Pattern recognition, steps in science and consciousness.

Tutorial, Barcelona, 7 July 2008

Robert P.W. Duin, Delft University of Technology
r.duin@ieee.org

Abstract

Pattern recognition is a very general technology useful for the automatic detection and classification of patterns in data. As such it can be used in almost any application of intelligent sensors as well as in all areas of research that need to find regularities in observations. The basic steps in pattern recognition, representation and generalization, will be introduced and discussed. It will thereby be made clear how learning from observations works, but also some paradoxical examples will be shown in which more observations result in more confusion instead of more knowledge.

Pattern recognition is primarily a human ability. Patterns are defined by human beings. Recognition is thereby related to finding concepts and the art of naming. As such, attempts to build automatic pattern recognition systems have to rely on the understanding of the human ability to do so. Consequently, pattern recognition is also a very basic area of science. It suffers from prejudices and paradigm shifts. It's progress depends on the human possibility of self-understanding, and thereby on his consciousness.

In the first part of the presentation the technology of pattern recognition will be discussed. In the second part a broader scientific perspective will be taken, illustrated by some personal failures and steps in science and consciousness.

Slides:

0. Introduction

1. Generalization

2. Representation

3. Paradigms

4. Consciousness

These slides, especially the ones on consciousness, are intended to illustrate the oral presentation and generate a discussion. They should not be considered to present some definite facts or truths.

Papers:

R.P.W. Duin and E. Pekalska, Structural inference of sensor-based measurements, in: D.-Y. Yeung, J.T. Kwok, A. Fred, F. Roli and D. de Ridder (eds.), *Structural, Syntactic, and Statistical Pattern Recognition*, LNCS, vol. 4109, Springer Verlag, Berlin, 2006, 41-55.

R.P.W. Duin and E. Pekalska, The Science of Pattern Recognition; Achievements and Perspectives, in: W. Duch, J. Mandziuk (eds.), *Challenges for Computational Intelligence*, Studies in Computational Intelligence, vol. 63, Springer, 2007, 221-259.

Pattern Recognition: Steps in Science and Consciousness

Tutorial
Barcelona, 7 July 2008

Robert P.W. Duin

Pattern Recognition Group
Delft University of Technology
The Netherlands

//ict.ewi.tudelft.nl/~duin

TUDelft

Contents

- Pattern Recognition in Science
- The Generalization Problem
- The Representation Problem
- Steps in Science: Paradigm Shifts
- Steps in Consciousness: Patterns in Pattern Recognition

TUDelft

Pattern Recognition Problem

What is this? What occasion? Where are the faces? Who is who?

TUDelft

Pattern Recognition: OCR

Kurzweil Reading Edge

Automatic text reading machine with speech synthesizer

TUDelft

Pattern Recognition: Traffic

Road sign detection
Road sign recognition

TUDelft

Pattern Recognition: Traffic

License Plates

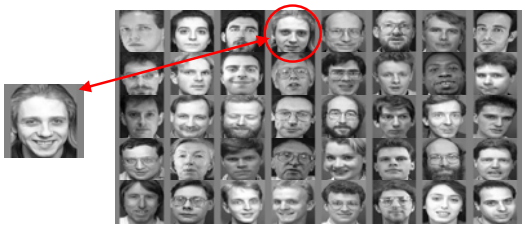
TUDelft

Pattern Recognition: Images

100 Objects

TU Delft

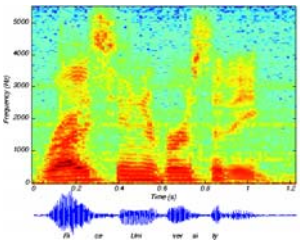
Pattern Recognition: Faces



Is he in the database? **Yes!**

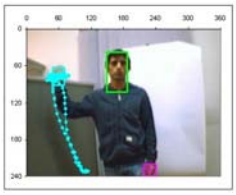
TU Delft

Pattern Recognition: Speech



TU Delft

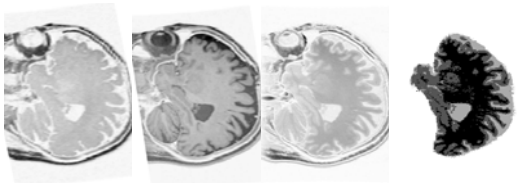
Gesture Recognition



Is this gesture in the database?

TU Delft

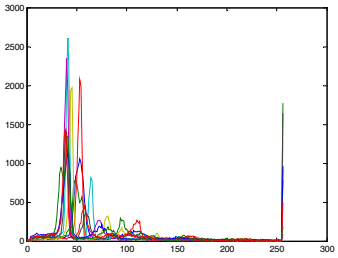
Pattern Recognition: Pathology



MRI Brain Image

TU Delft

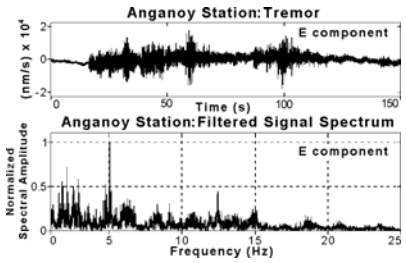
Pattern Recognition: Pathology



Flow cytometry

TU Delft

Pattern Recognition: Seismics



Earthquakes

TUDelft

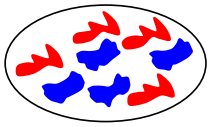
Pattern Recognition: Shape Recognition

Pattern Recognition is very often Shape Recognition:

- Images: B/W, grey value, color, 2D, 3D, 4D
- Time Signals
- Spectra

TUDelft

Pattern Recognition: Shapes



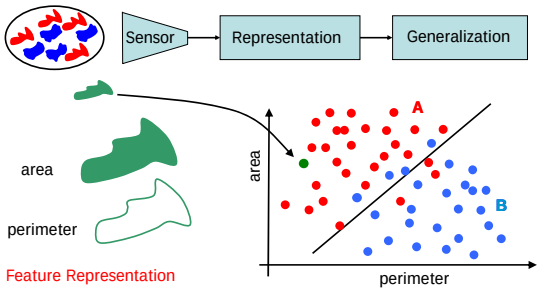
Examples of objects for different classes

Object of unknown class to be classified

A ? B

TUDelft

Pattern Recognition System



TUDelft

Pattern Recognition Issues

- **Generalization**: how to define mathematically a recognition system, e.g. a classifier.
- **Evaluation**: how to determine the performance of a recognition system.
- **Complexity**: how is the performance of a recognition system related to its complexity?
- **Representation**: how to represent a real world problem in mathematical terms such that generalization can be done well.
- How does **pattern recognition** relate to **science** in general?
- Examples of **paradigm shifts** in pattern recognition.
- **Consciousness** as a condition for pattern recognition.

TUDelft

Pattern Recognition: Steps in Science and Consciousness

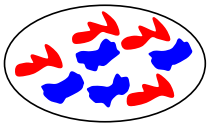
Tutorial
Barcelona, 7 July 2008

Robert P.W. Duin

Pattern Recognition Group
Delft University of Technology
The Netherlands

//ict.ewi.tudelft.nl/~duin

Generalization: Learning from Examples

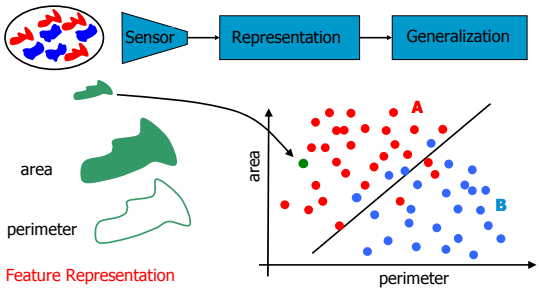


Examples of objects for different classes

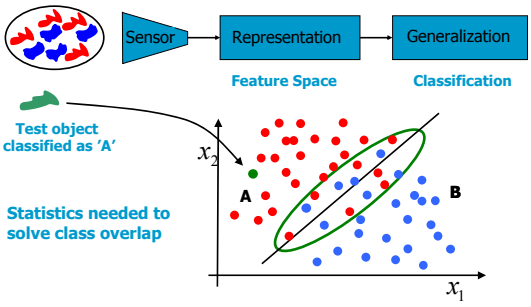
Object of unknown class to be classified

A ? B

Pattern Recognition System



Pattern Recognition System



Bayes decision rule, formal

$$\text{Bayes:}\quad \frac{p(x|A)\,p(A)}{p(x)} > \frac{p(x|B)\,p(B)}{p(x)} \rightarrow \text{A else B}$$

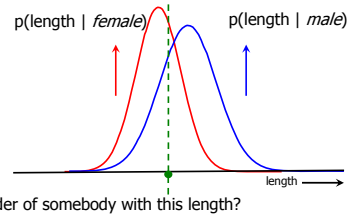
$$p(A|x) > p(B|x) \rightarrow \text{A else B}$$

$$p(x|A)\,p(A) > p(x|B)\,p(B) \rightarrow \text{A else B}$$

2-class problems: $S(x) = p(x|A)\,p(A) - p(x|B)\,p(B) > 0 \rightarrow$ A else B

n-class problems: $\text{Class}(x) = \text{argmax}_{\omega}(p(x|\omega)\,p(\omega))$

Decisions based on densities



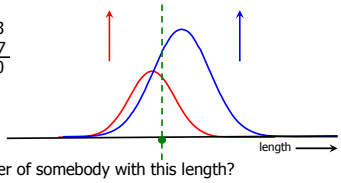
What is the gender of somebody with this length?

Bayes: { p(*female* | length) = p(length | *female*) p(*female*) / p(length)
p(*male* | length) = p(length | *male*) p(*male*) / p(length)

Decisions based on densities

p(length | *female*) p(*female*) p(length | *male*) p(*male*)

p(*female*) = 0.3
p(*male*) = 0.7
1.0



What is the gender of somebody with this length?

Bayes: { p(*female* | length) = p(length | *female*) p(*female*) / p(length)
p(*male* | length) = p(length | *male*) p(*male*) / p(length)

Bayes decision rule

p(*female* | length) > p(*male* | length) → *female* else *male*

Bayes:

$$\frac{p(\text{length} \mid \textit{female})\, p(\textit{female})}{p(\text{length})} > \frac{p(\text{length} \mid \textit{male})\, p(\textit{male})}{p(\text{length})}$$

p(length | *female*) p(*female*) > p(length | *male*) p(*male*) → *female* else *male*

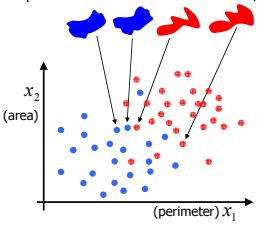
pdf estimated from training set

class prior probabilities known, guessed or estimated

Compactness

Representations of real world similar objects are close.
There is no ground for any generalization (induction) on representations that do not obey this demand.

(A.G. Arkedev and E.M. Braverman, Computers and Pattern Recognition, 1966.)



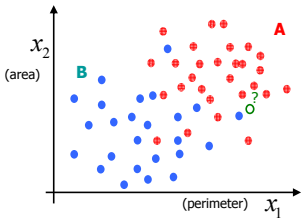
The compactness hypothesis is not sufficient for perfect classification as dissimilar objects may be close.
→ class overlap
→ probabilities

Distances and Densities

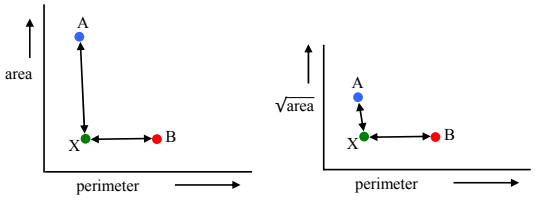
? to be classified as

B – because it is most close to an object A

A – because the local density of B is larger.



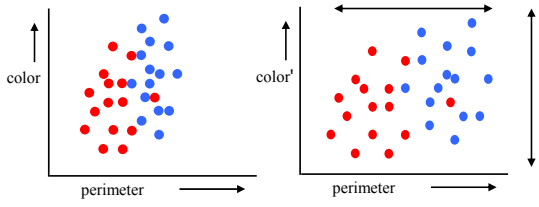
Distances: Scaling Problem



Before scaling: D(X,A) < D(X,B)

After scaling: D(X,A) > D(X,B)

How to Scale in Case of no Natural Features

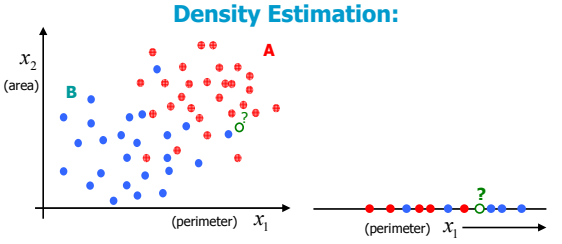


Make variances equal:

$$\text{color}' = \frac{\text{color}}{\sqrt{\text{var}(\text{color})}}$$

$$\text{perimeter}' = \frac{\text{perimeter}}{\sqrt{\text{var}(\text{perimeter})}}$$

Density Estimation:



What is the probability of finding an object of class A (B) on this place in the 2D space?

What is the probability of finding an object of class A (B) on this place in the 1D space?

TUDelft

Density estimation (1)

- The density is defined on the whole feature space.
- Around object x, the density is defined as:

$$p(x) = \frac{dP(x)}{dx} = \left(\frac{\text{fraction of objects}}{\text{volume}} \right)$$

- Given n measured objects, e.g. person's height (m) how can we estimate $p(x)$?

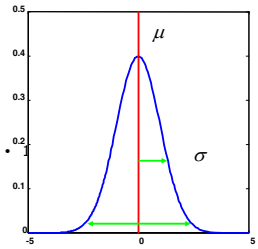
TUDelft

Density estimation (2)

- **Parametric** estimation:
  - Assume a parameterized model, e.g. Gaussian
  - Estimate parameters from data
  - Resulting density is of the assumed form
- **Non parametric** estimation:
  - Assume no 'formal' structure/model, choose 'approach'
  - Estimate density with chosen approach
  - Resulting density has no formal form

TUDelft

The Gaussian distribution (3)

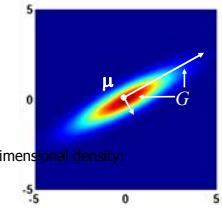


- Normal distribution = Gaussian distribution
- Standard normal distribution: $\mu = 0,\ \sigma^2 = 1$
- 95% of data between $[\mu - 2\sigma, \mu + 2\sigma]$ (in 1D!)

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2} \right)$$

TUDelft

Multivariate Gaussians



$$G = \begin{bmatrix} 3 & 1\tfrac{1}{2} \\ 1\tfrac{1}{2} & 2 \end{bmatrix}$$

- k - dimensional density:

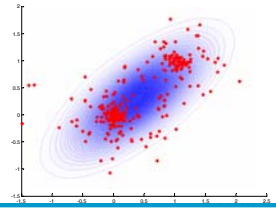
$$p(x) = \frac{1}{\sqrt{2\pi^k \det(G)}} \exp\left(-\frac{1}{2} (x-\mu)^{\mathrm{T}} G^{-1} (x-\mu) \right)$$

TUDelft

Parametric estimation

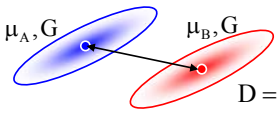
- Assume Gaussian model
- Estimate mean and covariance from data

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad G = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})^{\mathrm{T}} (x_i - \hat{\mu})$$



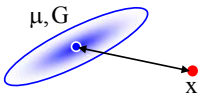
TUDelft

Mahalanobis Distance



$$D = (\mu_A - \mu_B)^T G^{-1} (\mu_A - \mu_B)$$

Distance between two classes



$$D = (\mu - x)^T G^{-1} (\mu - x)$$

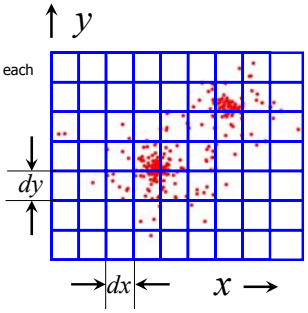
Distance between a point and a class

Nonparametric estimation

Histogram method:

1. Divide feature space in N^2 bins
2. Count the number of objects in each bin
3. Normalize:

$$\hat{p}(x) = \frac{n_{ij}}{\sum_{i,j=1}^{N} n_{ij} dx dy}$$



Parzen density estimation (1)

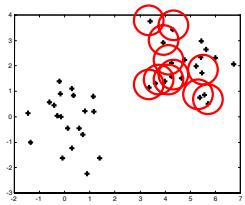
- Fix volume of bin, vary positions of bins, add contribution of each bin
- Define 'bin'-shape (kernel):

$$K(\mathbf{r}) > 0$$

$$\int K(\mathbf{r})\, d\mathbf{r} = 1$$

- For test object z sum all bins

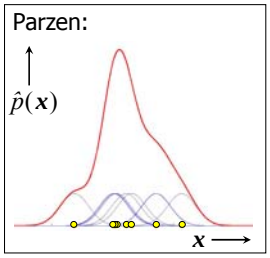
$$p(\mathbf{z}) = \frac{1}{hn} \sum_i K\left(\frac{\mathbf{z} - \mathbf{x}_i}{h}\right)$$



Parzen density estimation (2)

- With Gaussian kernel:

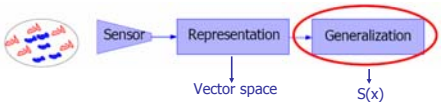
$$K(x) = \frac{1}{h\sqrt{2\pi}} \exp\left(\frac{x^2}{2h^2}\right)$$

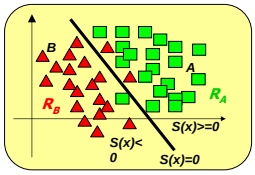


Parametric vs. Nonparametric

- **Parametric** estimation, based on some model:
  - Model parameters to estimate
  - More samples required than parameters
  - Model assumption could be incorrect resulting in erroneous conclusions
- **Non parametric** estimation, hangs on data directly:
  - Assume no 'formal' structure/model
  - Almost no parameters to estimate
  - Erroneous estimates are less likely

Classification, Discriminant Analysis





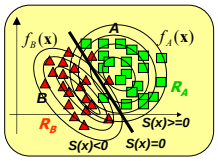
Supervised learning: how to learn S(x) from examples?

Classification error

Classification error ε is the probability that an arbitrary **x** is erroneously classified by a decision rule S(**x**):

If $S(\mathbf{x}) \geq 0$, then $\mathbf{x} \rightarrow$ A

If $S(\mathbf{x}) < 0$, then $\mathbf{x} \rightarrow$ B



$$\varepsilon = P(S(\mathbf{x}) < 0, \mathbf{x} \in A) + P(S(\mathbf{x}) \geq 0, \mathbf{x} \in B)$$

$$\varepsilon = P(S(\mathbf{x}) < 0 \mid \mathbf{x} \in A)p(A) + P(S(\mathbf{x}) \geq 0 \mid \mathbf{x} \in B)p(B)$$

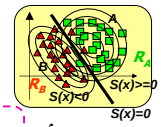
$$\varepsilon = \underbrace{\int_{S(\mathbf{x})<0} p(A) f_A(\mathbf{x}) d\mathbf{x}}_{\varepsilon_A} + \underbrace{\int_{S(\mathbf{x})\geq 0} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{\varepsilon_B} \qquad p(A) + p(B) = 1$$

$f_A(\mathbf{x})$ and $f_B(\mathbf{x})$ are the probability density functions (pdf's) of A and B.

TUDelft

The optimal rule is the Bayes decision rule

Determine the optimal S(**x**) such that $\varepsilon = \underbrace{\int_{S(\mathbf{x})<0} p(A) f_A(\mathbf{x}) d\mathbf{x}}_{\varepsilon_A} + \underbrace{\int_{S(\mathbf{x})\geq 0} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{\varepsilon_B}$ is **minimum**.



$$\varepsilon = \int_{\substack{R_B: \\ S(\mathbf{x})<0}} p(A) f_A(\mathbf{x}) d\mathbf{x} + \underbrace{\int_{\substack{R_A: \\ S(\mathbf{x})\geq 0}} p(B) f_B(\mathbf{x}) d\mathbf{x} + \int_{\substack{R_B: \\ S(\mathbf{x})<0}} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{P(B)} - \int_{\substack{R_B: \\ S(\mathbf{x})<0}} p(B) f_B(\mathbf{x}) d\mathbf{x}$$

$$\varepsilon = p(B) + \int_{R_B: S(\mathbf{x})<0} [p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})] d\mathbf{x}$$

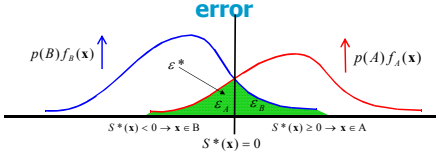
This is minimum if $[p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})] < 0$ over R_B: S(x) <0.

So, the **optimal rule** is the **Bayes decision rule.**

$$S^*(x) = p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})$$

TUDelft

Optimal classification error = Bayes error



Classification error is minimal, ε*, if the decision function is optimal. This is the **Bayes error**, the lowest achievable error!

Bayes decision rule: $S^*(\mathbf{x}) = p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})$

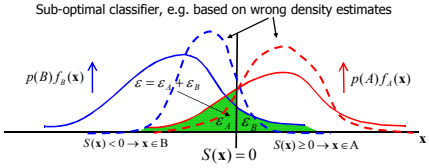
Bayes error: $\varepsilon^* = \int \min\{p(A) f_A(\mathbf{x}),\ p(B) f_B(\mathbf{x})\} d\mathbf{x}$

Bayes error is only reachable if true distributions are known.

