelleerd als verschil, in plaats van overeenkomst. Dit is echter niet essentieel.

Het hoofddoel van dit proefschrift is het leggen van een wiskundige onderbouwing van, en het ontwikkelen van leermethodologieën voor, verschilrepresentaties. Dit proefschrift is ingedeeld in een theoretisch deel (hoofstukken 2-5) en een experimenteel deel waarin de voorgestelde methodologieën worden geverifieerd (hoofdstukken 6-10).

Hoofdstuk 2 beschrijft kort een aantal ruimten, zoals (pre)topologische, genormeerde, metrische en inwendig produkt-ruimten, alsook hun onderlinge verhoudingen. Er wordt enige aandacht besteed aan Kreĭn ruimten als indefiniete inwendig-produkt ruimten. Deze worden later beschreven als pseudo-Euclidische ruimten van eindige inbeddingen van ongelijkendheden. De introductie van dergelijke ruimten is een voorbereiding op een wiskundig raamwerk om met willekeurige verschil gegevens om te kunnen gaan en het latere ontwerp van leerprocedures.

[2] Het Euclidisch zijn van een verschilmaat is inherent aan het positief definiet zijn van de overeenkomstige kernel, zoals wordt uitgelegd in hoofdstuk 3.

Hoofdstuk 3 karakteriseert $n \times n$ verschil matrices voor wat betreft Euclidische, l_1 en metrische eigenschappen en enkele gerelateerde transformaties. Het introduceert bruikbare gereedschappen om metrisch of Euclidisch gedrag te controleren. In het bijzonder wordt het geval van (bij benadering) lineaire en niet-lineaire inbedding in Euclidische ruimten, alsook in pseudo-Euclidische ruimten, besproken. Hiervoor dienen spatiële vector-representaties te worden gevonden zodanig dat de ongelijkendheden bewaard worden, hetgeen mogelijk is voor elke symmmetrische verschilmaat. Dit legt de basis voor het ontwerpen van leeralgorithmen voor spatiële representaties, zoals beschreven in hoofdstuk 4. Vervolgens wordt de *city block*-afstand gekarakteriseerd door haar additiviteits-eigenschap. Deze afstand kan perfect worden weergegeven met een additief boommodel, waarin afstand wordt gevonden als het kortste pad in de boom. Ook andere verschilmaten kunnen met dergelijke boommodellen worden geïnterpreteerd, doch alleen bij benadering. Zulke verschilmodellen kunnen helpen in het begrijpen van de organisatie van objecten.

Hoofdstuk 4 begint met een korte introductie over statistisch leren op basis van kenmerken. Vervolgens wordt een gedetailleerdere beschrijving van verschilrepresentaties gegeven. De (relatieve) verschilrepresentatie wordt beschreven door een verschilmatrix $D(T, R)$ tussen de verzameling (leer-) objecten T, en de representatieverzameling met prototype objecten R. Verder richt het hoofdstuk zich op mogelijke beslissingsfuncties voor zulke representaties. De drie belangrijkste leerbenaderingen hangen af hoe de verschillen in de diverse ruimten worden geïnterpreteerd:

1. pretopologische ruimten, waarin verschilwaarden rechtstreeks worden gebruikt om buurrelaties tussen objecten te beschrijven;
2. ingebedde ruimten, meestal pseudo-Euclidisch, ruimten, dit zijn vectorruimten die gevonden zijn met verschil-behoudende projecties;
3. verschilruimten, waarin elke dimensie overeenkomt met het verschil met een specifiek representatie-object.

Deze methoden kunnen gebruik maken van willekeurige niet-negatieve verschilmaten die voldoen aan de reflexiviteitsvoorwaarde (en de symmetrievoorwaarde als de ingebedde ruimte-benadering wordt gevolgd), voor zover zij van betekenis zijn voor het leerprobleem. Verschillende bekende niet-Euclidische of niet-metrische statistische of structurele verschilmaten kunnen worden gebruikt, of zelfs worden geconstrueerd om beter aan praktische eisen te kunnen voldoen. In *computer vision*, bijvoorbeeld, is het bekend dat voor *template matching* niet-metrische maten de voorkeur verdienen als gedeeltelijk bedekte objecten aanwezig zijn [206].