TUDelft

Classification error

Sub-optimal classifier, e.g. based on wrong density estimates



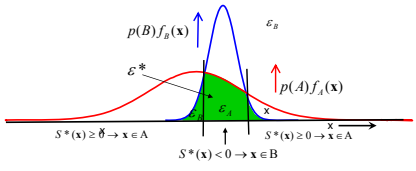
$$\varepsilon = \underbrace{\int_{S(\mathbf{x})<0} p(A) f_A(\mathbf{x}) d\mathbf{x}}_{\varepsilon_A} + \underbrace{\int_{S(\mathbf{x})\geq 0} p(B) f_B(\mathbf{x}) d\mathbf{x}}_{\varepsilon_B}$$

$p(A), p(B), f_A(\mathbf{x}), f_B(\mathbf{x})$ estimated by parametric or non-parametric approaches

S(**x**)=0 discriminant function, e.g. piece-wise linear

TUDelft

Bayes rule for different distributions



$$S^*(\mathbf{x}) = p(A) f_A(\mathbf{x}) - p(B) f_B(\mathbf{x})$$

$$\varepsilon^*(\mathbf{x}) = \int \min\{p(A) f_A(\mathbf{x}),\ p(B) f_B(\mathbf{x})\} d\mathbf{x}$$

TUDelft

Bayes decision making (how-to)

$$\text{class}(\mathbf{x}) = \arg\max_{\omega_i} p(\mathbf{x} \mid \omega_i) p(\omega_i)$$

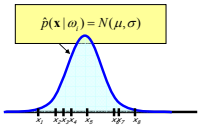
Assumes knowledge about class pdf's.

$$f_i(\mathbf{x}) = \hat{p}(\mathbf{x} \mid \omega_i) \qquad \hat{p}(\omega_i) = \frac{|\omega_i|}{\text{Total}}$$

Two approaches to estimate pdf's:

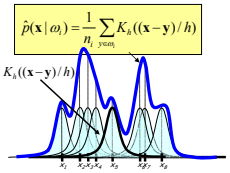
1) Parametric

- Assume a particular model for each pdf
- Estimate parameters from the training set
- E.g. Gaussian, Mixtures of Gaussians



2) Non-parametric

- No assumption about the underlying pdf (restrict the smoothness of the estimate)
- Estimate locally and combine globally
- E.g. Histogram, k-NN, Parzen



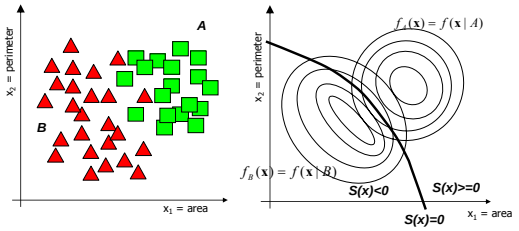
TUDelft

Bayes rule: summary

- **Bayes decision rule** is optimal when both class priors and pdf's are known.
- Usually, we have to estimate both pdf's and priors from the data, which leads to estimation errors. We may approach the Bayes error only for very large training sets.
- In other cases additional costs or risk are involved. E.g:
  - it is very risky to classify an ill patient as healthy
  - it is less risky to classify a healthy patient as ill (extra tests)

  In this situation we have to adapt the formulation to the minimum cost classification.

Discriminant Analysis



Probability density estimates of the classes

Quadratic discriminant=Bayes rule for Normal Distributions [G]

Bayes rule $S(\mathbf{x}) = p(A)p(\mathbf{x}\mid A) - p(B)p(\mathbf{x}\mid B) = 0$

logs don't matter

$$p(A)p(\mathbf{x}\mid A) = p(B)p(\mathbf{x}\mid B)$$

$$\log[p(A)p(\mathbf{x}\mid A)] = \log[p(B)p(\mathbf{x}\mid B)]$$

R(x) and S(x) have the same signs

$$R(\mathbf{x}) = \log(p(A)p(\mathbf{x}\mid A)) - \log(p(B)p(\mathbf{x}\mid B))$$

$$R(\mathbf{x}) = \log(p(\mathbf{x}\mid A)) - \log(p(\mathbf{x}\mid B) + \log[p(A)/p(B)]$$

Normal distribution

$$p(\mathbf{x}\mid A) = \frac{1}{\sqrt{2\pi^k \det(\boldsymbol{\Sigma}_A)}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_A)^T \Sigma_A^{-1}(\mathbf{x}-\boldsymbol{\mu}_A)\right)$$

$$\log(p(\mathbf{x}\mid A)) = -\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_A)^T \Sigma_A^{-1}(\mathbf{x}-\boldsymbol{\mu}_A) - \log(\sqrt{2\pi^k \det(\boldsymbol{\Sigma}_A)})$$

Quadratic expression

Substitute

$$R(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_A)^T \hat{\Sigma}_A^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_A) + \frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_B)^T \hat{\Sigma}_B^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_B) + \text{const}$$

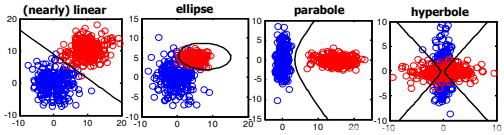
$$\text{const} = \log\{p(A)/(p(B)\} + \frac{1}{2}\log\{\det(\hat{\Sigma}_B)/\det(\hat{\Sigma}_A)\}$$

qdc udc

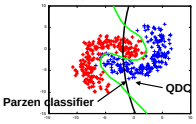
Quadratic discriminant functions

$$R(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_A)^T \hat{\Sigma}_A^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_A) + \frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_B)^T \hat{\Sigma}_B^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_B) + \text{const}$$

$$\text{const} = \log\{p(A)/(p(B)\} + \frac{1}{2}\log\{\det(\hat{\Sigma}_B)/\det(\hat{\Sigma}_A)\}$$



QDC assumes that classes are normally distributed. **Wrong** decision boundaries are estimated if this does not hold.



Bayes rule for Normal Distributions with Equal Covariances [G]

QDC

$$R(\mathbf{x}) = -\frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_A)^T \hat{\Sigma}_A^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_A) + \frac{1}{2}(\mathbf{x}-\hat{\boldsymbol{\mu}}_B)^T \hat{\Sigma}_B^{-1}(\mathbf{x}-\hat{\boldsymbol{\mu}}_B) + \text{const}$$

Assume Σ_A and Σ_B are equal: $\Sigma=\Sigma_A=\Sigma_B$. Quadratic term disappears.

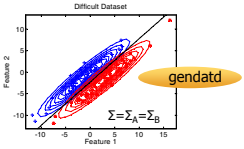
Linear expression

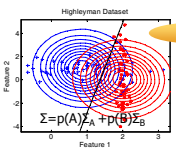
LDC

$$R(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T \hat{\Sigma}^{-1}\mathbf{x} + \text{const}$$

ldc

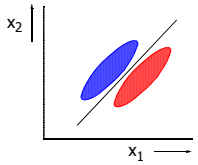
$$const = -\frac{1}{2}\hat{\boldsymbol{\mu}}_A{}^T \hat{\Sigma}^{-1}\hat{\boldsymbol{\mu}}_A + \frac{1}{2}\hat{\boldsymbol{\mu}}_B{}^T \hat{\Sigma}^{-1}\hat{\boldsymbol{\mu}}_B + \log[p(A)/p(B)]$$





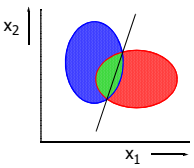
Unequal covariance matrices → use linear approximation $\Sigma=p(A)\Sigma_A+p(B)\Sigma_B$

Linear discriminant function (summary) [G]



Normal distributions with equal covariance matrices Σ are optimally separated by a linear classifier

$$S(\mathbf{x}) = (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T \Sigma^{-1}\mathbf{x} + \text{const}$$



Optimal classifier for normal distributions with unequal covariance matrices Σ_A and Σ_B can be approximated by:

$$S(\mathbf{x}) = (\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T (p(A)\Sigma_A + p(B)\Sigma_B)^{-1}\mathbf{x} + \text{const}$$

ldc

Fisher linear discriminant (I)

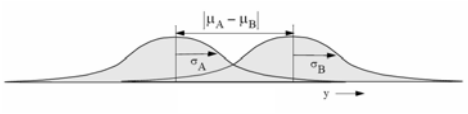
Assume a two-class problem. We look for a linear discriminant:

$$S(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

such that the **separability** between the classes **is maximized** along **w.**

Fisher criterion:

$$J_F = \frac{\sigma^2_{\text{Between-class}}}{\sigma^2_{\text{Within-class}}} = \frac{|\mu_A - \mu_B|^2}{\sigma_A^2 + \sigma_B^2}$$



TUDelft

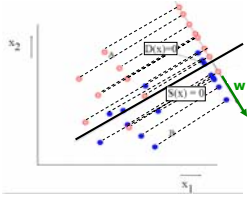
Fisher linear discriminant (II)

Fisher criterion along the direction w: fisherc

$$J_F = \frac{|\mathbf{w}^T\boldsymbol{\mu}_A - \mathbf{w}^T\boldsymbol{\mu}_B|^2}{\mathbf{w}^T p_A\Sigma_A\mathbf{w} + \mathbf{w}^T p_B\Sigma_B\mathbf{w}} = \frac{\mathbf{w}^T(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)^T(\boldsymbol{\mu}_A - \boldsymbol{\mu}_B)\mathbf{w}}{\mathbf{w}^T(p_A\Sigma_A + p_B\Sigma_B)\mathbf{w}} = \frac{\mathbf{w}^T\Sigma_B\mathbf{w}}{\mathbf{w}^T\Sigma_W\mathbf{w}}$$

Σ_B is the between-class covariance matrix.

Σ_W is the within-class covariance matrix.



Solution for $\Sigma_W = \Sigma$:

$$\mathbf{w} = \hat{\Sigma}_W^{-1}(\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)$$

$$S(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T\hat{\Sigma}^{-1}\mathbf{x} + \text{const}$$

Same as LDC up to a constant.

No assumption is made about normality of the data.

TUDelft

Nearest mean classifier (NMC) [G]

Assume $\Sigma = \Sigma_A = \Sigma_B = I$. Linear discriminant becomes the nearest mean classifier.

NMC

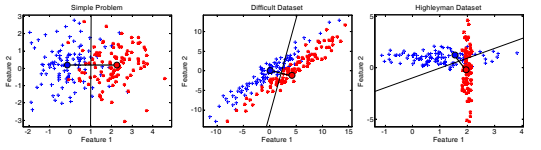
$$R(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T\mathbf{x} - (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T(\hat{\boldsymbol{\mu}}_A + \hat{\boldsymbol{\mu}}_B)/2$$

nmc

LDC, FisherC

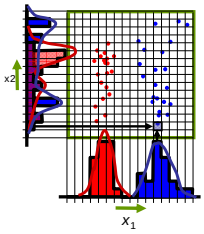
$$R(\mathbf{x}) = (\hat{\boldsymbol{\mu}}_A - \hat{\boldsymbol{\mu}}_B)^T\hat{\Sigma}^{-1}\mathbf{x} + \text{const}$$

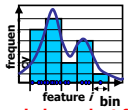
ldc fisherc



TUDelft

Naïve-Bayes classifier [G]





Assume class-independent features.

$$p(\mathbf{x}|A) = \prod_{i=1}^{k} p(x_i \mid A)$$

$$p(\mathbf{x}|B) = \prod_{i=1}^{k} p(x_i \mid B)$$

Estimate class probability density functions per feature: 1D histograms, 1D normal distributions, 1D Parzen estimates, etc. Multiply estimates.

Use Bayes decision rule: $S(\mathbf{x}) = p(A)f_A(\mathbf{x}) - p(B)f_B(\mathbf{x})$

$$\text{class}(\mathbf{x}) = \arg\max_{\omega_k} p(\mathbf{x} \mid \omega_k)p(\omega_k)$$

naivebc

TUDelft

Logistic model – logistic classifier [D]

- It holds for the Bayes discriminant:

$$p(A)p(\mathbf{x} \mid \text{A}) = p(B)p(\mathbf{x} \mid \text{B}) \rightarrow \log\{p(A)p(\mathbf{x} \mid \text{A})\} = \log\{p(B)p(\mathbf{x} \mid \text{B})\}$$

$$\rightarrow \log\left(\frac{p(A)p(\mathbf{x} \mid \text{A})}{p(B)p(\mathbf{x} \mid \text{B})}\right) = 0$$

- For linear discriminants, we have:

$$\log\left(\frac{p(A \mid \mathbf{x})}{p(B \mid \mathbf{x})}\right) = \log\left(\frac{p(A)p(\mathbf{x} \mid \text{A})}{p(B)p(\mathbf{x} \mid \text{B})}\right) = \mathbf{w}^T\mathbf{x} + w_0$$

Given that $p(B \mid \mathbf{x}) = 1 - p(A \mid \mathbf{x})$

$$p(A \mid \mathbf{x}) = \frac{p(A)p(\mathbf{x} \mid \text{A})}{p(A)p(\mathbf{x} \mid \text{A}) + p(B)p(\mathbf{x} \mid \text{B})} = \frac{e^{\mathbf{w}^T\mathbf{x}+w_0}}{1+e^{\mathbf{w}^T\mathbf{x}+w_0}} = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}-w_0}}$$

$$p(B \mid \mathbf{x}) = \frac{p(B)p(\mathbf{x} \mid \text{B})}{p(A)p(\mathbf{x} \mid \text{A}) + p(B)p(\mathbf{x} \mid \text{B})} = \frac{1}{1+e^{\mathbf{w}^T\mathbf{x}+w_0}}$$

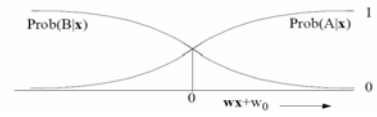
loglc

TUDelft

Logistic function

- It appears that $\log\left(\frac{p(A)p(\mathbf{x} \mid \text{A})}{p(B)p(\mathbf{x} \mid \text{B})}\right)$ is linear for many distributions.
- E.g. normal, binary, multimodal and mixtures of them.

$f(\mathbf{x}) = p(A \mid \mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}-w_0}}$ is called the logistic function.



See: Anderson, Logistic Discrimination, in : Handbook of Statistics, vol. 2, Krishnaiah and Kanal (eds.), North Holland, 1982, pp. 169 - 191

TUDelft

The Logistic Model, ML Estimation

- Observations X={$\mathbf{x}_1,..,\mathbf{x}_n$} depend on the unknown parameter $\boldsymbol{\theta}$.
- **Assumption:** data samples are **independent, identically distributed (iid):** f($\mathbf{x}_1,..,\mathbf{x}_n|\,\boldsymbol{\theta}$) = Π f($\mathbf{x}_i|\,\boldsymbol{\theta}$).
- Likelihood is a function of $\boldsymbol{\theta}$, samples $\mathbf{x}_i$ are fixed. L($\boldsymbol{\theta}$|X)=f($\mathbf{x}_1,..,\mathbf{x}_n|\boldsymbol{\theta}$)=$\Pi$ f($\mathbf{x}_i|\boldsymbol{\theta}$).
- **Maximum Likelihood**: $\boldsymbol{\theta}_{ML}$=argmax$_{\boldsymbol{\theta}}$ L($\boldsymbol{\theta}$|X) =argmax$_{\boldsymbol{\theta}}$ log L($\boldsymbol{\theta}$|X).

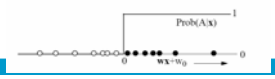
In the logistic model, we maximize the conditional log-likelihood:

$$\log L(\mathbf{w}) = \log\{\prod_{\mathbf{x}\in A} p(A\,|\,\mathbf{x};\mathbf{w})\prod_{\mathbf{x}\in B} p(B\,|\,\mathbf{x};\mathbf{w})\}$$

by using a gradient-descent method (steepest ascent or Newton) :

$$0 = \frac{\partial \log L(\mathbf{w})}{\partial w_j} = \sum_{\mathbf{x}\in A} x_j\, p(A\,|\,\mathbf{x};\mathbf{w}) - \sum_{\mathbf{x}\in B} x_j\, p(B\,|\,\mathbf{x};\mathbf{w})$$

For separable classes, maximum is at ∞, as p(A|**x**)=1 for x in A, and p(B|**x**)=1 for x in B.

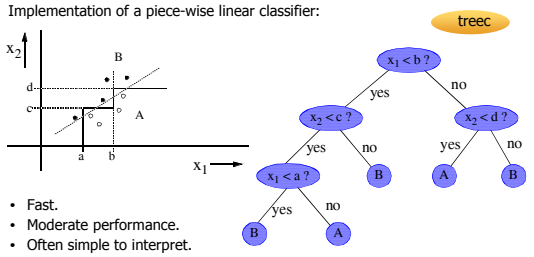


loglc

Decision trees [D]

Implementation of a piece-wise linear classifier:

treec



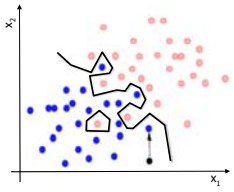
- Fast.
- Moderate performance.
- Often simple to interpret.
- Can handle numerical and categorical variables.

C4.5*-decision tree. An algorithm used to generate a decision tree developed by Ross Quinlan. See: J.R.Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.*

Nearest neighbor rule (1-NN rule) [D]

Assign a new object to the class of the nearest neighbor in the training set.

knnc



1-NN rule:

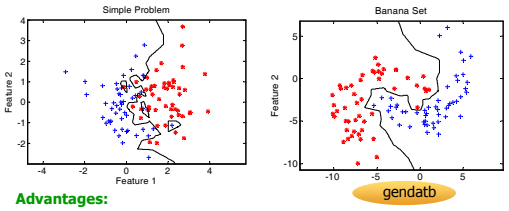
• Often relies on the Euclidean distance. Other distance measures can be used.

• Insensitive to prior probabilities!

• Scaling dependent. *Features should be scaled properly.*

There are **no** errors on the training set. The classifier is **overtrained**.

1-NN rule: examples



gendatb

Advantages:

- Simple.
- Works well for almost separable classes.
- Useful to shape non-linear decision functions.

Disadvantages:

- No training time. Long execution time.
- All data should be stored.

1-NN classification error

Asymptotically (for very large training sets):

$$\varepsilon^* \le \varepsilon_{\underset{n\to\infty}{1-NN}} \le 2\varepsilon^*(1-\varepsilon^*) \le 2\varepsilon^*$$

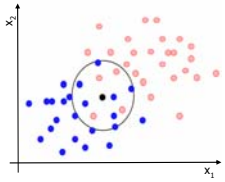
The nearest neighbor rule will not perform worse than twice the best possible classifier.

1-NN is often a very good classifier!!!!

k-nearest neighbor rule (k-NN) [D]

Assign an object to the class that is most frequently represented among k nearest neighbors in the training set of n objects.

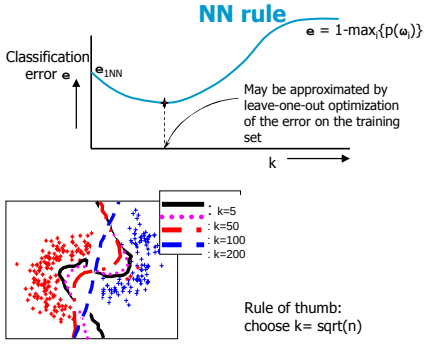
knnc



Less local than 1-NN.
More smooth.
Very global when k→n.

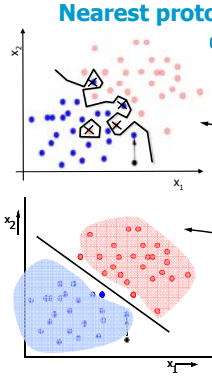
- k-NN class density estimates $\hat{p}(\mathbf{x}\,|\,\omega_j) = \frac{k_j}{n_j \mathrm{Vol}(\mathbf{x})}$
- Priors $\hat{p}(\omega_j) = \frac{n_j}{n}$
- Decision rule $\frac{k_k}{n_k \mathrm{Vol}(\mathbf{x})}\frac{n_k}{n} > \frac{k_j}{n_j \mathrm{Vol}(\mathbf{x})}\frac{n_j}{n}\ \forall j \ne k$ ($\hat{p}(\omega_j)\hat{p}(\mathbf{x}\,|\,\omega_j)$)
- Simplifies to **majority vote:** $k_k > k_j \quad \forall j \ne k$

k-NN decision boundaries: optimal k in the k-NN rule



Rule of thumb: choose k= sqrt(n)

Nearest prototype rule: editing and condensing



k-NN rule: distances to all training objects have to be computed.

Editing-and-condensing reduces the complexity while aiming at comparable classification accuracy.

edicon

Editing: removes objects that are misclassified by the k-NN rule.

Condensing: select a subset of prototypes such that the 1-NN rule performs similarly as on the complete training set.