In hoofdstuk 5 worden diverse overeenkomst- en verschilmaten, met hun basale eigenschappen beschreven. Daarnaast wordt een kort overzicht gegeven van maten die in praktische toepassingen gebruikt worden.

Hoofdstuk 6 en 7 behandelen fundamentele vragen op het gebied van exploratieve data analyse, zoals het begrip van relaties in data. Hoofdstuk 6 onderzoekt een aantal bekende visualisatietechnieken en het gebruik daarvan op verschildata. Meerdimensionale schaling en Isomap lijken het meest te onthullen over de structuur in de data. Additieve boommodellen zijn nuttig voor het begrijpen van hiërarchische of elkaar omvattende structuren in verschildata. De resultaten van deze methoden kunnen echter slechts goed worden geïnterpreteerd indien het aantal objecten beperkt is.

Hoofdstuk 7 onderzoekt structuur en complexiteit in verschilrepresentaties. Clustermethoden in verschilruimten kunnen nuttig zijn voor problemen waarin tenminste één van de clusters compact is en de andere meer verspreid. De intrinsieke dimensionaliteit kan worden geschat in ingebedde ruimten of verschilruimten. Het aantal significante eigenwaarden in een lineaire pseudo-Euclidische inbedding, of in een principale componenten analyse toegepast in verschilruimten, geven een redelijke indicatie. Daarnaast worden enkele maten voorgesteld en experimenteel onderzocht, welke gebruikt kunnen worden om te kwantificeren of een representatieverzameling een voldoende groot

aantal objecten bevat om klassevariabiliteit te beschrijven.

Een mogelijke benadering van uitbijterdetectie wordt geanalyseerd in hoofdstuk 8, door het ontwerpen van een één-klasse klassificator, of *one-class classifier* (OCC). Bestaande OCCs worden gebouwd in ofwel traditionele kenmerkruimten, ofwel op Euclidische afstanden gevonden in die ruimten. Twee nieuwe OCCs, één in de ingebedde ruimte en één in de verschilruimte, de zogenaamde *linear programming data description* (LPDD), worden voorgesteld en toegepast op problemen in machinebewaking, diagnostiek van verwondingen en diagnostiek van hartafwijkingen. Als de uitbijters niet zwaar overlappen met de doelobjecten kan de LPDD de beste afweging tussen prestatie en rekencomplexiteit opleveren. Opmerkelijk is dat voor de ruizige problemen van machinebewaking en diagnostiek van verwondingen, de beste maten niet-metrische afstanden zijn.

Hoofdstuk 9 behandelt klassificatie. Diverse verschilmaten zijn hiervoor geanalyseerd. Experimenten laten zien dat voor kleine representatieverzamelingen lineaire of kwadratische klassificatoren, opgebouwd in verschilruimten danwel ingebedde ruimten, vaak significant beter presteren dan de k-NN regel, ongeacht de gebruikte maat. Daarnaast worden sommige transformaties toegepast op niet-Euclidische verschilmaten om ze (meer) Euclidisch te maken (zoals beschreven in hoofdstuk 3). Dit opgelegde Euclidische gedrag leidt niet tot betere klassificatieprestaties. Het is belangrijker dat de maat gescheiden en mogelijk compacte klassen goed beschrijft, dan dat zij strikt Euclidisch of metrisch is.

Verscheidene methoden voor het selecteren van een representatieverzameling zijn bestudeerd, voor zowel ingebedde ruimten als voor verschilruimten. Als er kleine representatieverzamelingen worden gezocht, werken systematische procedures, die bijvoorbeeld de klassificatiefout optimaliseren, het best. Voor iets grotere afmetingen van de representatieverzameling werkt het k*-centers* algorithme snel en presteert gemiddeld goed. Het stelt de gebruiker in staat het aantal geselecteerde prototypen te beïnvloeden, en daarmee de complexiteit van de klassificatoren die verder geconstrueerd worden. In de verschilruimte-aanpak vormen *support* objecten geselecteerd met behulp van een *sparse linear programming machine* ook goede representatieverzamelingen, hoewel hun grootte niet regelbaar is (tenzij de k*-centres* methode tevoren wordt gebruikt). In de ingebedde ruimte-aanpak kan de representatieverzameling bestaan uit die objecten die de hoogste benaderingsfout geven. Tenslotte, wanneer de representatieverzameling groot moet zijn, kan willekeurige selectie gebruikt worden. Een willekeurig geselecteerde representatieverzameling bestaand uit 20% (of meer) van de leerobjecten werkt goed in verschilruimten en in ingebedde ruimten.

Bovendien worden enkele ideeën over foutloze herkenning besproken. Onder een paar beperkende aannamen (over niet-ambigue *labeling* van objecten en over eigenschappen van de verschilmaat) werkt de k-NN regel perfect voor zeer grote leerverzamelingen. Aangezien dit onhaalbaar kan zijn, kunnen als alternatief lineaire klassificatoren in verschilruimten gebruikt worden. Hiervoor kan een kleine representatieverzameling voldoende zijn. Een foutloze oplossing kan echter niet altijd worden gevonden voor de testverzameling. Dit hangt af van de verschilmaat en de grootte van de representatieverzameling in relatie tot de gekozen klassificator.

In hoofdstuk 10 wordt behandeld hoe het combineren van verschillende informatiebronnen of verschillende leerstrategieën doeltreffend kan zijn voor het ontwerpen van een goed patroonherkennend systeem. Combineren is een natuurlijke methode om statistische en structurele verschilrepresentaties te verenigen. Methoden voor het combineren van verschilrepresentaties worden voorgesteld en experimenteel onderzocht. Klassificatoren geconstrueerd met dergelijke gecombineerde representaties presteren beter dan de beste klassificator (van hetzelfde type) gebouwd op enkelvoudige representaties. Klassificatoren gecombineerd met vaste regels werken ook goed. De produktregel lijkt bij uitstek van nut voor kleine representatieverzamelingen in twee-klasse-problemen, terwijl de meerderheidsregel kan worden toegepast op *one-class classifiers*. Daarnaast wordt een

manier voorgesteld om een groep klassificatoren te representeren in een projectieruimte, gebaseerd op een (benaderde) inbedding van paarsgewijze diversiteiten tussen klassificatoren. Het op deze manier bestuderen van verschillen tussen klassificatoren kan het begrip van het beschouwde herkenningsprobleem vergroten.

Onze studie naar verschilrepresentaties is van toepassing op alle verschilmaten, onafhankelijk van de manier waarop zij afgeleid zijn, bijvoorbeeld van ruwe data of van een oorspronkelijke representatie als kenmerken, *strings* of grafen. Expertkennis over de toepassing kan worden gebruikt om deze oorspronkelijke representatie te formuleren en een nabijheidsmaat te definiëren. Dit maakt de ontwikkelde verschilrepresentaties natuurlijke kandidaten voor het combineren van de sterke kanten van structurele en statistische benaderingen in de patroonherkenning en het machineleren. Het voordeel van de structurele benadering ligt in het coderen van zowel domeinkennis als de structuur van een object. Het voordeel van de statistische aanpak is de goed ontwikkelde wiskundige theorie van vectorruimten. Allereerst kan een beschrijving van objecten in het structurele raamwerk worden gevonden. Deze kan vervolgens worden gekwantificeerd om de verschilrelaties tussen objecten weer te geven. Indien nodig kunnen andere maten worden ontworpen en gecombineerd, die zijn afgeleid uit structurele of statistische aanpak. De uiteindelijke verschilrepresentatie kan dan gebruikt worden in statistisch leren. De resultaten in dit proefschrift rechtvaardigen het gebruik en de verdere onderzoek van verschilinformatie in de patroonherkenning.

Acknowledgments

Gratitude is the memory of the heart.
JEAN BAPTISTE MASSIEU

My scientific development cannot be separated from my personal development, since they are deeply connected. Many people, directly or indirectly, contributed to the accomplishment of this thesis in their own unique ways. My understanding of the various spaces builds upon the knowledge I received in my student years at University of Wrocław. *Prof. Stanisław Lewanowicz* was the person who practically introduced to me the concept of orthogonality and norms in a broad sense. The interest in statistical data analysis and in neural networks, in particular, arose thanks to *prof. Anna Bartkowiak*. My first steps in this direction were made with her. This experience influenced my further choices and I am grateful for that.