Parzen classifier [G]

- Bayes decision rule

$$S(\mathbf{x}) = p(A)f_A(\mathbf{x}) - p(B)f_B(\mathbf{x})$$
$$\text{class}(\mathbf{x}) = \arg\max_{\omega_k} p(\mathbf{x}\mid\omega_k)p(\omega_k)$$

- Substitute Parzen density estimates

$$\hat{p}(\mathbf{x}\mid\omega_k) = \frac{1}{n_k}\sum_{\mathbf{x}_i\in\omega_k} K_h\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)$$

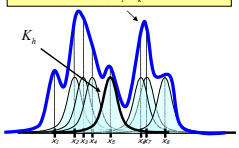
- Parzenc: optimize *h* for classification

parzenc

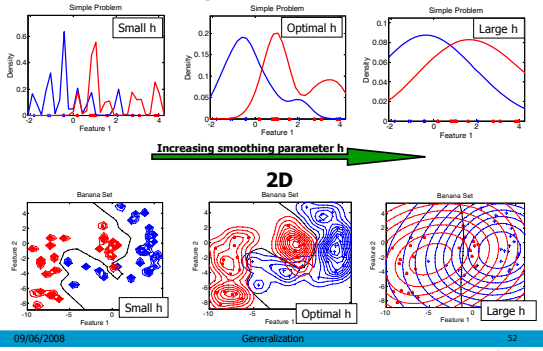
- Parzendc: optimize *h* for density estimation per class

parzendc

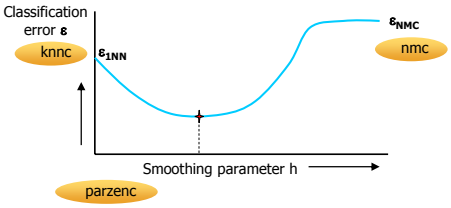
$$\hat{p}(\mathbf{x}\mid\omega_k) = \frac{1}{n_k}\sum_{\mathbf{x}_i\in\omega_k} K_h\left(\frac{\mathbf{x}_i-\mathbf{x}}{h}\right)$$



Parzen: density estimates vs the smoothing parameter h



Parzen classifier performance

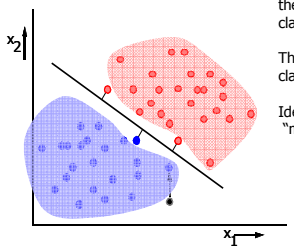


Parzen classifier:

- Small smoothing parameter: 1-NN performance, ε → ε$_{1NN}$
- Large smoothing parameter: Nearest mean performance, ε → ε$_{NMC}$

Support vector machine (SVM) [D]

1995-2005



For linearly-separable classes find the few objects that determine the classifier. These are **support vectors**.

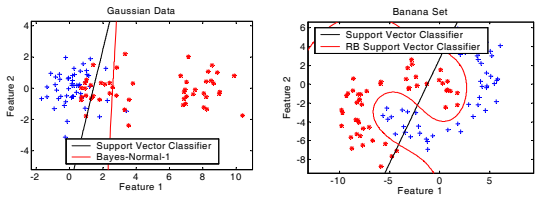
They have the same distance to the classifier: *the margin*.

Identical to "maximum-margin classifier"

$$S(x) = \sum_i \alpha_i (x_i^T x)$$
$$S(x) = w^T x, \min(w^T w)$$

SVC

SVM: Examples



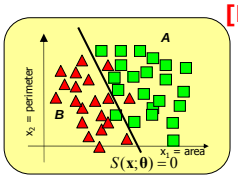
svc ldc

`svc` versus `ldc` for classes with very different domains

svc rbsvc

Linear (`svc`) versus nonlinear support vector classifier (`rbsvc`)

Classifiers based on error optimization [D]



perlc lmnc bpxnc

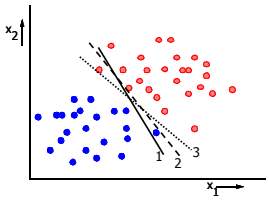
If $S(\mathbf{x};\boldsymbol{\theta}) \geq 0$, then $\mathbf{x} \rightarrow$ A
If $S(\mathbf{x};\boldsymbol{\theta}) < 0$, then $\mathbf{x} \rightarrow$ B

Change parameters **θ** of the decision function such that the classification error is minimized. Gradient-based techniques are used to solve nonlinear equations.

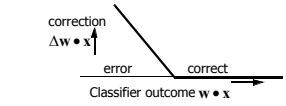
Error function: $J(\boldsymbol{\theta}) = \sum_{\mathbf{x} \text{ in Training set}} F(S(\mathbf{x};\boldsymbol{\theta}))$

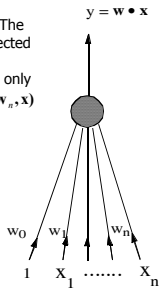
E.g. error count, average error, sum of distances to the boundary

Perceptron [D]



Linear classifier. The weights are corrected for erroneously classified objects only
$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}(\mathbf{w}_n, \mathbf{x})$





perlc

Neural network classifiers [D]

1985-1995

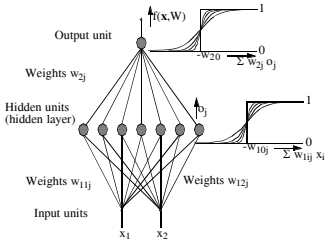
Number of layers with identical neurons (simple linear classifiers) with non-linear transitions in between (sigmoids). Leads to a moderately non-linear classifier.

Trained object by object to minimize the MSE on the output compared with targets (labels).

Tricky training procedure.

Slow training and execution, unless special hardware is used.

Good performance; danger of overtraining.



rnnc neurc bpxnc lmnc rbnc

Classifier outputs

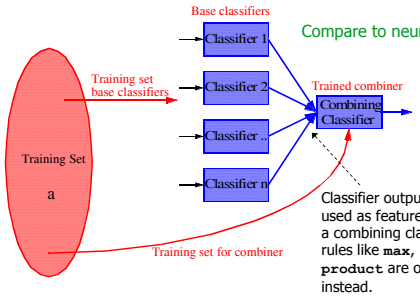
What are the possible outcomes of y = classifier(x)?

- Label, $y_1 \in$ {'apple','banana'}.
- $y_2 \in \{0,1\}$ as crisp numeric labels
- $y_3 \in [0,1]$ for soft labels (confidences)
- $y_4 \in [0,\infty)$ for distances to a class
- $y_5 \in (-\infty,+\infty)$ for distances to a classifier

Conversions are often made, e.g.:

```
y2 = (y1 == 'apple')
y2 = round(y3)
y3 = sigm(y5)
y5 = invsigm(y3)
```


Combining classifiers



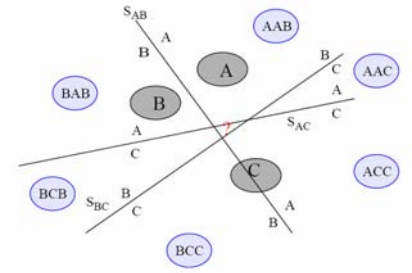
Compare to neural networks

Classifier outputs may be used as features for training a combining classifier. Fixed rules like `max`, `sum` and `product` are often used, instead.

parallel stacked

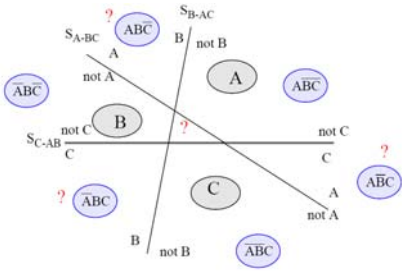
Multiple classifiers (I)

Undecidable region in case of multiple 2-class discriminants.



Multiple classifiers (II)

Undecidable regions in case of multiple one-vs-all-other discriminants.

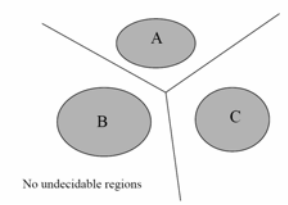


Multiple classifiers (III)

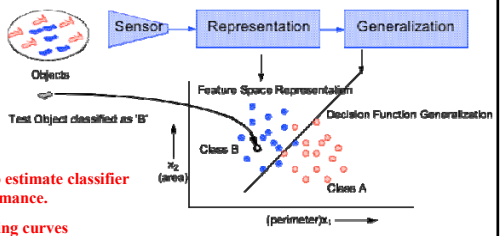
PRTools

Instead of discriminants, use class description functions: class probability density functions, Euclidean or Mahalanobis distances.

If $D(\mathbf{x},\omega_k) > D(\mathbf{x},\omega_i)$ for all $i \neq k$ then $x \rightarrow \omega_k$



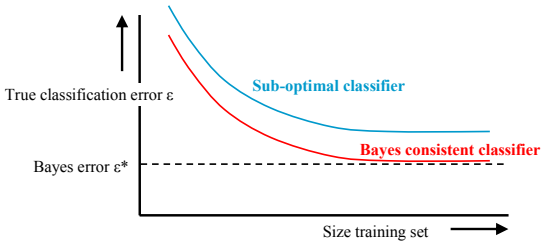
Classifier Evaluation



- How to estimate classifier performance.
- Learning curves
- Feature curves
- Rejects and ROC curves

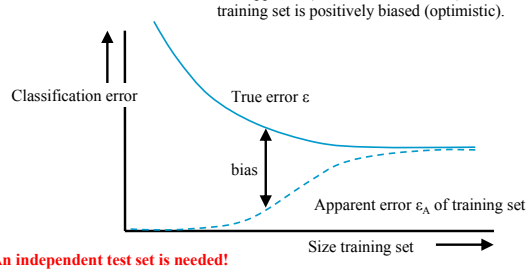
Learning Curve

`cleval`
`testc`



The Apparent Classification Error

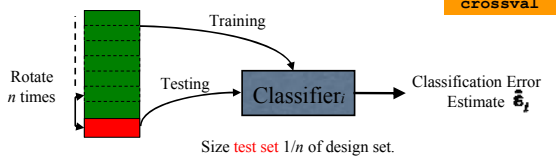
The apparent (or resubstitution error) of the training set is positively biased (optimistic).



An independent test set is needed!

Cross-validation

`crossval`



Size test set $1/n$ of design set.

Size training set is (n - 1)/n of design set.

Train and test n test times. Average errors. (Good choice: n = 10)

All objects are tested ones → most reliable test result that is possible.

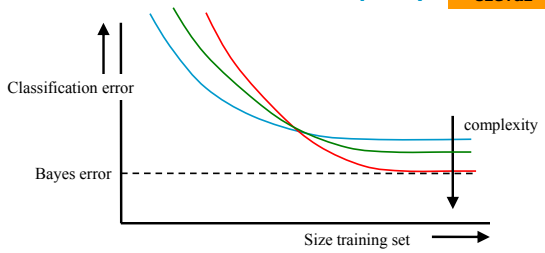
Final classifier: trained by all objects → best possible classifier.

Error estimate is slightly pessimistically biased.

TUDelft

Learning Curves for Different Classifier Complexity

`cleval`

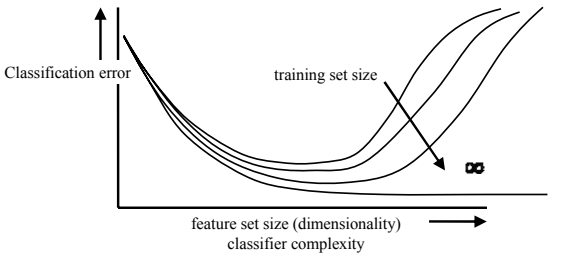


More complex classifiers are better in case of large training sets
and worse in case of small training sets

TUDelft

Peaking Phenomenon, Overtraining Curse of Dimensionality, Rao's Paradox

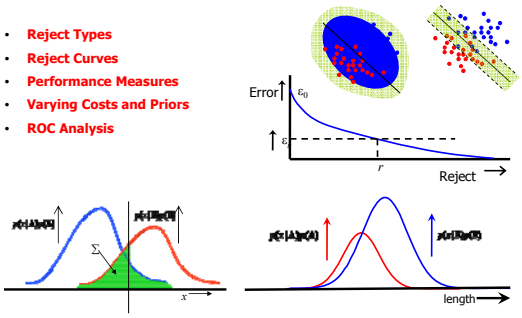
`clevalf`



TUDelft

Reject and ROC Analysis

- **Reject Types**
- **Reject Curves**
- **Performance Measures**
- **Varying Costs and Priors**
- **ROC Analysis**



TUDelft

How much to reject?

`reject` `plote`

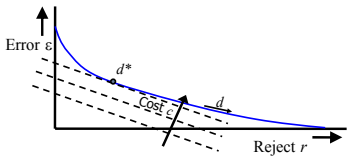
Compare the cost of a rejected object, c_r, with the cost of a classification error, c_ε:

$$c = c_r P(\text{reject}) + c_\varepsilon P(\text{error})$$

$$c = c_r r + c_\varepsilon \varepsilon$$

For given total cost c this is a linear function in the (r,ε) space.

Shift it until a possible operating point is reached.

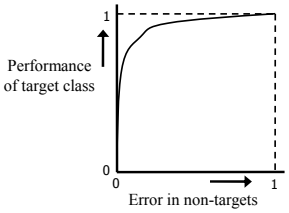


TUDelft

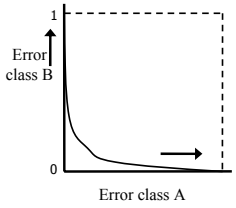
ROC Analysis

`roc` `plote`

ROC: Receiver-Operator Characteristic (from communication theory)



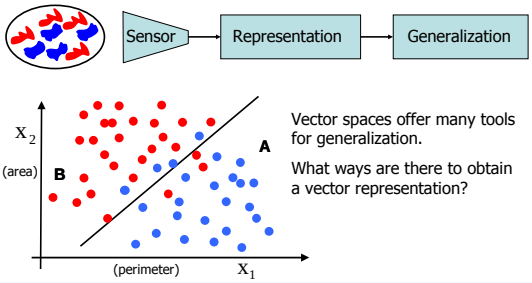
Medical diagnostics
Database retrieval



2-class pattern recognition

TUDelft

Pattern Representations in Vector Spaces

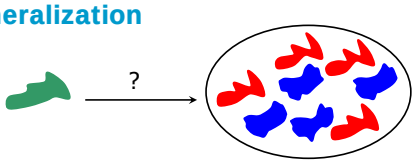


Vector spaces offer many tools for generalization.

What ways are there to obtain a vector representation?

TUDelft

Generalization

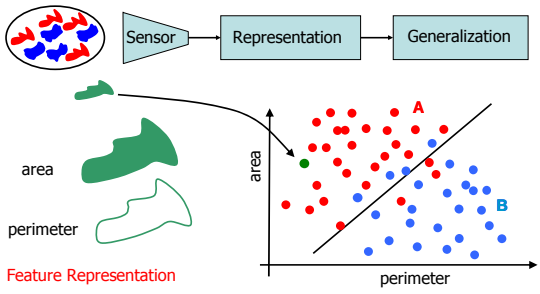


Generalization is the ability to make statements on unknown properties of new objects (e.g. their class) using a set of examples.

This can be done on the basis of:
- distances or (dis)similarities
- probabilities
- domains, decision functions

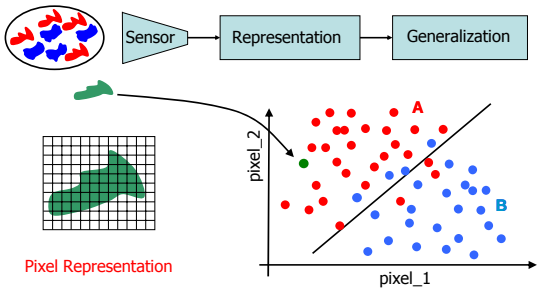
TUDelft

Pattern Recognition System



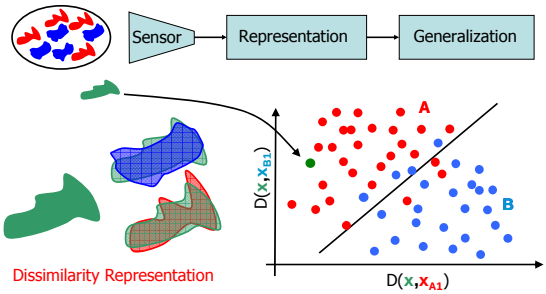
TUDelft

Pattern Recognition System



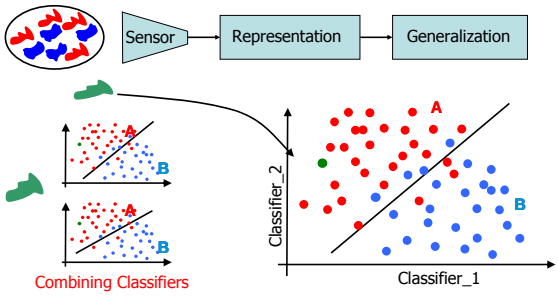
TUDelft

Pattern Recognition System



TUDelft

Pattern Recognition System

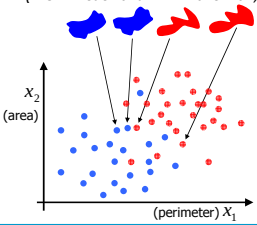


TUDelft

Compactness

Representations of real world similar objects are close.
There is no ground for any generalization (induction) on representations that do not obey this demand.

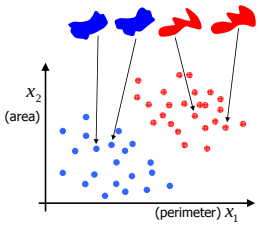
(A.G. Arkedev and E.M. Braverman, Computers and Pattern Recognition, 1966.)



The compactness hypothesis is not sufficient for perfect classification as dissimilar objects may be close.
→ class overlap
→ probabilities

TUDelft

True Representations



Similar objects are close
and
Dissimilar objects are distant.

→ no probabilities needed, domains are sufficient!

TUDelft

Demands for a Representation

The representation should enable generalisation:

Computation of:
dissimilarities, probabilities, domains, decision functions

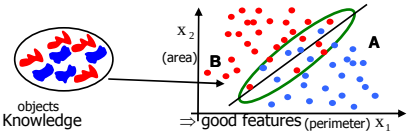
Measuring objects ⇒ numbers
Compactness ⇐ e.g. continuity

TUDelft

Feature Representation

Features are Based on Knowledge

Training set → Representation → Generalization



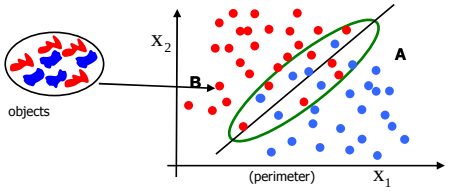
Knowledge ⇒ good features
⇒ (almost) separable classes

Lack of knowledge ⇒ (too many) bad features
⇒ hardly separable classes

Many features: ~ Lack of knowledge

TUDelft

Features Reduce



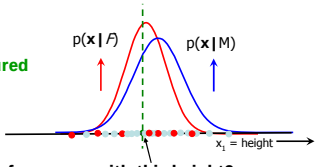
Due to reduction essentially different objects are represented identically.

→ The feature space representation needs a statistical, probabilistic generalization

TUDelft

Probabilistic Generalization

x = height measured



What is the gender of a person with this height?

Best guess is to choose the most 'probable' class (→ small error).

⇒ Good for overlapping classes.

⇒ Assumes the existence of a probabilistic class distribution and a representative set of examples.

TUDelft

Pixel Representation

Measuring Human Relevant Information

A

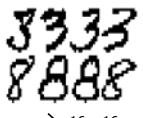
B

Nearest neighbours sorted:

B A A A B A A B A B

TUDelft

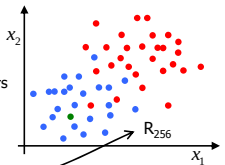
Pixel Representation



16 x 16

Pixels

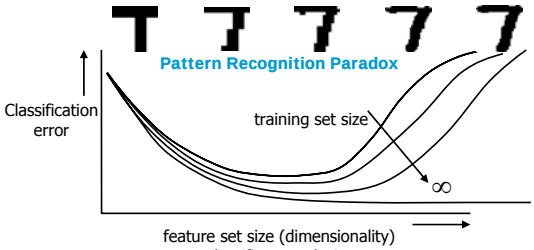
Features
Shapes
Moments
Fourier descriptors
Faces
Morphology



Pixels are more general, initially complete representation
Large datasets are available → good results for OCR

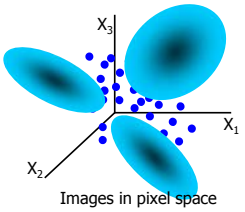
TUDelft

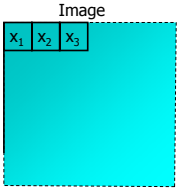
Peaking Phenomenon, Overtraining Curse of Dimensionality, Rao's Paradox



TUDelft

The Connectivity Problem in the Pixel Representation

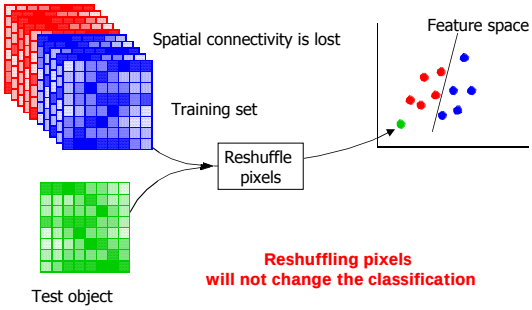




Dependent (connected) measurements are represented independently.
The dependency has to be refound from the data.

TUDelft

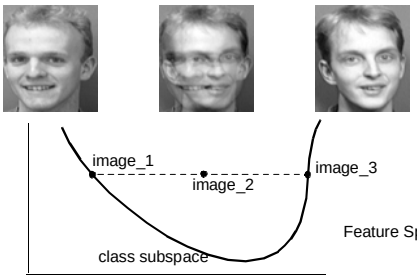
The Connectivity Problem in the Pixel Representation



Reshuffling pixels will not change the classification

TUDelft

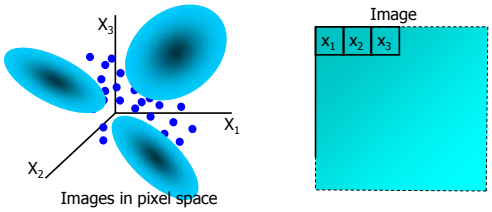
The Connectivity Problem in the Pixel Representation



Interpolation does not yield valid objects

TUDelft

The Connectivity Problem in the Pixel Representation



Dependent (connected) measurements are represented independently.
The dependency has to be refound from the data.