My PhD research was done in the Pattern Recognition group (renamed to Quantitative Imaging in 2004). There were many people in the group who contributed to my development, but more importantly, it was the **entire PR group** itself which nurtured my growth and infused me with the appreciation of scientific thinking. I am grateful to the staff members, supporting staff, postdocs, PhD students and master students. These are memorable years. Thank you all!

My special thanks goes to:

- *Prof. Ted Young* for being the group leader and my supervisor. I took my time to write the thesis and it was possible also because you offered me a quiet working place, when it was needed. Your critical remarks on my work made me more careful in presentation of the research line and the results.
- *Bob Duin* for being a great mentor. I remember your explanation about stars and support vector machines, which really amazed me. I could not always follow your imagination, but I was intrigued to face the questions and I felt inspired to look for answers. I appreciate all the discussions, scientific (dis)agreements, enlightenments and disappointments that we had. In good and bad times, your encouragement was there. I have learned from your exceptional insights and I have developed through your guidance. Thank you.
- *Prof. Lucas van Vliet*, *Pieter Jonker*, *Piet Verbeek* and *Albert Vossepoel* for creating a good scientific atmosphere in the group. Thanks for your sense of humor and spontaneous and vivid discussions!

I am grateful to the people of the Neuro team for being the **team**, indeed. These are: Bob Duin, Marina Skurichina, Alexander Ypma, Dick de Ridder, David Tax, Pavel Paclík, Piotr Juszczak, Carmen Lai, Serguei Verzakov, Thomas Landgrebe and Artsiom Harol. We had many scientific and philosophical discussions, participated in conferences together, shared tasks and responsibilities, prepared and taught courses, supported each other and had fun. We created something unique through the diversity of our personalities. My appreciation to you all!

I thank *David Tax* and *Dick de Ridder* for proofreading and commenting on parts of the thesis manuscript, as well as for the required Dutch translations. I also acknowledge *prof. Jan Aarts* for his profound reading and improvements of my mathematical formalism and patience in explaining various topological issues.

I want to thank some people for specific things, as they are examples for me. These are (alphabetically):

- *Judith Dijk* for being a person who acts and not only listens.
- *Richard van den Doel* for encouragement, support and care. Thanks for dutch poems and dutch songs and for learning from each other. I was really amazed by your love for children.
- *Inge Duin* for the joy and for sharing your shrewd observations with me. Thanks to you I understood that antipathetic abilities, handled with consciousness, are important in life.

- *Michael van Ginkel* for the substantial help in my first years. Thanks for your care and patience in explaining things, which made my living in the Netherlands easier. You are the best to tell a story about Fourier! Thanks for your friendship.
- *Piotr Juszczak* for showing me how creativity can be born from chaos. Your direct observations confronted me with who I am. It was difficult to face, but I have learned a lot. I am ready to fly.
- *Gerold Kraft* for appealing remarks and observations.
- *Carmen Lai* for intriguing observations. Thanks for showing me beautiful Sardinia.
- *Ronald Ligteringen* and *Wouter Smaal* for patience in handling my computer complaints.
- *Cris Luengo Hendriks* for a gentleman-like way of doing things. Thanks for support.
- *Wim van Oel* for serving others with small things. You have a big heart!
- *Pavel Paclík* for help, support and enthusiasm towards the understanding of dissimilarities. You are a person who has a power to change himself from inside. Thanks for your solid thinking and openness.
- *Barbara de Ridder* for being a person with an unusual clarity of mind.
- *Dick de Ridder* for assistance and help with practical problems. You are a person who plans carefully and acts accordingly. Your words have a precise meaning and your honesty is a true quality. I appreciate it a lot.
- *Bernd Rieger* for direct questions and sharp observations.
- *Klara Schaafsma* for spontaneous and kind assistance in little, but important administrative issues.
- *Marina Skurichina* for cheerfulness, positive thinking, support and all kinds of practical advices.
- *David Tax* for explanations and scientific discussions that helped me to make steps in my research. You are an enthusiastic teacher and a master of 'cutting' things. Thanks for the joy in free thinking. Thanks for your care and your friendship.
- *Serguei Verzakov* for the care and for showing me how physics meets math.
- *Alexander Ypma* for the care, constructive criticism and practical help in my first years.
- *John Zevenbergen* for a listening ear, jolly presence, support and help in many practical problems.

On the way to develop the researcher in me, I have also met *prof. Ana Fred*, *Tin Kam Ho* and *Lucy Kuncheva*. I wish to acknowledge their prodigious professionalism, which is an example for me.

The final work on the thesis was done in the Information and Communication Theory (ICT) group. There is an open and friendly atmosphere in the ICT group that supports scientific development. I am grateful for that to all! I especially thank *prof. Inald Lagendijk* for the friendly welcome in the group. *Anja van den Berg* is acknowledged for the assistance in practical issues. My appreciation goes also to the system administrators *Ben van den Boom*, *Hans Verschuur* and *Robbert Eggermont* for the help I needed.

Alicja and Christiaan Baljé, *Jola and Jacek Offierscy*, *Danka and Mirek Drabarek*, *Jola and Jurek Duszczyk*, *Renia and Irek Karkowscy* made the first years in Delft more cheerful. Thanks a lot.

I have grown through common experiences and activieties as well as good discussions with many. My special thanks goes to *Krzysiek Stawiarski* for being a person with passion, *Ania and Jacek Wojdeł* for an example of organizational mastership, *Mirka and Leszek Góra* for a remarkable sense of humor, *Ania Tworowska and Wojtek Stec* for craziness and jolly presence, *Elwira and Adrian Bohdanowicz* as well as *Ewelina and Michał Sobera* for straightforward, reliable and open relations, *Agnieszka and Radek Gnutek*, *Michal Glazer*, *Iwona and Hans Kardol* and *GianLuca di Nola*. Thanks for your daring spirit!

My exercises in independent thinking started many years ago in Wrocław. Some of these thinking abilities were developed thanks to many discussions with *Natasza Sprutta*, *Paweł Furman* and *Jan Lewanowicz* at that time. I am grateful for an listening ear and possible continuations when we meet. I express also my appreciation to *Christie Murphy* for an open heart and for friendship.

I also wish to thank *Zofia and Józef Pękalscy* for showing me practical, intellectual and moral approaches to life and *Anna Pękalska* for an example of inspiring courage.

I express my full gratitude to my parents, *Irena and Roman Górniewicz* who supported my love for math and made me realize that education is an important gift that I need to respect. Dziękuję za wszystko.

Finally, I thank my husband *Andrzej* for a half-life of friendship, patience, support and love. You are a remarkable person and a researcher with an exceptional engineering thinking. I have learned a lot from you.

My journey to become a scientist has reached a milestone. I have developed scientific skills and I hope to continue this path further on as I aspire for wisdom. It is a privilege to learn, to build and to discover. I am grateful to God for all this.

With respect to all,

Ela Pękalska

Curriculum vitae

Elżbieta Małgorzata Pękalska (Górniewicz) was born on 18th June 1972 in Wrocław, Poland.

Education and scientific work

2004: ..... Post-doc.
Information and Communication Theory group, Faculty of Electrical Engineering, Mathematics and Computer Science, TU Delft.

Main project: Proximity representations in pattern recognition.
Involved in various research, teaching and organizational tasks. Research oriented on theoretical foundations for the proximity-based learning and the design of domain-oriented proximity measures. Search for new applications.

1999 to 2003: ..... PhD student.
Pattern Recognition (currently Quantitative Imagining) group, Department of Imaging Science and Technology, Faculty of Applied Sciences, TU Delft.
PhD project: Dissimilarity-based pattern recognition.

Involved in various tasks, such as teaching MSc students and lecturing on postgraduate industrial courses, formulating research questions and finding answers, cooperation with other researchers, reviewing articles, preparation of publications and making presentations.

2002 to 2003: ..... Computer Service Practicum assistant (25% time).
Organizing and supervising practical self-study courses for MSc and PhD students at TU Delft.

2001: ..... 3-month research work.
Involved in the PowderScan project, a joint effort of TU Delft, TNO-TPD and Unilever R&D Vlaardingen, on structure analysis of detergents using multi-spectral imaging.

1998 to 1999: ..... Research fellow.
Involved in a project on multidimensional scaling in cooperation between TU Delft and Shell E&P.

1996 to 1997: ..... Research assistant (PhD student).
Department of Mathematics and Computer Science, Wrocław University, Poland.