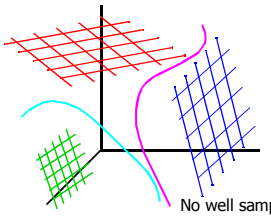
TUDelft

Reasons for selecting a pixel (sample) representation

- No good features can be found
- Sufficient training samples are available
- Direct, fast classification of the image (linear classifier == convolution)

TUDelft

Domains instead of Densities



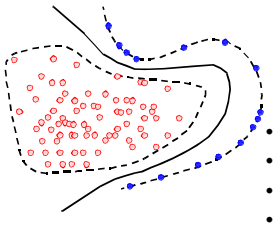
No well sampled training sets are needed.

Statistical classifiers have still to be developed.

Class structure ←→ Object invariants

TUDelft

Domain-based Classification



- Do not trust class densities.
- Estimate for each class a domain.
- Outlier sensitive.
- Distances instead of densities

How to construct domain based classifiers?

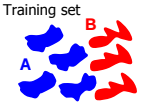
TUDelft

Dissimilarity Representation

Dissimilarity Representation

Not used by NN Rule

Training set



Dissimilarities d_{ij} between all training objects

$$D_T = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} & d_{47} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} & d_{67} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & d_{77} \end{pmatrix}$$

Unlabeled object **x** to be classified

$$d_x = (d_{x1}\ d_{x2}\ d_{x3}\ d_{x4}\ d_{x5}\ d_{x6}\ d_{x7})$$

The traditional Nearest Neighbor rule (template matching) finds:
$\text{label}(\text{argmin}_{\text{trainset}}\{d_{xi}\})$,
without using D_T. Can we do any better?

TU Delft

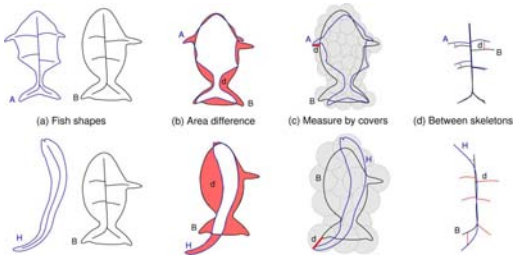
Dissimilarities – Possible Assumptions

Metric

1. Positivity: $d_{ij} \geq 0$
2. Reflexivity: $d_{ii} = 0$
3. Definiteness: $d_{ij} = 0$ objects i and j are identical
4. Symmetry: $d_{ij} = d_{ji}$
5. Triangle inequality: $d_{ij} < d_{ik} + d_{kj}$
6. Compactness: if the objects i and j are very similar then $d_{ij} < \delta$.
7. True representation: if $d_{ij} < \delta$ then the objects i and j are very similar.
8. Continuity of d.

TU Delft

Examples Dissimilarity Measures (1)

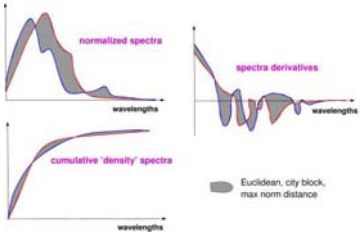


The measure should be descriptive. If there is no preference, a number of measures can be combined.

TU Delft

Examples Dissimilarity Measures (2)

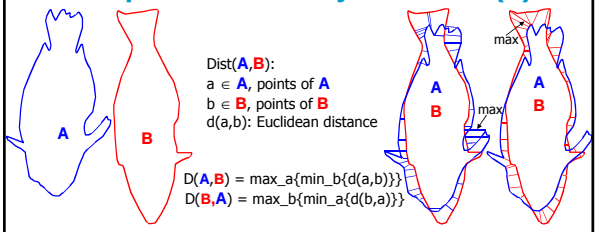
Comparison of spectra: some examples



In real applications, the dissimilarity measure should be robust to noise and small aberrations in the (raw) measurements.

TU Delft

Examples Dissimilarity Measures (3)



Dist(**A**,**B**):
a ∈ **A**, points of **A**
b ∈ **B**, points of **B**
d(a,b): Euclidean distance

D(**A**,**B**) = max_a{min_b{d(a,b)}}
D(**B**,**A**) = max_b{min_a{d(b,a)}}

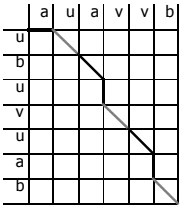
Hausdorff Distance (metric):
DH = max{max_a{min_b{d(a,b)}} , max_b{min_a{d(b,a)}}}

D(**A**,**B**) ≠ D(**B**,**A**)

Modified Hausdorff Distance (non-metric):
DM = max{mean_a{min_b{d(a,b)}},mean_b{min_a{d(b,a)}}}

TU Delft

Examples Dissimilarity Measures (4)



$X = (x_1, x_2, \ldots, x_k)$ $Y = (y_1, y_2, \ldots, y_n)$

$D_E(X,Y)$: Σ edit operations $X \Rightarrow Y$ (insertions, deletions, substitutions)

DE(snert ,meer) = 3:
snert ⇒ seert ⇒ seer ⇒ meer

DE(ner ,meer) = 2:
ner ⇒ mer ⇒ meer

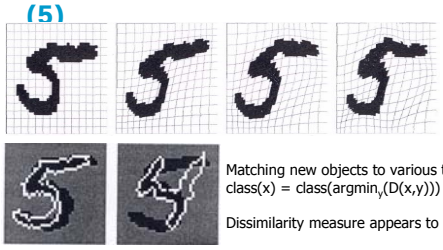
Possibly weighted

Triangle inequality ⇒ computationally feasible

D(aa,bb) < D(abcdef,bcdd)

TUDelft

Examples Dissimilarity Measures (5)



Matching new objects to various templates:
class(x) = class(argmin$_y$(D(x,y)))

Dissimilarity measure appears to be non-metric.

A.K. Jain, D. Zongker, Representation and recognition of handwritten digit using deformable templates, IEEE-PAMI, vol. 19, no. 12, 1997, 1386-1391.

TUDelft

Prospect of Dissimilarity based Representations: Zero

Let us assume that we deal with true representations:
$d_{ab} < \delta$ if and only if the objects a and b are very similar.

If δ is sufficiently small then a and b belong to the same class, as b is just a minor distortion of a (assuming true representations).

However, as Prob(b) > 0, there will be such an object for sufficiently large training sets ⇒ zero classification error possible!

⇒ Dissimilarity representation can be a true representation

TUDelft

Why a Dissimilarity Representation?

Many (exotic) dissimilarity measures are used in pattern recognition
- they may solve the connectivity problem (e.g. pixel based features)
- they may offer a way to integrate structural and statistical approaches
  e.g. by graph distances.

Prospect of zero-error classifiers by avoiding class overlap

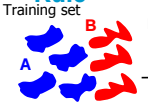
Better rules than the nearest neighbor classifier appear possible (more accurate, faster)

TUDelft

Classification of Dissimilarity Data

Alternatives for the Nearest Neighbor Rule

Training set



Dissimilarities d_{ij} between all training objects

$$D_T = \begin{pmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} \\ d_{31} & d_{32} & d_{33} & d_{34} & d_{35} & d_{36} & d_{37} \\ d_{41} & d_{42} & d_{43} & d_{44} & d_{45} & d_{46} & d_{47} \\ d_{51} & d_{52} & d_{53} & d_{54} & d_{55} & d_{56} & d_{57} \\ d_{61} & d_{62} & d_{63} & d_{64} & d_{65} & d_{66} & d_{67} \\ d_{71} & d_{72} & d_{73} & d_{74} & d_{75} & d_{76} & d_{77} \end{pmatrix}$$

Unlabeled object **x** to be classified

$$d_x = (d_{x1}\ d_{x2}\ d_{x3}\ d_{x4}\ d_{x5}\ d_{x6}\ d_{x7})$$

1. Dissimilarity Space
2. Embedding

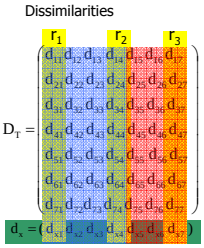
Pekalska, The dissimilarity representation for PR. World Scientific, 2005.

TUDelft

The Dissimilarity Space

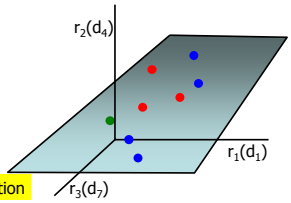
Alternative 1: Dissimilarity Space

Dissimilarities



Given labeled training set

Unlabeled object to be classified



Selection of 3 objects for representation

TUDelft

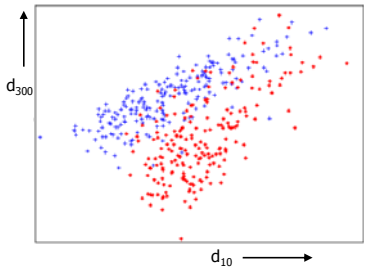
Example Dissimilarity Space: NIST Digits 3 and 8

Example of raw data

TUDelft

Example Dissimilarity Space: NIST Digits 3 and 8

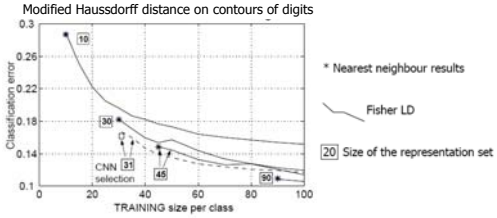
NIST digits: Hamming distances of 2 x 200 digits



TUDelft

Dissimilarity Space Classification ←→ Nearest Neighbor Rule

Modified Haussdorff distance on contours of digits



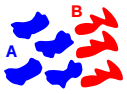
Dissimilarity based classification outperforms the nearest neighbor rule.

TUDelft

Embedding of non-Euclidean Dissimilarities

Embedding of (non-Euclidean) Dissimilarities

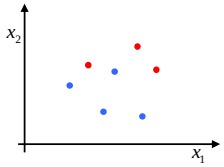
Embedding



Training set

→ Dissimilarity matrix D → X

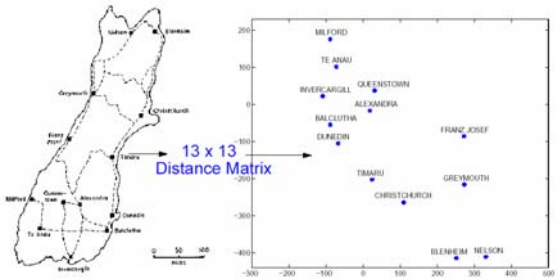
Is there a feature space for which Dist(X,X) = D ?



Position points in a vector space such that their Euclidean distances → D

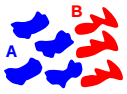
TUDelft

Embedding

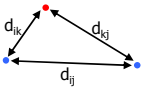


TUDelft

Embedding of non-metric measurements



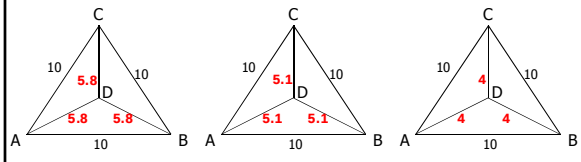
→ Dissimilarity matrix D → X



If the dissimilarity matrix cannot be explained from a vector space,
(e.g. for Hausdorff and Hamming distance of images)
or if $d_{ij} > d_{ik} + d_{kj}$ (triangle inequality not satisfied)
embedding in Euclidean space not possible
→ Pseudo-Euclidean embedding

TUDelft

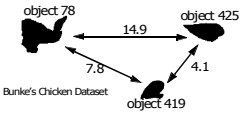
Euclidean - Non Euclidean - Non Metric



TUDelft

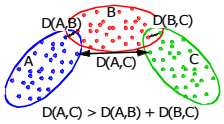
Non-metric distances

Weighted-edit distance for strings

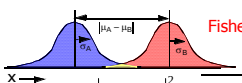


Bunke's Chicken Dataset

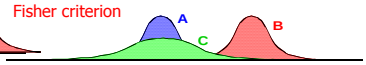
Single-linkage clustering



D(A,C) > D(A,B) + D(B,C)



Fisher criterion



$$J(A,B) = \frac{|\mu_A - \mu_B|^2}{\sigma_A^2 + \sigma_B^2}$$

$J(A,C) = 0 \quad J(A,B) = \text{large}$

$J(C,B) = \text{small} \neq J(A,B)$

TUDelft

(Pseudo-)Euclidean Embedding

$m \times m$ D is a given, imperfect dissimilarity matrix of training objects.

Construct inner-product matrix: $B = -\frac{1}{2} J D^{(2)} J \quad J = I - \frac{1}{m}\mathbf{1}\mathbf{1}$

Eigenvalue Decomposition , $B = Q \Lambda Q^T$

Select k eigenvectors: $X = Q_k \Lambda_k^{\frac{1}{2}}$ (problem: $\Lambda_k < 0$)

Let $\Im_k$ be a k x k diag. matrix, $\Im_k(i,i) = \text{sign}(\Lambda_k(i,i))$

$\Lambda_k(i,i) < 0 \rightarrow$ Pseudo-Euclidean

$n \times m$ D_z is the dissimilarity matrix between new objects and the training set.

The inner-product matrix: $B_z = -\frac{1}{2}(D_z^{(2)} J - \frac{1}{n}\mathbf{1}\mathbf{1}^T D^{(2)} J)$

The embedded objects: $Z = B_z Q_k |\Lambda_k|^{-\frac{1}{2}} \Im_k$

TU Delft

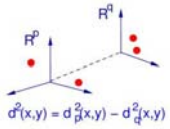
PES: Pseudo-Euclidean Space (Krein Space)

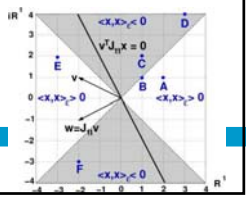
If D is non-Euclidean, B has p positive and q negative eigenvalues.

A pseudo-Euclidean space $\mathcal{E}$ with signature (p,q), k =p+q, is a non-degenerate inner product space $\mathfrak{R}_k = \mathfrak{R}_p \oplus \mathfrak{R}_q$ such that:

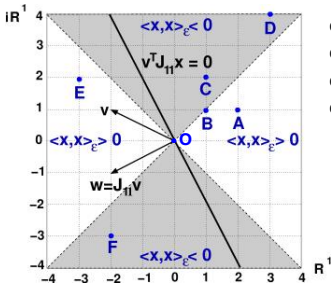
$$\langle x, y \rangle_\varepsilon = x^T \Im_{pq} y = \sum_{i=1}^{p} x_i y_i - \sum_{j=p+1}^{q} x_j y_j \qquad \Im_{pq} = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix}$$

$$d_\varepsilon^2(x,y) = \langle x-y, x-y \rangle_\varepsilon = d_p^2(x,y) - d_q^2(x,y)$$





Distances in PES



$d^2(O,A) > 0$
$d^2(O,E) > 0$
$d^2(O,B) = 0$
$d^2(O,D) < 0$

All points in the grey area are closer to O than O itself !?

Any point has a negative square distance to some points on the line $v^T Jx=0$.

Can it be used as a classifier? Can we define a margin as in the SVM?

TU Delft

PE Space ←→ Kernels

$K(x,y) = -\frac{1}{2} J D(x,y)^{(2)} J \quad J = I - \frac{1}{m}\mathbf{1}\mathbf{1}$ may by considered as a kernel.

If

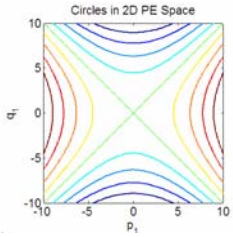
$$K(x,y) = < L(x), L(y) >$$

- The **kernel trick** may be used: operations defined on inner products in kernel space can be operated directly on K(x,y) **without embedding**!
- True for **Mercer kernels** (all eigenvalues ≥ 0).
- Difficult for **indefinite kernels**.
- Studying classifiers in **PE space** is studying the indefinite **kernel space**.
- Dissimilarities are more informative than kernels (due to normalization).

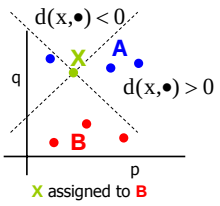
TU Delft

Classifiers in Pseudo-Euclidean Space

Distance based classifiers in PE Space



Metric in PE Space.
Equidistant points to the origin.



X assigned to B

Nearest Neighbour and **Nearest Mean** can be properly defined.
SVM ? What is the distance to a line?

TU Delft

SVM in PE Space

- SVM on indefinite kernels may not converge as Mercer's conditions are not fulfilled.
- However, if it converges the solution is proper:

  $|w^T \Im w|$

  is minimized.
- *See also: B. Haasdonk, Feature Space Interpretation of SVMs with Indefinite Kernels, IEEE PAMI, 24, 482-492, 2005.*

TUDelft

Densities in PE Space

- Densities can be defined in a vector space on de basis of volumes, without the need of a metric.
- Density estimates however, often need a metric.
  E.g. the Parzen estimator:

  $$\hat{f}(x) = \frac{1}{n}\sum_{y_i} c \exp\left(-\frac{d(x, y_i)^2}{2h^2}\right)$$

  needs a distance definition $d(x,y)$.
- There is no problem, however, in case for all objects $d(x,y) > 0$.
- How can Gaussian densities be defined?
- Note that QDA in PES is identical to the QDA in AES as the signature cancels. The relation with a Gaussian distribution, however, is lost.

TUDelft

Dissimilarity based classifiers compared

Prototype Selection

Assume D(T,R) are the distances between a training set T and a representation set R.
A classifier can be trained:

- on the distances directly
- in dissimilarity spaces
- in embedded spaces defined by D(R,R)

Selection of prototypes $R \subset T$:

- Random
- k-centres, mode seeking or some clustering procedure
- Feature selection methods
- Editting and condensing methods
- Sparse linear programming methods (L_1-norm SVM)

TUDelft

Dissimilarity based classification procedured compared

Training set 

→ Dissimilarity matrix D

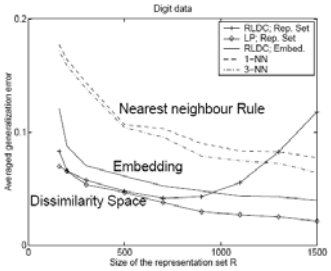
Test object x  → Dissimilarities d_x with training set

1. Nearest Neighbour Rule
2. Reduce training set to representation set
   ⇒ dissimilarity space
3. Embedding:Select large $\Lambda_{ii} > 0 \Rightarrow$ Euclidean space
   Select large $|\Lambda_{ii}| > 0 \rightarrow$ pseudo-Euclidean space

} discriminant function

TUDelft

Three Approaches Compared for the Zongker Data



Dissimilarity Space equivalent to Embedding better than Nearest Neighbour Rule

TUDelft

Polygon Data

Convex Pentagons

Heptagons

no class overlap
zero error

Minimum edge length: 0.1 of maximum edge length

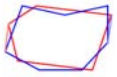
Distance measures: Hausdorff $D = \max \{ \max_i(\min_j(d_{ij})) , \max_j(\min_i(d_{ij})) \}$.

Modified Hausdorff $D = \max \{ \text{mean}_i(\min_j(d_{ij})), \text{mean}_j(\min_i(d_{ij})) \}$. (no metric!)

d_{ij} = distance between vertex i of polygon_1 and vertex j of polygon_2.

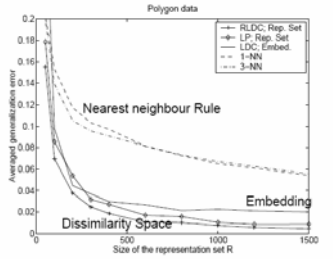
Polygons are scaled and centered.

Find the largest of the smallest vertex distances



TUDelft

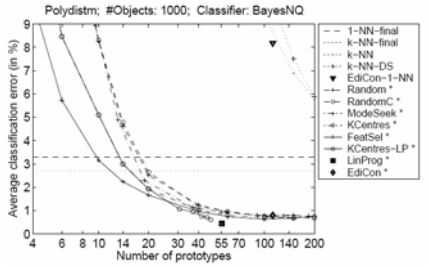
Dissimilarity Based Classification of Polygons



Zero error difficult to reach!

TUDelft

Prototype Selection: Polygon Dataset



The classification performance of the quadratic Bayes Normal classifier and the k-NN in dissimilarity spaces and the direct k-NN, as a function of the number of selected prototypes. Note that for 10-20 prototypes already better results are obtained than by using 1000 objects in the NN rules.

An Analysis of Causes of Non-Metric Data

Lack of information

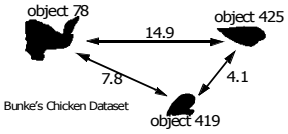
1800:
Crossing the Jostedalsbreen was impossible.
Travelling around (200 km) lasted 5 days.
Untill the shared point X was found.
People could visit each other in 8 hours.