1991 to 1996: ..... MSc in computer science, top 5%.
Entrance exam to the Wrocław University in 1991. Studying in Department of Mathematics and Computer Science, Wrocław University, Poland.
MSc project: 'Survival models applied to clearing time of invoices' (in Polish). A part of the MSc thesis was translated to English and published in 1998 in the Polish journal 'Operations research and decisions'.

1989 to 1990: ..... Math classes within the 'Talent Studies' program.
Technical University of Wrocław, Poland.
A special study program at mathematics organized for talented school pupils. After the exam in 1990, acceptance as a student at Technical University of Wrocław *one* year before the completion of secondary school.

1987 to 1991: ..... General education; major: math, chemistry and physics.
Secondary School no. XIV, Wrocław, Poland.
Acceptance to a special class in an earlier competitive entrance examination (30 pupils chosen from more than 600 candidates).

Awards

1995/1996: Scientific scholarship granted by the Ministry of National Education for the best students in Poland.

1995/1996: Scientific scholarship granted by the Foundation for Wrocław University for the best students at the university.

1992 to 1995: Scientific scholarships granted on yearly basis by Wrocław University for the best students.

Teaching and organizational experience

2000 to 2004: Lecturer and teaching instructor at pattern recognition courses for industry given under the lead of dr R.P.W. Duin at TU Delft. Participation in the set-up and a constant revision of these courses.

1999 to 2004: Teaching instructor at various lab courses (Matlab, statistics, pattern recognition) for MSc and PhD students at TU Delft. A lecturer for pattern recognition course for PhD students.

October-December 2003: Handling and evaluation of candidates for two available PhD positions.

2002 to 2003: Work (25%) for the Computer Service Practicum organizing and supervising practical self-study courses for MSc and PhD students at TU Delft. Set-up of the Matlab course.

March 2003: Co-organization of a pattern recognition course for international students within the Advanced Technology Higher Education Network (ATHENS) Socrates program.

2002: Teaching instructor at the industrial CBP course Image Processing for Industrial Applications given at TU Delft.

1996 to 1997: Teaching instructor at various computer lab courses, algebra and numerical analysis in the Institute of Computer Science, Wrocław University, Poland.

1996: Lecturer on computer science courses for school teachers organized by the Institute of Computer Science, Wrocław University, Poland.

1983 to 1995: Tutoring on weekly basis. Teaching and explaining mathematics, later also computer packages and computer software, to children, school pupils, students and adults.

Professional highlights

Membership:

- Advanced School for Computing and Imaging (ASCI); a Dutch research school
- International Association for Pattern Recognition (IAPR)

Reviewer, on regular basis, for a number of international conferences.
Reviewer for the following journals:

- IEEE Transactions on Pattern Analysis and Machine Intelligence
- IEEE Transactions on Neural Networks
- Journal of Machine Learning Research
- Pattern Recognition Letters

Scientific output

- **Conferences and Symposia:** Participation in 25 conferences and symposia.
- **Presentation of own work:** Approximately 30 oral presentations and six poster presentations on conferences, symposia or scientific meetings.
- **Publications:** Co-author of eight journal papers, one book section, 15 refereed international conference papers and four reports.

Journal Papers:

[1] E. Pękalska, R.P.W. Duin and P. Paclík, *Prototype Selection for Dissimilarity-based Classifiers*, to appear in Pattern Recognition.

[2] R.P.W.Duin, E. Pękalska, M.Skurichina, D.de Veld, H.J.C.M. Sterenborg, M.J.H. Witjes, L.N. Roodenburg, Combined classifiers for dissimilarity based representations of autofluorescence spectra applied to lesion recognition, to appear in IEEE Transactions on Systems, Man and Cybernetics.

[3] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pękalska and P. Paclík, *A study on combining image representations for image classification and retrieval*, International Journal of Pattern Recognition and Artificial Intelligence, vol. 18, no.5, 867-890, 2004.

[4] E. Pękalska and R.P.W. Duin, *Dissimilarity representations allow for building good classifiers*, Pattern Recognition Letters, vol. 23, no. 8, 943-956, 2002.