D(V,J) = 5 days
D(V,X) = 4 hours
D(X,J) = 4 hours

TUDelft

Computational Problems

Large distances are overestimated due to computational problems

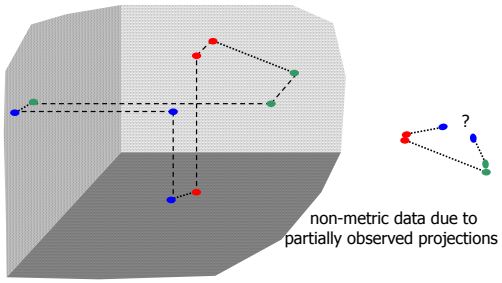


Weighted edit distance for strings

TUDelft

Projections - Occlusions

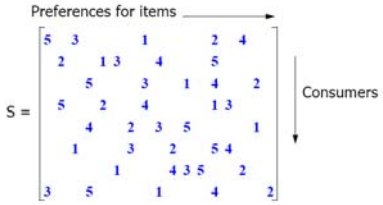
Small distances are underestimated



non-metric data due to
partially observed projections

TUDelft

Projections - Occlusions



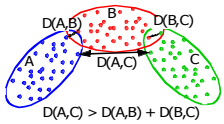
Example: consumer preferences for recommendation systems

TUDelft

Intrinsicly Different Dissimilarity Measures

Distance(Table,Book) = 0
Distance(Table,Cup) = 0
Distance(Book,Cup) = 1

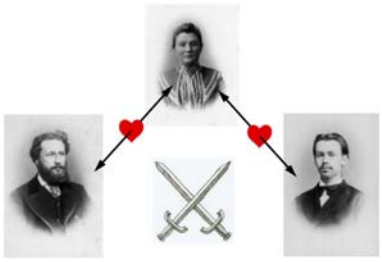
Single-linkage clustering



D(A,C) > D(A,B) + D(B,C)

TUDelft

Intrinsicly Different Dissimilarity Measures



Non-Euclidean human relations

TUDelft

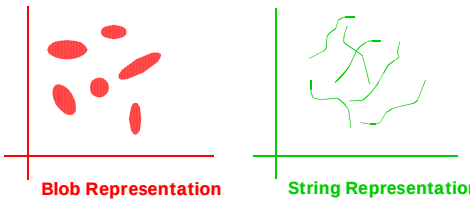
Causes of Non-Metric Dissimilarities

- Overestimated large distances (too difficult to compute)
- Underestimated small distances (one-sided view of objects) caused by the construction of complicated measures, needed to correspond with human observations.
- Essential non-metric distance definitions
  as the human concept of distance differs from the mathematical one.

TUDelft

Perspective

Human defined patterns do not fit in a Euclidean space.
Objects cannot be represented by points as they have an inner life.
Respect it.



Blob Representation **String Representation**

TUDelft

Pattern Recognition: Steps in Science and Consciousness

Tutorial
Barcelona, 7 July 2008

Robert P.W. Duin

Pattern Recognition Group
Delft University of Technology
The Netherlands

//ict.ewi.tudelft.nl/~duin

TUDelft

Paradigms in Pattern Recognition

- Paradigms: Popper versus Kuhn

- More features are good → peaking phenomenon
- Avoid multi-layer perceptrons → neural networks
- Peaking phenomenon → structural risk minimization
- Bridging the semantic gap

TUDelft

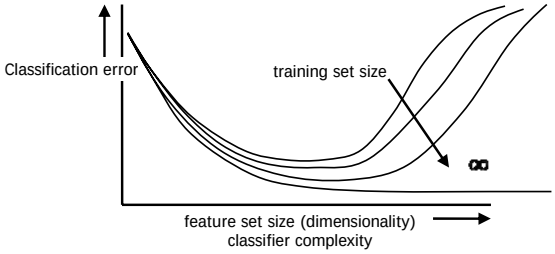
Popper versus Kuhn

Karl Popper:
We generate a **conjecture** and
try to **refute** it by an observation.

Thomas Kuhn:
Theories and counter examples can coincide
for a long time until by a **paradigm shift**
a new theory is accepted.

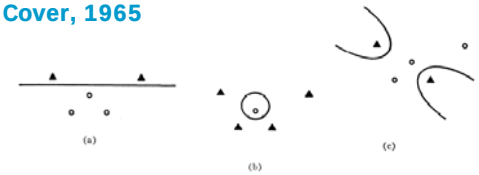
TUDelft

The peaking phenomenon



TUDelft

Cover, 1965



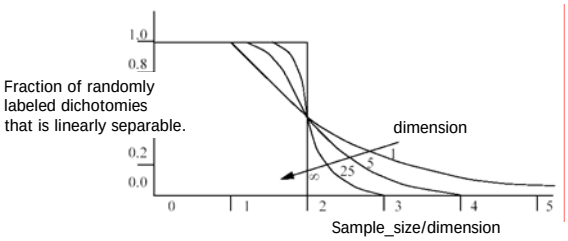
Arbitrary labeled sets of 5 points in 2D can always be separated by a quadratic classifier.

So a quadratic classifier does not generalize well for 5 points in 2D.

T.M. Cover, Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition, IEEE Trans. on Electr. Comp., June 1965, 326-334.

TUDelft

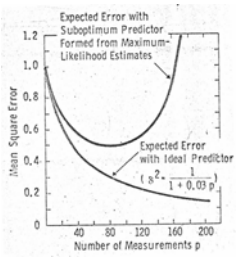
Cover, 1965, Theory on sample size / feature size

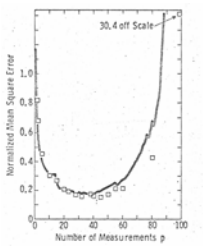


→Choose sample size > 2 x dimension

TUDelft

Allais, 1966: Theory and experiments

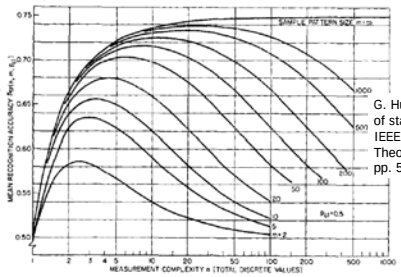




D.C. Allais, The problem of too many measurements in pattern recognition and prediction, IEEE Int. Con. Rec., vol. Part 7, March 1966, 124-130.

TUDelft

Hughes' Phenomenon (1968)



G. Hughes, On the mean accuracy of statistical pattern recognizers IEEE Transactions on Information Theory, vol. 14, no. 1, Jan 1968, pp. 55 - 63

number of measurement cells

TUDelft

Paradox solved ... , but the wrong one.

On the Peaking of the Hughes Mean Recognition Accuracy: The Resolution of an Apparent Paradox

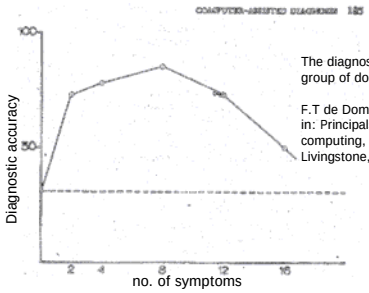
JAN M. VAN CAMPENHOUT

Larger numbers of measurement cells have different priors than smaller numbers: Hughes' study does not relate to a unique well-defined problem.

Still: peaking exists, but is not explained by Hughes.

TUDelft

Peaking in a study of human recognizers



The diagnostic classification accuracy of a group of doctors. Sample size is 100.

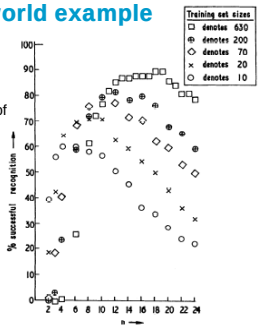
F.T de Dombal, Computer-assisted diagnosis, in: Principals and practice of medical computing, Whitby and Lutz (eds.), Churchill Livingstone, London, 1971, pp 179 - 199.

TUDelft

Ullmann 1969, real world example

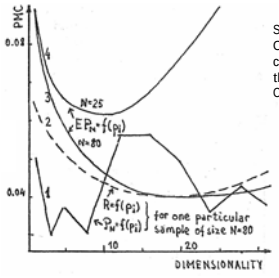
Character recognition classification performance as a function of the number of n-tuples used.

J.R. Ullmann,
Experiments with the n-tuple method of pattern recognition,
IEEE Trans. on Comp., 1969, 1135-1136



TUDelft

Raudys 1976, simulation study

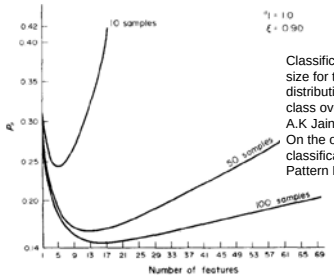


S. Raudys,
On dimensionality, learning, sample size and complexity of classification algorithms, Proc. of the 3rd Int. Conf. on Pattern Recognition, Coronado, California, 1976, 166-169.

1. single example
2. generalization of 1
3. expectation of 1
4. less samples

TUDelft

Jain and Waller, 1978, simulation study



Classification error as a function of the feature size for two overlapping Gaussian distributions. Higher features have increasing class overlap.
A.K Jain and W.G. Waller,
On the optimal number of features in the classification of multivariate Gaussian data,
Pattern Recognition, 10, pp 365 - 374, 1978

TUDelft

Foley, 1972, Rule of thumb on sample size / feature size

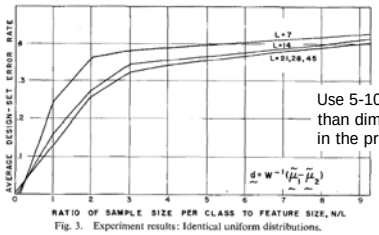


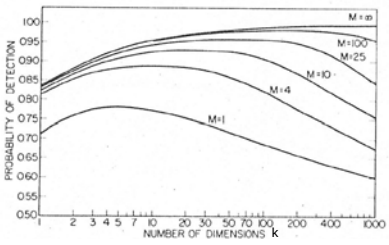
Fig. 3. Experiment results: Identical uniform distributions.

Use 5-10 times more samples than dimensions (parameters?) in the problem.

Resubstitution error (apparent error) for two identical uniform distributions as a function of sample size divided by feature size for a set of feature sizes.
D.H. Foley, Considerations of sample and feature size,
IEEE Trans on Information Theory, 18, 1972, pp 618 - 626.

TUDelft

Trunk's example



$A: N(\mu, I)$

$B: N(-\mu, I)$

$\mu_i = \frac{1}{\sqrt{k}}$

G.V. Trunk, A Problem of Dimensionality: A Simple Example,
IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 1, pp. 306-307, 1979

TUDelft

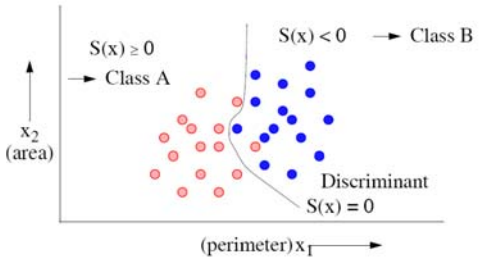
The peaking phenomenon

1978: Accepted and 'understood':

Use 5-10 times as many samples as features

TUDelft

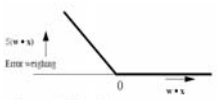
Multi-layer classifiers



How to construct arbitrary, non-linear classifiers?

TUDelft

Perceptron

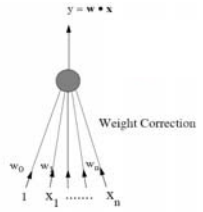


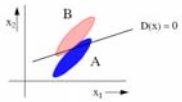
Sequential Learning

Criterion: $J(\mathbf{w}) = \sum_{i \in L} S(\mathbf{w} \bullet \mathbf{x}_i)$

Weights are changed after testing each learning object.

$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}(\mathbf{w}_n, \mathbf{x}_i)$ for $i \in L$, repeatedly.



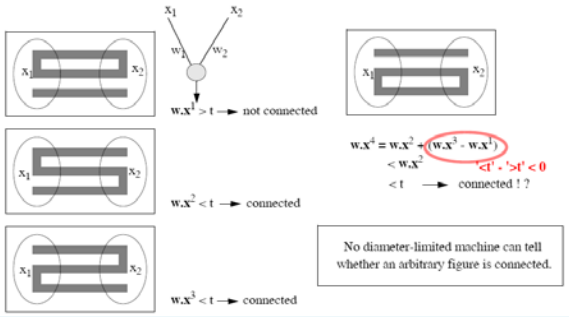


y = 1 if D(x) ≥ 0
y = 0 if D(x) < 0

A discriminant is a binary feature
A set of discriminants is a set of binary features

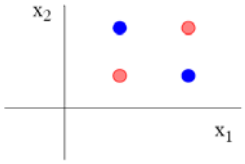
TUDelft

Minsky's and Papert's Theorem (1969)



TUDelft

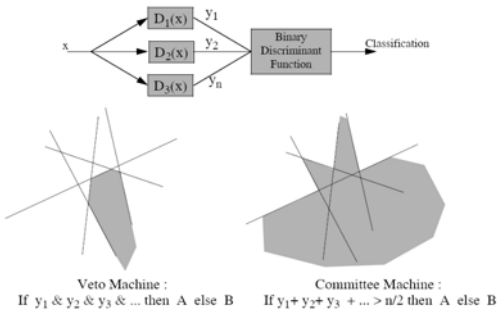
The XOR problem



A solution for the XOR problem is needed,
but cannot be reached by a simple perceptron

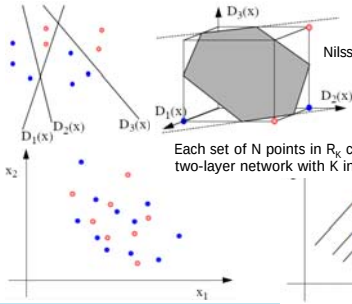
TUDelft

The Multi-Layer Perceptron



TUDelft

Zero-Error Multi-Layer Perceptron

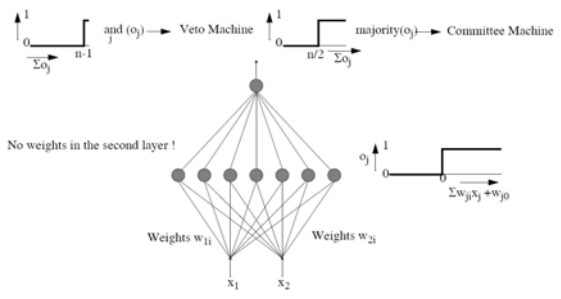


Nilsson, Learning Machines, 1965

Each set of N points in R_K can be separated by a two-layer network with K inputs and N-K+1 hidden units.

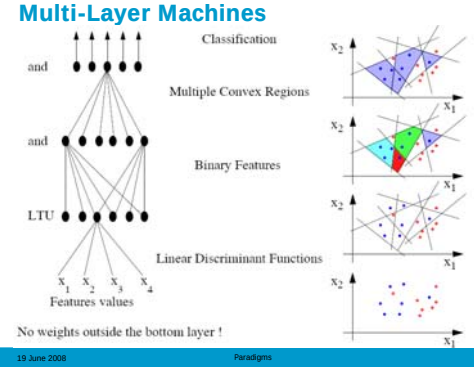
TUDelft

Two - Layer Machines



TUDelft

Multi-Layer Machines



TUDelft

Preliminary (1970) Conclusions on Multi-Layer Perceptrons

- Linear perceptrons are not powerful enough
- Large two-layer networks can make any separation but don't generalize
- Multi-layer networks are very difficult to train.

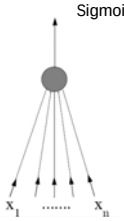
How to train a multi-stage classifier?

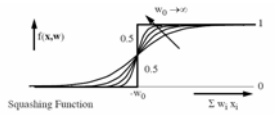
TU Delft

Artificial Neuron

$$\text{output} = f\left(\sum w_i x_i + w_0\right) = \frac{1}{1+e^{-\mathbf{w}\bullet\mathbf{x}+w_0}}$$

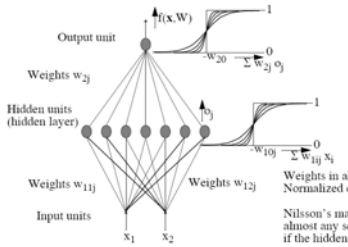
Sigmoid or logistic function





TU Delft

Neural Network



Weights in all layers
Normalized outputs (0,1) in all layers

Nilsson's machine is included, so almost any set of objects can be separated if the hidden layer is large enough.

More output units are possible
More hidden layers are possible.

TU Delft

Back-propagation training rule

Network: $f(\mathbf{x}_p, W)$

Training set: $\{(\mathbf{x}_1, t_1), (\mathbf{x}_2, t_2), \ldots (\mathbf{x}_n, t_n)\}$

Target values (labels): t_p

Network error: $E = \sum_p \{ t_p - f(\mathbf{x}_p, W) \}^2$

Gradient descent: $W \leftarrow W + \Delta W = W + \sum_p \Delta_p W$

Generalized delta rule: $\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}, \forall i,j$

Fixed stepsize: η

output units (layer k): $\delta_{pk} = (t_{pk} - o_{pk}) o_{pk} (1 - o_{pk})$

hidden units (layer j): $\delta_{pj} = o_{pj}(1 - o_{pj}) \sum_k \delta_{pk} w_{kj}$

The 'errors' in the lower layers (j) are computed using the corrections of the layers above (k): backpropagation.

TU Delft

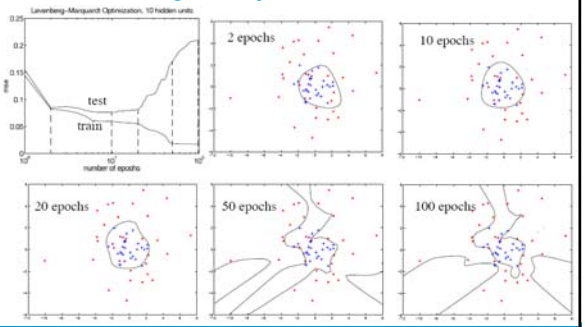
The art of training a neural network

Using a neural network classifier is not straightforward all:
- Architecture (numbers of hidden layers and hidden units)
- input representation
- output representation
- target values
- reject values
- initialization procedure
- batches, partitioned learning set (size?) or individual training
- adding noise (amount?) to input or weights?
- weight decay
- step size η
- momentum term α

"The backpropagation training procedure can be a user's nightmare"
(Weiss & Kulikowski)

TU Delft

Overtraining Example – 10 Hidden Units



TU Delft

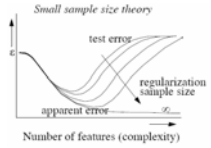
Early Neural Network Examples

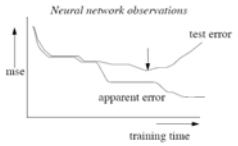
application	#weights	#samples	error	ref.
text -> speech	25000	5000	0.20	Sejnowski
sonar target rec	1105	192	0.15	Gorman
car control	>36000	1200	car drives on winding road	Pomerleau
back-gammon	>11000	3000	computer champion	Tesauro
sex rec from faces	>36000	90	0.09	Golomb
char rec	9900	5000	0.055	Sato
remote sensing	1800	50	0.05-0.10	Kamata
signature verif.	480	280	0.05	Sabourin

T.J. Sejnowski and C.R. Rosenberg, *NETtalk: a parallel network that learns to read aloud*, The John Hopkins University Electrical Eng. and Comp. Science, 1986.
P. Gorman and T.J. Sejnowski, *Learned Classification of Sonar Targets Using Massively Parallel Network*, IEEE Transactions on ASSP, vol. 36, no. 7, July 1988.
D. Pomerleau, *ALVINN: An Autonomous Land Vehicle in a Neural Network*, in: David S. Touretzky, Advances in Neural Information Processing Systems I, 1989
G. Tesauro, *Neurogammon wins computer olympiad*, Neural Computation, vol. 1, pp 312-323, 1990
B.A. Golomb, D.T. Lawrence, T.J. Sejnowski, Sexnet: *A neural network identifies sex from human faces*, Adv. in Neural Inf. Proc. Sys. 1, 1989
A. Sato, K. Yamada, J. Tsukumo, and T. Temma, *Neural network models for incremental learning*, ICNN, Helsinki, 1991.
S.-I. Kamata, R.O. Eason, A. Perez, and E. Kawaguchi, *A Neural Network Classifier for LANDSAT Image Data*, Proc. 11th ICPR, The Hague, Vol 2, 573-576, 1992
R. Sabourin and J-P. Drouhard, *Off-Line Signature Verification Using Directional PDF and Neural Networks*, Proc. 11th ICPR, The Hague, Vol 2, 321-325, 1992

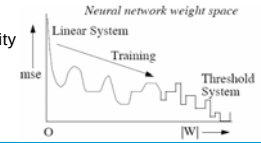
TUDelft

Understanding Neural Networks





Conclusion: the effective complexity of a neural network increases during training.



TUDelft

Neural Network Appreciation in Pattern Recognition

"Artificial Intelligence and Neural Networks have deceived and spoiled two generations of computer scientists just by these names"
(Azriel Rosenfeld, Oulu 1989)

"Neural Networks has brought new enthusiasm and spirit to the next generation of young researchers."
(Laveen Kanal, Den Haag 1992)

"Just a short look at the architecture of a Neural Network is sufficient to see that the thing simply doesn't have the moral right to show any reasonable performance"
(Leo Breiman, Edinburgh, 1995)

TUDelft

Neural Networks as Pattern Recognition Tools

Neural networks offer a way to implement moderately non-linear classifiers. They contain many parameters, so training should be done carefully, to avoid overtraining. Consequently, they offer a toolbox, which result depends on the implementation and/or analyst.

Neural networks are mainly applicable to complex (...) processing problems for which a good performance criterion can be given, but which cannot easily be divided into subproblems.
(Dick de Ridder, December 2001)

TUDelft

Understanding pattern recognition 1993

- Collect a-selectively examples.
- Get a feature set from an application expert.
  - Good expert: small set of good features.
  - Bad expert: large set of bad features.
    Collect more data.
- Reduce features as far as necessary.
- Find a good classifier by cross-validation.

or

- Train a neural network.

TUDelft

Neural Network Magic

"Neural networks can

[.... do something paradoxial ....]
{learn from a few examples,
learn a complicated task,
become conscious}

because they are so complex."

This is not true.
Let us study the most simple systems.

TUDelft

Roman architecture

TUDelft

Gothic architecture

How many examples do you need?

TUDelft

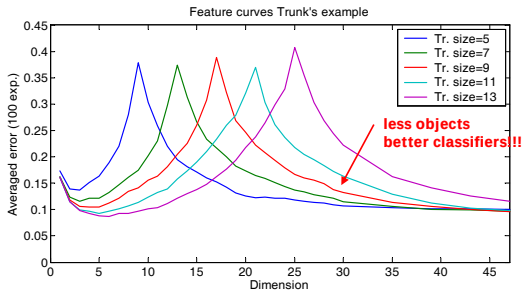
Classifiers in almost empty spaces

- Consider m objects in R_k, m < k
- Find the data subspace, R_{m-1}
- Compute a classifier in the subspace
- New objects: project and classify

- Let's try Fisher:

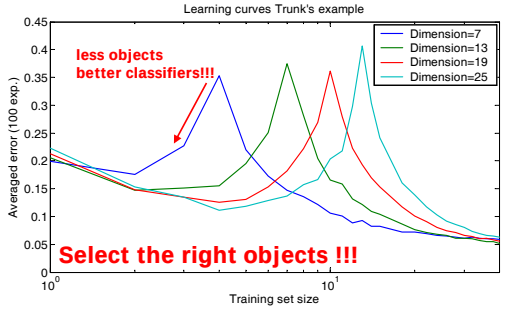
TUDelft

Feature curves for Fisher on Trunk's Example



TUDelft

Learning curves for Fisher on Trunk's Example



TUDelft

Classifiers in almost empty spaces

- Data in subspaces
- Classes in different subspaces
- Select the right objects
- Density based classifiers out

→Different classifiers
→Support Vector Classifier
Based on distances and data structure

TUDelft

More paradigms

- Regularization
- Kernels
- Indefinite representations
- Sparse classifiers
- Bridging the semantic gap

TU Delft

Shifting insights

Densities → Distances → Structure

TU Delft

Pattern Recognition: Steps in Science and Consciousness

Tutorial
Barcelona, 7 July 2008

Robert P.W. Duin

Pattern Recognition Group
Delft University of Technology
The Netherlands

//ict.ewi.tudelft.nl/~duin

TU Delft

Consciousness

- How do you know you are conscious?
- How do you know somebody else is conscious?
- How do you show you are conscious?
- How do you know something?
- How do you recognize pattern?

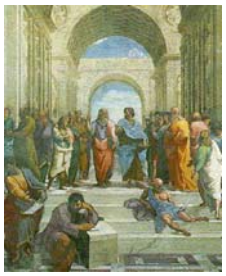
TU Delft

Ways to Prove some Truth

- By Authority
- By Demonstration
- By Reasoning

TU Delft

Concepts and Observations



School of Athens

Plato and Aristotle

TU Delft

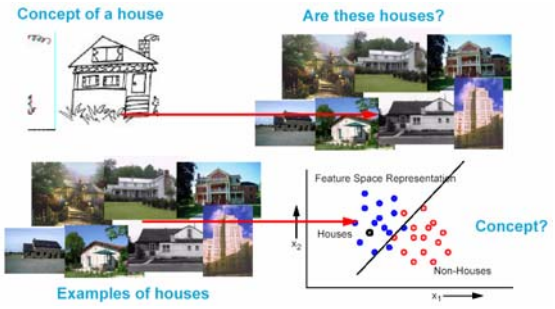
Concepts and Observations

Plato:
The **concepts live** somewhere **inside** us. We awake them by our observations. It is our **struggle to understand new observations** in terms of existing concepts.

Aristotle:
We learn the world by observing it. In this process we struggle with **creating concepts**, i.e. the generalizations of observations.

TU Delft

Structural versus Statistical Pattern Recognition



TU Delft

The Ways of Science
Reasoning and Observation

The Fields of Science
Introspection ←→ Extrospection

TUDelft

Occam versus Bayes

William of Occam (1284-1347)

One should not increase, beyond what is necessary, the number of entities required to explain anything. (Occam's razor): Take the most simple solution.

Rev. Thomas Bayes (1702-1761)

Take the solution that is based on a procedure which yields the best result on the average over all problems of interest.

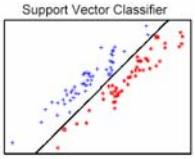
TUDelft

Vapnik versus Fisher

Chervonenkis
Vapnik:

Structural risk minimization

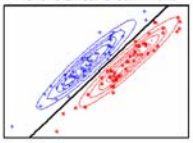
Support Vector Classifier



Fisher:

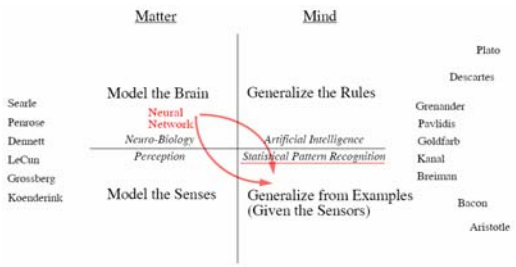
Mean square error minimization

Fisher's Linear Discriminant



TUDelft

Mind and Matter



TUDelft

Consciousness and Pattern Recognition

- *Consciousness* is an internal ability, (almost) not objectively detectable by physical means.
- *Pattern recognition* is an objective external ability, operating in the physical world.

TUDelft

Consciouness

- the ability to see meaningful structures in the world
- the ability to relate oneself (feel connected) to structures in the world
- the ability to shape and create new structures

TUDelft

Pattern Recognition

- the ability to detect structures in the physical world
- the ability to learn structures in the physical world
- the ability to develop a procedure for learning structures

TU Delft

Consciousness and pattern recognition ...

... two ides of the same coin

TU Delft

Structural Inference of Sensor-Based Measurements

Robert P.W. Duin[1] and Elżbieta Pękalska[1,2]

[1] ICT group, Faculty of Electr. Eng., Mathematics and Computer Science
Delft University of Technology, The Netherlands
r.duin@ieee.org, e.m.pekalska@tudelft.nl
[2] School of Computer Science, University of Manchester, United Kingdom

Abstract. Statistical inference of sensor-based measurements is intensively studied in pattern recognition. It is usually based on feature representations of the objects to be recognized. Such representations, however, neglect the object structure. Structural pattern recognition, on the contrary, focusses on encoding the object structure. As general procedures are still weakly developed, such object descriptions are often application dependent. This hampers the usage of a general learning approach.

This paper aims to summarize the problems and possibilities of general structural inference approaches for the family of sensor-based measurements: images, spectra and time signals, assuming a continuity between measurement samples. In particular it will be discussed when probabilistic assumptions are needed, leading to a statistically-based inference of the structure, and when a pure, non-probabilistic structural inference scheme may be possible.

1 Introduction

Our ability to recognize patterns is based on the capacity to generalize. We are able to judge new, yet unseen observations given our experience with the previous ones that are similar in one way or another. Automatic pattern recognition studies the ways which make this ability explicit. We thereby learn more about it, which is of pure scientific interest, and we construct systems that may partially take over our pattern recognition tasks in real life: reading documents, judging microscope images for medical diagnosis, identifying people or inspecting industrial production.

In this paper we will reconsider the basic principles of generalization, especially in relation with sensor measurements like images (e.g. taken from some video or CCD camera), time signals (e.g. sound registered by a microphone), and spectra and histograms (e.g. the infra-red spectrum of a point on earth measured from a satellite). These classes of measurements are of particular interest since they can very often replace the real object in case of human recognition: we can read a document, identify a person, recognize an object presented on a monitor screen as well as by a direct observation. So we deal here with registered signals which contain sufficient information to enable human recognition in an almost natural way. This is an entirely different approach to study the weather patterns

D.-Y. Yeung et al. (Eds.): SSPR&SPR 2006, LNCS 4109, pp. 41–55, 2006.

from a set of temperature and air pressure measurements than taken by a farmer who observes the clouds and the birds.

The interesting, common aspect of the above defined set of sensor measurements is that they have an observable structure, emerging from a relation between neighboring pixels or samples. In fact we do not perceive the pixel intensity values themselves, but we directly see a more global, meaningful structure. This structure, the unsampled continuous observation in space and/or time constitutes the basis of our recognition. Generalization is based on a direct observation of the similarity between the new and the previously observed structures.

There is an essential difference between human and automatic pattern recognition, which will be neglected here, as almost everywhere else. If a human observes a structure, he may directly relate this to a meaning (function or a concept). By assigning a word to it, the perceived structure is named, hence recognized. The word may be different in different languages. The meaning may be the same, but is richer than just the name as it makes a relation to the context (or other frame of reference) or the usage of the observed object. On the contrary, in automatic recognition it is often attempted to map the observations directly to class labels without recognizing the function or usage.

If we want to simulate or imitate the human ability of pattern recognition it should be based on object structures and the generalization based on similarities. This is entirely different from the most successful, mainline research in pattern recognition, which heavily relies on a feature-based description of objects instead of their structural representations. Moreover, generalization is also heavily based on statistics instead of similarities.

We will elaborate on this paradoxical situation and discuss fundamentally the possibilities of the structural approach to pattern recognition. This discussion is certainly not the first on this topic. In general, the science of pattern recognition has already been discussed for a long time, e.g. in a philosophical context by Sayre [1] or by Watanabe on several occasions, most extensively in his book on human and mechanical recognition [2]. The possibilities of a more structural approach to pattern recognition was one of the main concerns of Fu [3], but it was also clear that, thereby, the powerful tools of statistical approaches [4,5,6,7] should not be forgotten; see [8,9,10].

Learning from structural observations is the key question of the challenging and seminal research programme of Goldfarb [10,11,12]. He starts, however, from a given structural measurement, the result of a 'structural sensor' [13] and uses this to construct a very general, hierarchial and abstract structural description of objects and object classes in terms of primitives, the Evolving Transformation System (ETS) [11]. Goldfarb emphasizes that a good structural representation should be able to generate proper structures. We recognize that as a desirable, but very ambitious direction. Learning structures from examples in the ETS framework appears still to be very difficult, in spite of various attempts [14].

We think that starting from such a structural representation denies the quantitative character of the lowest level of senses and sensors. Thereby, we will again face the question how to come to structure, how to learn it from examples given

the numeric outcomes of a physical measurement process, that by its organization in time and space respects this structure. This question will not be solved here, as it is one of the most basic issues in science. However, we hope that a contribution is made towards the solution by our a summary of problems and possibilities in this area, presented from a specific point of view.

Our viewpoint, which will be explained in the next sections, is that the feature vector representation directly reduces the object representation. This causes a class overlap that can only be solved by a statistical approach. An indirectly reducing approach based on similarities between objects and proximities of their representations, may avoid, or at least postpone such a reduction. As a consequence, classes do not overlap intrinsically, by which a statistical class description can be avoided. A topological- or domain-based description of classes may become possible, in which the structural aspects of objects and object classes might be preserved. This discussion partially summarizes our previous work on the dissimilarity approach [15], proximities [16], open issues [17] and the science of pattern recognition [18].

2 Generalization Principles

The goal of pattern recognition may be phrased as the derivation of a general truth (e.g. the existence of a specified pattern) from a limited, not exhaustive set of examples. We may say that we thereby generalize from this set of examples, as the establishment of a general truth gives the possibility to derive non-observed properties of objects, similar to those of observed examples.

Another way to phrase the meaning of generalization is to state that the truth is *inferred* from the observations. Several types of inference can be distinguished:

Logical inference. This is the original meaning of inference: a truth is derived from some facts, by logical reasoning, e.g.

1. Socrates is a man.
2. All man are mortal.
3. Consequently, Socrates is mortal.

It is essential that the conclusion was derived before the death of Socrates. It was already known without having observed it.

Grammatical inference. This refers to the grammar of an artificial language of symbols, which describes the "sentences" that are permitted from a set of observed sequences of such symbols. Such grammars may be inferred from a set of examples.

Statistical inference. Like above, there are observations and a general, accepted or assumed, rule of a statistical (probabilistic) nature. When such a rule is applied to the observations, more becomes known than just the directly collected facts.

Structural inference. This is frequently used in the sense that structure is derived from observations and some general law. E.g. in some economical publications, "structural inference" deals with finding the structure of a statistical model (such as the set of dependencies) by statistical means [19]. On

the contrary, "structural inference" can also be understood as using structural (instead of statistical) properties to infer unobserved object properties.

Empirical inference. This term is frequently used by Vapnik, e.g. in his recent revised edition of the book on structural risk minimization [20]. It means that unnecessary statistical models are avoided if some value, parameter, or class membership has to be inferred from observational data. It is, however, still based on a statistical approach, in the sense that probabilities and expectations play a role. The specific approach of empirical inference avoids the estimation of statistical functions and models where possible: do not estimate an entire probability density function if just a decision is needed.

It should be noted that in logical, statistical and empirical inferences object properties are inferred by logical, statistical and empirical means, respectively. In the terms of "grammatical inference" and "structural inference", the adjective does not refer to the means but to the goal: finding a grammar or a structure. The means are in these cases usually either logical or statistical. Consequently, the basic tools for inference are primarily logic and statistics. They correspond to knowledge and observations. As logic cannot directly be applied to sensor data, statistical inference is the main way for generalization in this case.

We will discuss whether in addition to logic and statistics, also structure can be used as a basic means for inference. This would imply that given the structure of a set of objects and, for instance, the corresponding class labels, the class label of an unlabeled object can be inferred. As we want to learn from sensor data, this structure should not be defined by an expert, but should directly be given from the measurements, e.g. the chain code of an observed contour.

Consider the following example. A professor in archeology wants to teach a group of students the differences in the styles of A and B of some classical vases. He presents 20 examples for each style and asks the students to determine a rule. The first student observes that the vases in group A have either ears or are red, while those of group B may also have ears, but only if they are blue (a color that never occurs for A). Moreover, there is a single red vase in group B without ears, but with a sharp spout. In group A only some vases with ears have a spout. The rule he presents is: `if (ears` $\wedge$ `not_blue)` $\vee$ `(red` $\wedge$ `no_ears` $\wedge$ `no_spout) then A else B.` The second student measures the sizes (weight and height) of all vases, plots them on a 2D scatter plot and finds a straight line that separates the vases with just two errors. The third student manually inspects the vases from all sides and concludes that the lower part is ball-shaped in group A and egg-shaped in group B. His rule is thereby: `if ball-shaped then A, if egg-shaped then B.`

The professor asked the first student why he did not use characteristic paintings on the vases for their discrimination. The student answered that they were not needed as the groups could have perfectly been identified by the given properties. They may, however, be needed if more vases appear. So, this rule works for the given set of examples, but does it generalize?

The second solution did not seem attractive to the professor as some measurement equipment is needed and, moreover, two errors are made! The student

responded that these two errors showed in fact that his statistical approach was likely better than the logical approach of the first student, as it was more general (less overtrained). This remark was not appreciated by the professor: very strange to prove the quality of a solution by the fact that errors are made!

The third student seemed to have a suitable solution. Moreover, the shape property was in line with other characteristics of the related cultures. Although it was clear what was meant by the ball-ness and the egg-ness of the vase shapes, the question remained whether this could be decided by an arbitrary assistant. The student had a perfect answer. He drew the shapes of two vases, one from each group, on a glass window in front of the table with vases. To classify a given vase, he asked the professor to look through each of the two images to this vase and to walk to and from the window to adjust the size until a match occurs.

We hope that this example makes clear that logic, statistics and structure can be used to infer a property like a class label. Much more has to be explained about how to derive the above decision rules by automatic means. In this paper, we will skip the logical approach as it has little to do with the sensory data we are interested in.

3 Feature Representation

We will first shortly summarize the feature representation and some of its advantages and drawbacks. In particular, it will be argued how this representation necessarily demands a statistical approach. Hence, this has far reaching consequences concerning how learning data should be collected. Features are object properties that are suitable for their recognition. They are either directly measured or derived from the raw sensor data. The feature representation represents objects as vectors in a (Euclidean) feature space. Usually, but not always, the feature representation is based on a significant reduction. Real world objects cannot usually be reconstructed from their features. Some examples are:

- Pieces of fruit represented by their color, maximum length and weight.
- Handwritten digits represented by a small set of moments.
- Handwritten digits represented by the pixels (in fact, their intensities) in images showing the digits.

This last example is special. Using pixel values as features leads to pixel representations of the original digits that are reductions: minor digit details may not be captured by the given pixel resolution. If we treat, however, the digital picture of a digit as an object, the pixel representation is complete: it represents the object in its entirety. This is not strange as in handling mail and money transfers, data typists often have to recognize text presented on monitor screens. So the human recognition is based on the same data as used for the feature (pixels) representation.

Note that different objects may have identical representations, if they are mapped on the same vector in the feature space. This is possible if the feature representation reduces the information on objects, which is the main cause for class overlap, in which objects belonging to different classes are identically represented.

The most common and most natural way to solve the problem of class overlap is by using probability density functions. Objects in the overlap area are assigned to the class that is the most probable (or likely) for the observed feature vector. This not only leads to the fully Bayesian approaches, based on the estimation of class densities and using or estimating class prior probabilities, but also to procedures like decision trees, neural networks and support vector machines that use geometrical means to determine a decision boundary between classes such that some error criterion is minimized.

In order to find a probability density function in the feature space, or in order to estimate the expected classification performance for any decision function that is considered in the process of classifier design, a set of objects has to be available that is representative for the distribution of the future objects to be classified later by the final classifier. This last demand is very heavy. It requires that the designer of a pattern recognition system knows exactly the circumstances under which it will be applied. Moreover, he has to have the possibility to sample the objects to be classified. There are, however, many applications in which it is difficult or impossible. Even in the simple problem of handwritten digit recognition it may happen that writing habits change over time or are location dependent. In an application like the classification of geological data for mining purposes, one likes to learn from existing mining sites how to detect new ones. Class distributions, however, change heavily over the earth.

Another problem related to class overlap is that densities are difficult to estimate for more complete and, thereby, in some sense better representations, as they tend to use more features. Consequently, they have to be determined in high-dimensional vector spaces. Also the geometrical procedures suffer from this, as the geometrical variability in such spaces is larger. This results in the paradox of the feature representation: more complete feature representations need larger training sets or will deteriorate in performance [21].

There is a fundamental question of how to handle the statistical problem of overlapping classes in case no prior information is available about the possible class distributions. If there is no preference, the No-Free-Lunch-Theorem [22] states that all classifiers perform similarly to a random class assignment if we look over a set of problems on average. It is necessary to restrict the set of problems significantly, e.g. to compact problems in which similar objects have similar representations. It is, however, still an open issue how to do this [23]. As long as the set of pattern recognition problems is based on an unrealistic set, studies on the expected performance of pattern recognition systems will yield unrealistic results. An example is the Vapnik-Chervonenkis error bound based on the structural risk minimization [20]. Although a beautiful theoretical result is obtained, the prescribed training set sizes for obtaining a desired (test) performance are far from being realistic. The support vector machine (SVM), which is based on structural risk minimization, is a powerful classifier for relatively small training sets and classes that have a small overlap. As a general solution for overlapping classes, as they arise in the feature space, it is not suitable. We will point this out below.

We will now introduce the idea of domain-based classifiers [24]. They construct decision boundaries between classes that are described just by the domains they cover in the feature space (or in any representation space) and do not depend on (the estimates of) probability distributions. They are, thereby, insensitive to ill-sampled training sets, which may even be selectively sampled by an expert. Such classifiers may be beneficial for non-overlapping, or slightly overlapping classes and are optimized for distances instead of densities. Consequently, they are sensitive to outliers. Therefore, outliers should be removed in the firs step. This is possible as the training set can be sampled in a selective way. Domain-based classification may be characterized as taking care of the structure of the classes in the feature space instead of their probability density functions.

If Vapnik's concept of structural risk minimization [20] is used for optimizing a separation function between two sets of vectors in a vector space, the resulting classifier is the maximum margin classifier. In case no linear classifier exists to make a perfect separation, a kernel approach may be used to construct a non-linear separation function. Thanks to the reproducing property of kernels, the SVM becomes then a maximum margin hyperplane in a Hilbert space induced by the specified kernel [25]. The margin is only determined by support vectors. These are the boundary objects, i.e. the objects closest to the decision boundary $f(\mathbf{x};\boldsymbol{\theta})$ [26,25]. As such, the SVM is independent of class density models. Multiple copies of the same object added to the training set do not contribute to the construction of the SVM as they do for classifiers based on some probabilistic model. Moreover, the SVM is also not affected by adding or removing objects of the same class that lie further away from the decision boundary. This decision function is, thereby, a truly domain-based classifier, as it optimizes the separation of class domains and class density functions.