[5] E. Pękalska, P. Paclík and R.P.W. Duin, *A Generalized Kernel Approach to Dissimilarity-based Classification*, Journal of Machine Learning Research, Special Issue on Kernel Methods, vol. 2, no. 2, 175-211, 2002.

[6] E. Pękalska and R.P.W. Duin, *Automatic pattern recognition by similarity representations*, Electronics Letters, vol. 37, no. 3, 159-160, 2001.

[7] R.P.W. Duin, E. Pękalska and D. de Ridder, *Relational discriminant analysis*, Pattern Recognition Letters, vol. 20, no. 11-13, 1175-1181, 1999.

[8] E. Górniewicz (maiden), *Survival models applied to the clearing time of invoices*, Operations research and decisions (Badania Operacyjne i Decyzje) 2, 41-59, 1998.

Book Sections:

[9] E. Pękalska, Introduction to multidimensional scaling, in: J. Meij (eds), Dealing with the data flood, STT Netherlands, Study Center for Technology Trends, The Hague, 612-628, 2002.

Refereed International Conference Papers:

[10] R.P.W. Duin, E. Pękalska, P. Paclík and D.M.J.Tax, *The dissimilarity representation, a basis for domain based pattern recognition?*, in: L. Goldfarb (eds), Pattern representation and the future of pattern recognition (ICPR 2004 Workshop Proceedings, Cambridge UK, 22 August 2004), Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada, 43-56, 2004.

[11] R.P.W. Duin, E. Pękalska and D.M.J. Tax, *The characterization of classification problems by classifier disagreements*, in: J. Kittler, M. Petrou, M. Nixon (eds), Proc. International Conference on Pattern Recognition (22-26 August 2004, Cambridge UK), vol. 1, IEEE Computer Society, Los Alamitos, CA, 140-143, 2004.

[12] E. Pękalska, R.P.W. Duin, S. Günter and H. Bunke, *On not making dissimilarities Euclidean*, Proc. Joint IAPR International Workshops on SSPR and SPR, Lecture Notes in Computer Science, Springer-Verlag, 1143-1151, 2004.

[13] E. Pękalska, M. Skurichina and R.P.W. Duin, *Combining Dissimilarity Representations in One-class Classifier Problems*, Proc. International Workshop on Multiple Classifier Systems, (Cagliari, Italy), Lecture Notes in Computer Science, vol. 3077, Springer Verlag, Berlin, 2004.

[14] E. Pękalska, D.M.J. Tax and R.P.W. Duin, *One-class LP Classifiers for Dissimilarity Representations*, in: S. Becker, S. Thrun and K. Obermayer (eds), Advances in Neural Information Processing Systems, vol. 15, MIT Press, Cambridge, MA, 761-768, 2003.

[15] R.P.W. Duin and E. Pękalska, *Possibilities of zero-error recognition by dissimilarity representations*, in: J.M. Inesta, L. Mico (eds), Pattern Recognition in Information Systems (Proc. PRIS 2002, Alicante), ICEIS Press, Setubal, Portugal, 20-32, 2002.

[16] E. Pękalska and R.P.W. Duin, *Spatial representation of dissimilarity data via lower-complexity linear and nonlinear mappings*, in: T. Caelli, A. Amin, R.P.W. Duin, M. Kamel, D. de Ridder (eds), Structural and Syntactic, and Statistical Pattern Recognition, Proc. Joint IAPR International Workshops SSPR and SPR (Windsor, Canada), Lecture Notes in Computer Science, vol. 2396, Springer Verlag, Berlin, 470-478, 2002.

[17] E. Pękalska and R.P.W. Duin, *Prototype Selection for Finding Efficient Representations of Dissimilarity Data*, in: R. Kasturi, D. Laurendeau, C. Suen (eds), Proc. International Conference on Pattern Recognition (Quebec City, Canada), vol. III, IEEE Computer Society Press, Los Alamitos, 37-40, 2002.

[18] E. Pękalska, R.P.W. Duin and M. Skurichina, *A discussion on the classifier projection space for classifier combining*, in: F. Roli, J. Kittler (eds), Proc. International Workshop on Multiple Classifier Systems, MCS 2002 (Cagliari, Italy), Lecture Notes in Computer Science, vol. 2364, Springer Verlag, Berlin, 137-148, 2002.