For nonlinear classifiers defined on nonlinear kernels, the SVM has, however, a similar drawback as the nonlinear neural network. The distances to the decision boundary are computed in the output Hilbert space defined by the kernel and not in the input space. A second problem is that the soft-margin formulation [26], the traditional solution to overlapping classes, is not domain-based. Consider a two-class problem with the labels $y \in \{-1, +1\}$, where $y(\mathbf{x})$ denotes the true label of $\mathbf{x}$. Assume a training set $X = \{\mathbf{x}_i, y(\mathbf{x}_i)\}_{i=1}^{n}$. The optimization problem for a linear classifier $f(\mathbf{x}) = \mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0$ is rewritten into:

$$\begin{array}{ll} \min_{\mathbf{W}} & ||\mathbf{w}||^2 + C \sum_{\mathbf{x}_i \in X} \xi(\mathbf{x}_i), \\ s.t. & y(\mathbf{x}_i) f(\mathbf{x}_i) \geq 1 - \xi(\mathbf{x}_i), \\ & \xi(\mathbf{x}_i) \geq 0, \end{array} \tag{1}$$

where $\xi(\mathbf{x}_i)$ are slack variables accounting for possible errors and C is a trade-off parameter. $\sum_{\mathbf{x}_i \in X} \xi(\mathbf{x}_i)$ is an upper bound of the misclassification error on the training set, hence it is responsible for minimizing *a sum of error contributions.* Adding a copy of an erroneously assigned object will affect this sum and, thereby, will influence the sought optimum $\mathbf{w}$. The result is, thereby, based on a mixture of approaches. It is dependent on the distribution of objects (hence statistics) as well as on their domains (hence geometry).

A proper domain-based solution should minimize the class overlap in terms of distances and not in terms of probability densities. Hence, a suitable version of the SVM should be derived for the case of overlapping domains, resulting in the *negative margin SVM* [24]. This means that the distance of the furthest away misclassified object should be minimized. As the signed distance is negative, the negative margin is obtained. In the probabilistic approach, this classifier is unpopular as it will be sensitive to outliers. As explained above, outliers are neglected in domain-based classification, as they have to be removed beforehand.

Our conclusion is that the use of features yields a reduced representation. This leads to class overlap for which a probabilistic approach is needed. It relies on a heavy assumption that data are drawn independently from a fixed (but unknown) probability distribution. As a result, one demands training sets that are representative for the probability density functions. An approach based on distances and class structures may be formulated, but conflicts with the use of densities if classes overlap.

4 Proximity Representation

Similarity or dissimilarity measures can be used to represent objects by their proximities to other examples instead of representing them by a preselected set of features. If such measurements are derived from original objects, or from raw sensor data describing the objects fully (e.g. images, time signals and spectra that are as good as the real objects for the human observer), then the reduction in representation, which causes class overlap in the case of features, is circumvented. For example, we may demand that the dissimilarity of an object to itself is zero and that it can only be zero if it is related to an identical object. If it can be assumed that identical objects belong to the same class, classes do not overlap. (This is not always the case, e.g. a handwritten '7' may be identical to a handwritten '1').

In principle, such proximity representations may avoid class overlap. Hence, they may offer a possibility to use the structure of the classes in the representation, i.e. their domains, for building classifiers. This needs a special, not yet well studied variant of the proximity representation. Before a further explanation, we will first summarize two variants that have been worked out well. This summary is an adapted version of what has been published as [16]. See also [15].

Assume we are given a representation set R, i.e. a set of real-world objects that can be used for building the representation. $R = \{p_1, p_2, \ldots, p_n\}$ is, thereby, a set of prototype examples. We also consider a proximity measure d, which should incorporate the necessary invariance (such as scale or rotation invariance) for the given problem. Without loss of generality, let d denote dissimilarity. An object x is then represented as a vector of dissimilarities computed between x and the prototypes from R, i.e. $d(x, R) = [d(x, p_1), d(x, p_2), \ldots, d(x, p_n)]^{\mathsf{T}}$. If we are also given an additional labeled training set $T = \{t_1, t_2, \ldots, t_N\}$ of N real-world objects, our proximity representation becomes an $N \times n$ dissimilarity matrix $D(T, R)$, where $D(t_i, R)$ is now a row vector. Usually R is selected out of T (by

various prototype selection procedures) in a way to guarantee a good tradeoff between the recognition accuracy and the computational complexity. R and T may also be different sets.

The k-NN rule can directly be applied to such proximity data. Although it has good asymptotic properties for metric distances, its performance deteriorates for small training (here: representation) sets. Alternative learning strategies represent proximity information in suitable representation vector spaces, in which traditional statistical algorithms can be defined. So, they become more beneficial. Such vector spaces are usually determined by some local or global embedding procedures. Two approaches to be discussed here rely on a linear isometric embedding in a pseudo-Euclidean space (where necessarily $R \subseteq T$) and the use of proximity spaces; see [16,15].

Pseudo-Euclidean linear embedding. Given a symmetric dissimilarity matrix $D(R,R)$, a vectorial representation X can be found such that the distances are preserved. It is usually not possible to determine such an isometric embedding into a Euclidean space, but it is possible into a pseudo-Euclidean space $\mathcal{E} = \mathbb{R}^{(p,q)}$. It is a $(p+q)$-dimensional non-degenerate indefinite inner product space such that the inner product $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ is positive definite on $\mathbb{R}^p$ and negative definite on $\mathbb{R}^q$ [10]. Then, $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{E}} = \mathbf{x}^{\mathsf{T}} \mathcal{J}_{pq} \mathbf{y}$, where $\mathcal{J}_{pq} = \mathrm{diag}\,(I_{p\times p}; -I_{q\times q})$ and I is the identity matrix. Consequently, $d^2_{\mathcal{E}}(\mathbf{x},\mathbf{y}) = \langle \mathbf{x}-\mathbf{y}, \mathbf{x}-\mathbf{y} \rangle_{\mathcal{E}} = d^2_{\mathbb{R}^p}(\mathbf{x},\mathbf{y}) - d^2_{\mathbb{R}^q}(\mathbf{x},\mathbf{y})$, hence $d^2_{\mathcal{E}}$ is a difference of square Euclidean distances found in the two subspaces, $\mathbb{R}^p$ and $\mathbb{R}^q$. Since $\mathcal{E}$ is a linear space, many properties related to inner products can be extended from the Euclidean case [10,15].

The (indefinite) Gram matrix G of X can be expressed by the square distances $D^{\star 2} = (d^2_{ij})$ as $G = -\frac{1}{2} J D^{\star 2} J$, where $J = I - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}}$ [10,27,15]. Hence, X can be determined by the eigendecomposion of G, such that $G = Q \Lambda Q^{\mathsf{T}} = Q |\Lambda|^{1/2} \mathrm{diag}(\mathcal{J}_{p'q'}; 0)\, |\Lambda|^{1/2} Q^{\mathsf{T}}$. $|\Lambda|$ is a diagonal matrix of first decreasing p' positive eigenvalues, then decreasing magnitudes of q' negative eigenvalues, followed by zeros. Q is a matrix of the corresponding eigenvectors. X is uncorrelated and represented in $\mathbb{R}^k$, $k = p' + q'$, as $X = Q_k |\Lambda_k|^{1/2}$ [10,27]. Since only some eigenvalues are significant (in magnitude), the remaining ones can be disregarded as non-informative. The reduced representation $X_r = Q_m\, |\Lambda_m|^{1/2}$, $m = p+q < k$, is determined by the largest p positive and the smallest q negative eigenvalues. New objects $D(T_{test}, R)$ are orthogonally projected onto $\mathbb{R}^m$; see [10,27,15]. Classifiers based on inner products can appropriately be defined in $\mathcal{E}$. A linear classifier $f(\mathbf{x}) = \mathbf{v}^{\mathsf{T}} \mathcal{J}_{pq} \mathbf{x} + v_0$ is e.g. constructed by addressing it as $f(\mathbf{x}) = \mathbf{w}^{\mathsf{T}} \mathbf{x} + v_0$, where $\mathbf{w} = \mathcal{J}_{pq} \mathbf{v}$ in the associated Euclidean space $\mathbb{R}^{(p+q)}$ [10,27,15].

Proximity spaces. Here, the dissimilarity matrix $D(X,R)$ is interpreted as a data-dependent mapping $D(\cdot, R)\colon X \to \mathbb{R}^n$ from some initial representation X to a vector space defined by the set R. This is the *dissimilarity space* (or a similarity space, if similarities are used), in which each dimension $D(\cdot, p_i)$ corresponds to a dissimilarity to a prototype $p_i \in R$. The property that dissimilarities should be small for similar objects (belonging to the same class) and large for distinct objects, gives them a discriminative power. Hence, the vectors $D(\cdot, p_i)$

can be interpreted as 'features' and traditional statistical classifiers can be defined [28,15]. Although the classifiers are trained on $D(\cdot, R)$, the weights are still optimized on the complete set T. Thereby, they can outperform the k-NN rule as they become more global in their decisions.

Normal density-based classifiers perform well in dissimilarity spaces [27,28,15]. This especially holds for summation-based dissimilarity measures, summing over a number of components with similar variances. Such dissimilarities are approximately normally distributed thanks to the central limit theorem (or they approximate the χ^2 distribution if some variances are dominant) [15]. For instance, for a two-class problem, a quadratic normal density based classifier is given by $f(D(x,R)) = \sum_{i=1}^{2} \frac{(-1)^i}{2}(D(x,R)-\mathbf{m}_i)^\mathsf{T} S_i^{-1}\,(D(x,R)-\mathbf{m}_i) + \log\frac{p_1}{p_2} + \frac{1}{2}\log\frac{|S_1|}{|S_2|}$, where $\mathbf{m}_i$ are the mean vectors and S_i are the class covariance matrices, all estimated in the dissimilarity space $D(\cdot, R)$. p_i are the class prior probabilities. By replacing S_1 and S_2 by the average covariance matrix, a linear classifier is obtained.

The two learning frameworks of pseudo-Euclidean embedding and dissimilarity spaces appear to be successful in many problems with various kinds of dissimilarity measures. They can be more accurate and more efficient than the nearest neighbor rule, traditionally applied to dissimilarity data. Thereby, they provide beneficial approaches to learning from structural object descriptions for which it is more easy to define dissimilarity measures between objects than to find a good set of features. As long as these approaches are based on a fixed representation set, however, class overlap may still arise as two different objects may have the same set of distances to the representation set. Moreover, most classifiers used in the representation spaces are determined based on the traditional principle of minimizing the overlap. They do not make a specific use of principles related to object distances or class domains. So, what is still lacking are procedures that use class distances to construct a structural description of classes. The domain-based classifiers, introduced in Section 3, may offer that in future provided that the representation set is so large that the class overlap is (almost) avoided. A more fundamental approach is described below.

Topological spaces. The topological foundation of proximity representations is discussed in [15]. It is argued that if the dissimilarity measure itself is unknown, but the dissimilarity values are given, the topology cannot, as usual, be based on the traditional idempotent closures. An attempt has been made to use neighborhoods instead. This has not resulted yet in a useful generalization over finite training sets.

Topological approaches will aim to describe the class structures from local neighborhood relations between objects. The inherent difficulty is that many of the dissimilarity measures used in structural pattern recognition, like the normalized edit distance, are non-Euclidean, and even sometimes non-metric. It has been shown in a number of studies that straightforward Euclidean corrections are counter productive in some applications. This suggests that the non-Euclidean aspects may be informative. Consequently, a non-Euclidean topology would be needed. This area is still underdeveloped.

A better approach may rely on two additional sources of information that are additionally available. These are the definition of the dissimilarity measure and the assumption of class compactness. They may together tell us what is really local or how to handle the non-Euclidean phenomena of the data. This should result in a topological specification of the class structure as learned from the training set.

5 Structural Representation

In the previous section we arrived at *a structure of a class* (or a concept), i.e. the structural or topological relation of the set of all objects belonging to a particular class. This structure is influenced by the chosen representation, but is in fact determined by the class of objects. It reflects, for instance, the set of continuous transformations of the handwritten digits '7' that generate exclusively all other forms that can be considered as variants of a handwritten '7'. This basically reflects the concept used by experts to assign the class label. Note, however, that this rather abstract structure of the concept should be clearly distinguished from the structure of individual objects that are the manifestations of that concept.

The *structure of objects*, as presented somewhere in sensory data of images, such as time signals and spectra, is directly related to shape. The shape is a one- or multi-dimensional set of connected boundary points that may be locally characterized by curvature and described more globally by morphology and topology. Note that the object structure is related to an outside border of objects, the place where the object ends. If the object is a black blob in a two-dimensional image (e.g. a handwritten digit) then the structure is expressed by the contour, a one-dimensional closed line. If the grey-value pixel intensities inside the blob are relevant, then we deal with a three-dimensional blob on a two-dimensional surface. (As caves cannot exist in this structure it is sometimes referred to as a 2.5-dimensional object).

It is important to realize that the sensor measurements are characterized by a sampling structure (units), such as pixels or time samples. This sampling structure, however, has nothing to do with the object structure. In fact, it disturbs it. In principle, objects (patterns describing real objects) can lie anywhere in an image or in a time frame. They can also be rotated in an image and appear in various scales. Additionally, we may also vary the sampling frequency. If we analyze the object structure for a given sampling, then the object is "nailed" to some grid. Similar objects may be nailed in an entirely different way to this grid. How to construct structural descriptions of objects that are independent of the sampling grid on which the objects are originally presented is an important topic of structural pattern recognition.

The problem of structural inference, however, is not the issue of representation itself. It is the question how we can establish the membership of an object to a given set of examples based on their structure. Why is it more likely that a new object X belongs to a set A than a set B? A few possible answers are presented below.

1. X is an example of A, because the object in $A \cup B$ that is most similar to X belongs to A. This decision may depend on the accidental availability of particular objects. Moreover, similarity should appropriately be defined.
2. X is an example of A, because the object from $A \cup B$ that is most easily transformed to X belongs to A. In this case similarity relies on the effort of transformation. This may be more appropriate if structures need to be compared. The decision, however, still depends on a single object. The entire sets or classes simply store examples that may be used when other objects have to be classified.
3. X is an example of A, because it can more easily be generated by transforming the prototype of set A than by transforming the prototype of set B. The *prototype* of a set may be defined as the (hypothetical) object that can most easily be transformed into any of the objects of the set. In this assignment rule (as well as in the rule above) the definition of transformation is universal, i.e. independent of the considered class.
4. X is an example of A, because it can more easily be transformed from a (hypothetical) prototype object by the transformations T_A that are used to generate the set A than by the transformations T_B that are used to generate the set B. Note that we now allow that the sets are generated from possibly the same prototype, but by using different transformations. These are derived (learnt) from the sets of examples. The transformations T_A and T_B may be learnt from a training set.

There is a strong resemblance with the statistical class descriptions: classes may differ by their means as well as by the shape of their distributions. A very important difference, however, between structural and statistical inference is that for an additional example that is identical to a previous one changes the class distribution, but not the (minimal) set of necessary transformations.

This set of assignment rules can easily be modified or enlarged. We like to emphasize, however, that the natural way of comparing objects, i.e. by accounting for their similarity, may be defined as the effort of transforming one structure into another. Moreover, the set of possible transformations may differ from class to class. In addition, classes may have the same or different prototypes. E.g. a sphere can be considered as a basic prototype both for apples as well as for pears. In general, classes may differ by their prototypes and/or by their set of transformations.

What has been called easiness in transformation can be captured by a measurable cost, which is an example of a similarity measure. It is, thereby, related to the proximity approaches, described above. Proximity representations are naturally suitable for structural inference. What is different, however, is the use of statistical classifiers in embedded and proximity spaces. In particular, the embedding approach has to be redefined for structural inference as it makes use of averages and the minimization of an expected error, both statistical concepts. Also the use of statistical classifiers in these spaces conflicts with structural inference. In fact, they should be replaced by domain-based classifiers. The discussed topological approach, on the other hand, fits to the concept of structural inference.

The idea that transformations may be class-dependent has not been worked out by us in the proximity-based approach. There is, however, not a fundamental

objection against the possibility to attribute set of objects, or even individual objects in the training set with their own proximity measure. This will very likely lead to non-Euclidean data, but we have shown ways how to handle them. What is not studied is how to optimize proximity measures (structure transformations) over the training data. A possibility might be to normalize for differences in class structure by adapting the proximity measures that determined these structures.

There is, however, an important aspect of learning from structures that cannot currently be covered by domain-based classifiers built for a proximity representation. Structures can be considered as assemblies of more primitive structures, similarly as a house is built from bricks. These primitives may have a finite size, or may also be infinitesimally small. The corresponding transformations from one structure into another become thereby continuous. In particular, we are interested in such transformations as they may constitute the compactness of classes on which a realistic set of pattern recognition problems can be defined. It may be economical to allow for locally-defined functions in order to derive (or learn) transformations between objects. For instance, while comparing dogs and wolves, or while describing these groups separately, other transformations may be of interest for the description of ears then for the tails. Such a decomposition of transformations is not possible in the current proximity framework, as it starts with relations between entire objects. A further research is needed.

The automatic detection of parts of objects where different transformations may be useful for the discrimination (or a continuous varying transformation over the object) seems very challenging, as the characteristics inside an object are ill-defined as long as classes are not fully established during training. Some attempts in this direction have been made by Paclík [29,30] when he tries to learn the proximity measure from a training set.

In summary, we see three ways to link structural object descriptions to the proximity representation:

- Finding or generating prototypical objects that can easily be transformed into the given training set. They will be used in the representation set.
- Determining specific proximity measures for individual objects or for groups of objects.
- Learning locally dependent (inside the object) proximity measures.

6 Discussion and Conclusions

In this paper, we presented a discussion of the possibilities of structural inference as opposed to statistical inference. By using the structural properties of objects and classes of a given set of examples, knowledge such as class labels is inferred for new objects. Structural and statistical inference are based on different assumptions with respect to the set of examples needed for training and for the object representation. In a statistical approach, the training set has to be representative for the class distributions as the classifiers have to assign objects to the most probable class. In a structural approach, classes may be assumed to be separable. As a consequence, domain-based classifiers may be used [18,24]. Such classifiers, which are mainly still under development, do not need training sets

that are representative for the class distributions, but which are representative for the class domains. This is greatly advantageous as these domains are usually stable with respect to changes in the context of application. Training sets may thereby be collected by a selective, instead of unselective sampling.

The below table summarizes the main differences between representations based on features (F), proximities (P) and structures (S) for the statistical and structural inference.

	Statistical inference	Structural inference
F	Features reduce; statistical inference is almost obligatory.	The structural information is lost by representing the aspects of objects by vectors and/or due to the reduction.
P	Proximity representations can be derived by comparing pairs of objects (e.g. initially described by features or structures). Statistical classifiers are built in proximity spaces or in (pseudo-Euclidean) embedded spaces.	Transformations between the structures of objects may be used to build proximity representations. Classes of objects should be separated by domain-based classifiers.
S	Statistical learning is only possible if a representation vector space is built (by features or proximities), in which density functions can be defined.	Transformations might be learnt by using a domain-based approach that transforms one object into another in an economical way.

This paper summarizes the possibilities of structural inference. In particular, the possibilities of the proximity representation are emphasized, provided that domain-based learning procedures follow. More advanced approaches, making a better usage of the structure of individual objects have to be studied further. They may be based on the generation of prototypes or on trained, possibly local transformations, which will separate object classes better. Such transformations can be used to define proximity measures, which will be further used to construct a proximity representation. Representations may have to be directly built on the topology derived from object neighborhoods. These neighborhoods are constructed by relating transformations to proximities. The corresponding dissimilarity measures will be non-Euclidean, in general. Consequently, non-Euclidean topology has to be studied to proceed in this direction fundamentally.

Acknowledgments. This work is supported by the Dutch Organization for Scientific Research (NWO).