[19] D. de Ridder, E. Pękalska and R.P.W. Duin, *The Economics of Classification: Error vs. Complexity*, in: R. Kasturi, D. Laurendeau, C. Suen (eds), Proc.Proc. International Conference on Pattern Recognition (Quebec City, Canada), vol. II, IEEE Computer Society Press, Los Alamitos, 244-247, 2002.

[20] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pękalska and P. Paclík, *On combining one-class classifiers for image database retrieval*, in: F. Roli, J. Kittler (eds), Proc. International Workshop on Multiple Classifier Systems, MCS 2002 (Cagliari, Italy), Lecture Notes in Computer Science, vol. 2364, Springer Verlag, Berlin, 212-221, 2002.

[21] R.P.W. Duin and E. Pękalska, *Complexity of Dissimilarity based Pattern Classes*, in: I. Austvoll (eds), Proc. Scandinavian Conference on Image Analysis, SCIA 2001 (Bergen, Norway), NOBIM, Stavanger, Norway, 663-670, 2001.

[22] E. Pękalska and R.P.W. Duin, *On Combining Dissimilarity Representations*, in: J. Kittler, F. Roli (eds), Proc. International Workshop on Multiple Classifier Systems, MCS 2001 (Cambridge, UK), Lecture Notes in Computer Science, vol. 2096, Springer Verlag, Berlin, 359-368, 2001.

[23] E. Pękalska and R.P.W. Duin, *Classifiers for dissimilarity-based pattern recognition*, in: A. Sanfeliu, J.J. Villanueva, M. Vanrell, R. Alquezar, A.K. Jain, J. Kittler (eds), Proc. International Conference on Pattern Recognition (Barcelona, Spain), vol. 2, Pattern Recognition and Neural Networks, IEEE Computer Society Press, Los Alamitos, 12-16, 2000.

[24] E. Pękalska, M. Skurichina and R.P.W. Duin, *Combining Fisher Linear Discriminants for Dissimilarity Representations*, in: J. Kittler, F. Roli (eds), Proc. International Workshop on Multiple Classifier Systems, MCS 2000 (Cagliari, Italy), Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 117-126, 2000.

Other Conference Papers:

[25] C. Lai, D.M.J. Tax, R.P.W. Duin, E. Pękalska and P. Paclík, *Database retrieval: the use of combined dissimilarities*, in: S. Vassiliades, L.M.J. Florack, J.W.J. Heijnsdijk, A. van der Steen (eds), Proc. Annual Conference of the Advanced School for Computing and Imaging (Heijen, NL), 177-184, 2003.

[26] E. Pękalska and R.P.W. Duin, *Is combining useful for dissimilarity representations?*, in: R.L. Lagendijk, J.W.J. Heijnsdijk, A.D. Pimentel, M.H.F. Wilkinson (eds), Proc. Annual Conference of the Advanced School for Computing and Imaging (Heijen, NL), 154-161, 2001.

[27] E. Pękalska and R.P.W. Duin, *Classification on dissimilarity data: a first look*, in: L.J. van Vliet, J.W.J. Heijnsdijk, T. Kielman, P.M.W. Knijnenburg (eds), Proc. Annual Conference of the Advanced School for Computing and Imaging (Lommel, Belgium), 221-228, 2000.

[28] E. Pękalska, D. de Ridder, R.P.W. Duin and M.A. Kraaijveld, *A new method of generalizing Sammon mapping with application to algorithm speed-up*, in: M. Boasson, J.A. Kaandorp, J.F.M. Tonino, M.G. Vosselman (eds), Proc. Annual Conference of the Advanced School for Computing and Imaging (Heijen, NL), 221-228, 1999.

[29] A. Ypma, E. Pękalska and R.P.W. Duin, *Domain approximation for condition monitoring*, in: B.M.ter Haar Romeny, D.H.J. Epema, J.F.M. Tonino, A.A. Wolters (eds), Proc. Annual Conference of the Advanced School for Computing and Imaging (Lommel, Belgium), 257-263, 1998.

Miscellaneous

[30] R.P.W. Duin, P. Juszczak, D. de Ridder, P. Paclík, E. Pękalska and D.M.J. Tax, *PRTools, a Matlab toolbox for pattern recognition*, http://prtools.org, 2004.