References

1. Sayre, K.: Recognition, a study in the philosophy of artificial intelligence. University of Notre Dame Press (1965)
2. Watanabe, S.: Pattern Recogn. Human and Mechanical. Academic Press (1974)
3. Fu, K.: Syntactic Pattern Recognition and Applications. Pretice-Hall (1982)
4. Fukunaga, K.: Introduction to Statistical Pattern Recogn. Academic Press (1990)
5. Duda, R., Hart, P., Stork, D.: Pattern Classification. John Wiley & Sons, Inc. (2001)
6. Webb, A.: Statistical Pattern Recognition. John Wiley & Sons, Ltd. (2002)

7. Jain, A., Duin, R.P.W., Mao, J.: Statistical pattern recognition: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(1) (2000) 4–37
8. Bunke, H., Günter, S., Jiang, X.: Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching. In: International Conference on Advances in Pattern Recognition. (2001) 1–11
9. Fu, K.: A step towards unification of syntactic and statistical pattern recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **8** (1986)
10. Goldfarb, L.: A new approach to pattern recognition. In Kanal, L., Rosenfeld, A., eds.: Progress in Pattern Recognition. Volume 2. Elsevier Science Publishers BV (1985) 241–402
11. Goldfarb, L., Gay, D.: What is a structural representation? Fifth variation. Technical Report TR05-175, University of New Brunswick, Fredericton, Canada (2005)
12. Goldfarb, L.: What is distance and why do we need the metric model for pattern learning? Pattern Recognition **25**(4) (1992) 431–438
13. Goldfarb, L., Golubitsky, O.: What is a structural measurement process? Technical Report TR01-147, University of New Brunswick, Fredericton, Canada (2001)
14. Gutkin, A., Gya, D., Goldfarb, L., Webster, M.: On the articulatory representation of speech within the evolving transformation system formalism. In Goldfarb, L., ed.: Pattern representation and the future of pattern recognition, ICPR 2004 Workshop Proceedings, Cambridge, United Kingdom (2004) 57–76
15. Pękalska, E., Duin, R.P.W.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
16. Pękalska, E., Duin, R.P.W.: Learning with general proximity measures. In: Pattern Recognition in Information Systems. Volume 6. (2006)
17. Duin, R.P.W., Pękalska, E.: Open issues in pattern recognition. In: Computer Recognition Systems. Springer, Berlin (2005) 27–42
18. Duin, R.P.W., Pękalska, E.: The science of pattern recognition. Achievements and perspectives. (2006, submitted)
19. Dawid, A., Stone, M., Zidek, J.: Marginalization paradoxes in Bayesian and structural inference. J. Royal Stat. Soc., B **35** (1973) 180–223
20. Vapnik, V.: Estimation of Dependences based on Empirical Data, 2nd ed. Springer Verlag (2006)
21. Jain, A.K., Chandrasekaran, B.: Dimensionality and sample size considerations in pattern recognition practice. In Krishnaiah, P.R., Kanal, L.N., eds.: Handbook of Statistics. Volume 2. North-Holland, Amsterdam (1987) 835–855
22. Wolpert, D.: The Mathematics of Generalization. Addison-Wesley (1995)
23. Duin, R.P.W., Roli, F., de Ridder, D.: A note on core research issues for statistical pattern recognition. Pattern Recognition Letters **23**(4) (2002) 493–499
24. Duin, R.P.W., Pękalska, E.: Domain-based classification. Technical report, TU Delft (2005) `http://ict.ewi.tudelft.nl/~{}duin/papers/Domain_class_05.pdf`.
25. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Inc. (1998)
26. Cristianini, N., Shawe-Taylor, J.: Support Vector Machines and other kernel-based learning methods. Cambridge University Press, UK (2000)
27. Pękalska, E., Paclík, P., Duin, R.P.W.: A Generalized Kernel Approach to Dissim. Based Classification. Journal of Machine Learning Research **2**(2) (2002) 175–211
28. Pękalska, E., Duin, R.P.W., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition **39**(2) (2006) 189–208
29. Paclík, P., Novovicova, J., Duin, R.P.W.: Building road sign classifiers using a trainable similarity measure. Journal of Intelligent Transportations Systems (2006)
30. Paclík, P., Novovicova, J., Duin, R.P.W.: A trainable similarity measure for image classification. In: 17th Int. Conf. on Pattern Recognition. (2006)

The Science of Pattern Recognition. Achievements and Perspectives

Robert P.W. Duin[1] and Elżbieta Pękalska[2]

[1] ICT group, Faculty of Electr. Eng., Mathematics and Computer Science
Delft University of Technology, The Netherlands
r.duin@ieee.org
[2] School of Computer Science, University of Manchester, United Kingdom
pekalska@cs.man.ac.uk

Summary. Automatic pattern recognition is usually considered as an engineering area which focusses on the development and evaluation of systems that imitate or assist humans in their ability of recognizing patterns. It may, however, also be considered as a science that studies the faculty of human beings (and possibly other biological systems) to discover, distinguish, characterize patterns in their environment and accordingly identify new observations. The engineering approach to pattern recognition is in this view an attempt to build systems that simulate this phenomenon. By doing that, scientific understanding is gained of what is needed in order to recognize patterns, in general.

Like in any science understanding can be built from different, sometimes even opposite viewpoints. We will therefore introduce the main approaches to the science of pattern recognition as two dichotomies of complementary scenarios. They give rise to four different schools, roughly defined under the terms of expert systems, neural networks, structural pattern recognition and statistical pattern recognition. We will briefly describe what has been achieved by these schools, what is common and what is specific, which limitations are encountered and which perspectives arise for the future. Finally, we will focus on the challenges facing pattern recognition in the decennia to come. They mainly deal with weaker assumptions of the models to make the corresponding procedures for learning and recognition wider applicable. In addition, new formalisms need to be developed.

1 Introduction

We are very familiar with the human ability of pattern recognition. Since our early years we have been able to recognize voices, faces, animals, fruits or inanimate objects. Before the speaking faculty is developed, an object like a ball is recognized, even if it barely resembles the balls seen before. So, except for the memory, the skills of abstraction and generalization are essential to find our way in the world. In later years we are able to deal with much more

Robert P.W. Duin and Elżbieta Pękalska: *The Science of Pattern Recognition. Achievements and Perspectives*, Studies in Computational Intelligence (SCI) **63**, 221–259 (2007)
www.springerlink.com

complex patterns that may not directly be based on sensorial observations. For example, we can observe the underlying theme in a discussion or subtle patterns in human relations. The latter may become apparent, e.g. only by listening to somebody's complaints about his personal problems at work that again occur in a completely new job. Without a direct participation in the events, we are able to see both analogy and similarity in examples as complex as social interaction between people. Here, we learn to distinguish the pattern from just two examples.

The pattern recognition ability may also be found in other biological systems: the cat knows the way home, the dog recognizes his boss from the footsteps or the bee finds the delicious flower. In these examples a direct connection can be made to sensory experiences. Memory alone is insufficient; an important role is that of generalization from observations which are similar, although not identical to the previous ones. A scientific challenge is to find out how this may work.

Scientific questions may be approached by building models and, more explicitly, by creating simulators, i.e. artificial systems that roughly exhibit the same phenomenon as the object under study. Understanding will be gained while constructing such a system and evaluating it with respect to the real object. Such systems may be used to replace the original ones and may even improve some of their properties. On the other hand, they may also perform worse in other aspects. For instance, planes fly faster than birds but are far from being autonomous. We should realize, however, that what is studied in this case may not be the bird itself, but more importantly, the ability to fly. Much can be learned about flying in an attempt to imitate the bird, but also when differentiating from its exact behavior or appearance. By constructing fixed wings instead of freely movable ones, the insight in how to fly grows. Finally, there are engineering aspects that may gradually deviate from the original scientific question. These are concerned with how to fly for a long time, with heavy loads, or by making less noise, and slowly shift the point of attention to other domains of knowledge.

The above shows that a distinction can be made between the scientific study of pattern recognition as the ability to abstract and generalize from observations and the applied technical area of the design of artificial pattern recognition devices without neglecting the fact that they may highly profit from each other. Note that patterns can be distinguished on many levels, starting from simple characteristics of structural elements like strokes, through features of an individual towards a set of qualities in a group of individuals, to a composite of traits of concepts and their possible generalizations. A pattern may also denote a single individual as a representative for its population, model or concept. Pattern recognition deals, therefore, with patterns, regularities, characteristics or qualities that can be discussed on a low level of sensory measurements (such as pixels in an image) as well as on a high level of the derived and meaningful concepts (such as faces in images). In this work, we will focus on the scientific aspects, i.e. what we know about the way pattern

recognition works and, especially, what can be learned from our attempts to build artificial recognition devices.

A number of authors have already discussed the science of pattern recognition based on their simulation and modeling attempts. One of the first, in the beginning of the sixties, was Sayre [64], who presented a philosophical study on perception, pattern recognition and classification. He made clear that classification is a task that can be fulfilled with some success, but recognition either happens or not. We can stimulate the recognition by focussing on some aspects of the question. Although we cannot set out to fully recognize an individual, we can at least start to classify objects on demand. The way Sayre distinguishes between recognition and classification is related to the two subfields discussed in traditional texts on pattern recognition, namely unsupervised and supervised learning. They fulfill two complementary tasks. They act as automatic tools in the hand of a scientist who sets out to find the regularities in nature.

Unsupervised learning (also related to exploratory analysis or cluster analysis) gives the scientist an automatic system to indicate the presence of yet unspecified patterns (regularities) in the observations. They have to be confirmed (verified) by him. Here, in the terms of Sayre, a pattern is recognized. **Supervised learning** is an automatic system that verifies (confirms) the patterns described by the scientist based on a representation defined by him. This is done by an automatic classification followed by an evaluation.

In spite of Sayre's discussion, the concepts of pattern recognition and classification are still frequently mixed up. In our discussion, classification is a significant component of the pattern recognition system, but unsupervised learning may also play a role there. Typically, such a system is first presented with a set of known objects, the training set, in some convenient representation. Learning relies on finding the data descriptions such that the system can correctly characterize, identify or classify novel examples. After appropriate preprocessing and adaptations, various mechanisms are employed to train the entire system well. Numerous models and techniques are used and their performances are evaluated and compared by suitable criteria. If the final goal is prediction, the findings are validated by applying the best model to unseen data. If the final goal is characterization, the findings may be validated by complexity of organization (relations between objects) as well as by interpretability of the results.

Fig. 1 shows the three main stages of pattern recognition systems: Representation, Generalization and Evaluation, and an intermediate stage of Adaptation [20]. The system is trained and evaluated by a set of examples, the Design Set. The components are:

- **Design Set.** It is used both for training and validating the system. Given the background knowledge, this set has to be chosen such that it is representative for the set of objects to be recognized by the trained system.

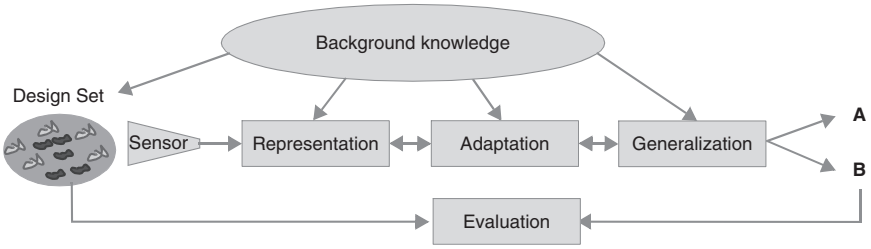


Fig. 1. Components of a pattern recognition system

There are various approaches how to split it into suitable subsets for training, validation and testing. See e.g. [22, 32, 62, 77] for details.

- **Representation.** Real world objects have to be represented in a formal way in order to be analyzed and compared by mechanical means such as a computer. Moreover, the observations derived from the sensors or other formal representations have to be integrated with the existing, explicitly formulated knowledge either on the objects themselves or on the class they may belong to. The issue of representation is an essential aspect of pattern recognition and is different from classification. It largely influences the success of the stages to come.
- **Adaptation.** It is an intermediate stage between Representation and Generalization, in which representations, learning methodology or problem statement are adapted or extended in order to enhance the final recognition. This step may be neglected as being transparent, but its role is essential. It may reduce or simplify the representation, or it may enrich it by emphasizing particular aspects, e.g. by a nonlinear transformation of features that simplifies the next stage. Background knowledge may appropriately be (re)formulated and incorporated into a representation. If needed, additional representations may be considered to reflect other aspects of the problem. Exploratory data analysis (unsupervised learning) may be used to guide the choice of suitable learning strategies.
- **Generalization or Inference.** In this stage we learn a concept from a training set, the set of known and appropriately represented examples, in such a way that predictions can be made on some unknown properties of new examples. We either generalize towards a concept or infer a set of general rules that describe the qualities of the training data. The most common property is the class or pattern it belongs to, which is the above mentioned classification task.
- **Evaluation.** In this stage we estimate how our system performs on known training and validation data while training the entire system. If the results are unsatisfactory, then the previous steps have to be reconsidered.

Different disciplines emphasize or just exclusively study different parts of this system. For instance, perception and computer vision deal mainly with

the representation aspects [21], while books on artificial neural networks [62], machine learning [4, 53] and pattern classification [15] are usually restricted to generalization. It should be noted that these and other studies with the words "pattern" and "recognition" in the title often almost entirely neglect the issue of representation. We think, however, that *the main goal of the field of pattern recognition is to study generalization in relation to representation* [20].

In the context of representations, and especially images, generalization has been thoroughly studied by Grenander [36]. What is very specific and worthwhile is that he deals with infinite representations (say, unsampled images), thereby avoiding the frequently returning discussions on dimensionality and directly focussing on a high, abstract level of pattern learning. We like to mention two other scientists that present very general discussions on the pattern recognition system: Watanabe [75] and Goldfarb [31, 32]. They both emphasize the structural approach to pattern recognition that we will discuss later on. Here objects are represented in a form that focusses on their structure. A generalization over such structural representations is very difficult if one aims to learn the *concept*, i.e. the underlying, often implicit definition of a pattern class that is able to generate possible realizations. Goldfarb argues that traditionally used numeric representations are inadequate and that an entirely new, structural representation is necessary. We judge his research program as very ambitious, as he wants to learn the (generalized) structure of the concept from the structures of the examples. He thereby aims to make explicit what usually stays implicit. We admit that a way like his has to be followed if one ever wishes to reach more in concept learning than the ability to name the right class with a high probability, without having built a proper understanding.

In the next section we will discuss and relate well-known general scientific approaches to the specific field of pattern recognition. In particular, we like to point out how these approaches differ due to fundamental differences in the scientific points of view from which they arise. As a consequence, they are often studied in different traditions based on different paradigms. We will try to clarify the underlying cause for the pattern recognition field. In the following sections we sketch some perspectives for pattern recognition and define a number of specific challenges.

2 Four Approaches to Pattern Recognition

In science, new knowledge is phrased in terms of existing knowledge. The starting point of this process is set by generally accepted evident views, or observations and facts that cannot be explained further. These foundations, however, are not the same for all researchers. Different types of approaches may be distinguished originating from different starting positions. It is almost a type of taste from which perspective a particular researcher begins. As a

consequence, different 'schools' may arise. *The point of view, however, determines what we see.* In other words, staying within a particular framework of thought we cannot achieve more than what is derived as a consequence of the corresponding assumptions and constraints. To create more complete and objective methods, we may try to integrate scientific results originating from different approaches into a single pattern recognition model. It is possible that confusion arises on how these results may be combined and where they essentially differ. But the combination of results of different approaches may also appear to be fruitful, not only for some applications, but also for the scientific understanding of the researcher that broadens the horizon of allowable starting points. This step towards a unified or integrated view is very important in science as only then a more complete understanding is gained or a whole theory is built.

Below we will describe four approaches to pattern recognition which arise from two different dichotomies of the starting points. Next, we will present some examples illustrating the difficulties of their possible interactions. This discussion is based on earlier publications [16, 17].

2.1 Platonic and Aristotelian Viewpoints

Two principally different approaches to almost any scientific field rely on the so-called Platonic and Aristotelian viewpoints. In a first attempt they may be understood as top-down and bottom-up ways of building knowledge. They are also related to deductive (or holistic) and inductive (or reductionistic) principles. These aspects will be discussed in Section 4.

The Platonic approach starts from generally accepted concepts and global ideas of the world. They constitute a coherent picture in which many details are undefined. The primary task of the Platonic researcher is to recognize in his[3] observations the underlying concepts and ideas that are already accepted by him. Many theories of the creation of the universe or the world rely on this scenario. An example is the drifts of the continents or the extinction of the mammoths. These theories do not result from a reasoning based on observations, but merely from a more or less convincing global theory (depending on the listener!) that seems to extrapolate far beyond the hard facts. For the Platonic researcher, however, it is not an extrapolation, but an adaptation of previous formulations of the theory to new facts. That is the way this approach works: existing ideas that have been used for a long time are gradually adapted to new incoming observations. The change does not rely on an essential paradigm shift in the concept, but on finding better, more appropriate relations with the observed world in definitions and explanations. The essence of the theory has been constant for a long time. So, in practise the

[3] For simplicity, we refer to researchers in a male form; we mean both women and men.

Platonic researcher starts from a theory which can be stratified into to a number of hypotheses that can be tested. Observations are collected to test these hypotheses and, finally, if the results are positive, the theory is confirmed.

The observations are of primary interest in the Aristotelian approach. Scientific reasoning stays as closely as possible to them. It is avoided to speculate on large, global theories that go beyond the facts. The observations are always the foundation on which the researcher builds his knowledge. Based on them, patterns and regularities are detected or discovered, which are used to formulate some tentative hypotheses. These are further explored in order to arrive at general conclusions or theories. As such, the theories are not global, nor do they constitute high level descriptions. A famous guideline here is the so-called Occam's razor principle that urges one to avoid theories that are more complex than strictly needed for explaining the observations. Arguments may arise, however, since the definition of complexity depends, e.g. on the mathematical formalism that is used.

The choice for a particular approach may be a matter of preference or determined by non-scientific grounds, such as upbringing. Nobody can judge what the basic truth is for somebody else. Against the Aristotelians may be held that they do not see the overall picture. The Platonic researchers, on the other hand, may be blamed for building castles in the air. Discussions between followers of these two approaches can be painful as well as fruitful. They may not see that their ground truths are different, leading to pointless debates. What is more important is the fact that they may become inspired by each other's views. One may finally see real world examples of his concepts, while the other may embrace a concept that summarizes, or constitutes an abstraction of his observations.

2.2 Internal and the External Observations

In the contemporary view science is *'the observation, identification, description, experimental investigation, and theoretical explanation of phenomena'*[4] or *'any system of knowledge that is concerned with the physical world and its phenomena and that entails unbiased observations and systematic experimentation.*[5] So, the aspect of observation that leads to a possible formation of a concept or theory is very important. Consequently, the research topic of the science of pattern recognition, which aims at the generalization from observations for knowledge building, is indeed scientific. Science is in the end a brief explanation summarizing the observations achieved through abstraction and their generalization.

Such an explanation may primarily be observed by the researcher in his own thinking. Pattern recognition research can thereby be performed by introspection. The researcher inspects himself how he generalizes from

[4] `http://dictionary.reference.com/`
[5] `http://www.britannica.com/`

observations. The basis of this generalization is constituted by the primary observations. This may be an entire object ('I just see that it is an apple') or its attributes ('it is an apple because of its color and shape'). We can also observe pattern recognition in action by observing other human beings (or animals) while they perform a pattern recognition task, e.g. when they recognize an apple. Now the researcher tries to find out by experiments and measurements how the subject decides for an apple on the basis of the stimuli presented to the senses. He thereby builds a model of the subject, from senses to decision making.

Both approaches result into a model. In the external approach, however, the senses may be included in the model. In the internal approach, this is either not possible or just very partially. We are usually not aware of what happens in our senses. Introspection thereby starts by what they offer to our thinking (and reasoning). As a consequence, models based on the internal approach have to be externally equipped with (artificial) senses, i.e. with sensors.

2.3 The Four Approaches

The following four approaches can be distinguished by combining the two dichotomies presented above:

(1) Introspection by a Platonic viewpoint: object modeling.
(2) Introspection by an Aristotelian viewpoint: generalization.
(3) Extrospection by an Aristotelian viewpoint: system modeling.
(4) Extrospection by a Platonic viewpoint: concept modeling.

These four approaches will now be discussed separately. We will identify some known procedures and techniques that may be related to these. See also Fig. 2.

Object modeling. This is based on introspection from a Platonic viewpoint. The researcher thereby starts from global ideas on how pattern recognition systems may work and tries to verify them in his own thinking and reasoning. He thereby may find, for instance, that particular color and shape descriptions of an object are sufficient for him to classify it as an apple. More generally, he may discover that he uses particular reasoning rules operating on a fixed set of possible observations. The so-called syntactic and structural approaches to pattern recognition [26] thereby belong to this area, as well as the case-based reasoning [3]. There are two important problems in this domain: how to constitute the general concept of a class from individual object descriptions and how to connect particular human qualitative observations such as 'sharp edge' or 'egg shaped' with physical sensor measurements.

Generalization. Let us leave the Platonic viewpoint and consider a researcher who starts from observations, but still relies on introspection. He wonders what he should do with just a set of observations without any framework. An important point is the nature of observations. Qualitative observations such as 'round', 'egg-shaped' or 'gold colored' can be judged as recognitions in themselves based on low-level outcomes of senses. It is difficult to

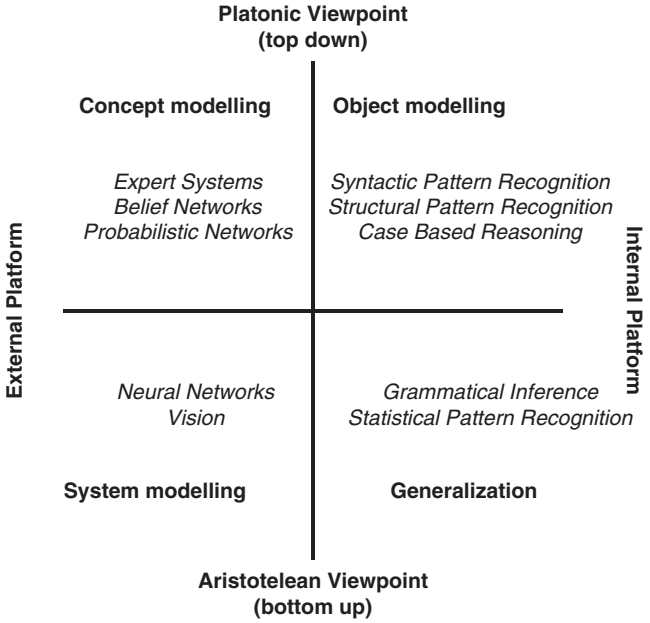


Fig. 2. Four approaches to Pattern Recognition

neglect them and to access the outcomes of senses directly. One possibility for him is to use artificial senses, i.e. of sensors, which will produce quantitative descriptions. The next problem, however, is how to generalize from such numerical outcomes. The physiological process is internally unaccessible. A researcher who wonders how he himself generalizes from low level observations given by numbers may rely on statistics. This approach thereby includes the area of statistical pattern recognition.

If we consider low-level inputs that are not numerical, but expressed in attributed observations as 'red, egg-shaped', then the generalization may be based on logical or grammatical inference. As soon, however, as the structure of objects or attributes is not generated from the observations, but derived (postulated) from a formal global description of the application knowledge, e.g. by using graph matching, the approach is effectively top-down and thereby starts from object or concept modeling.

System modeling. We now leave the internal platform and concentrate on research that is based on the external study of the pattern recognition abilities of humans and animals or their brains and senses. If this is done in a bottom-up way, the Aristotelian approach, then we are in the area of low-level modeling of senses, nerves and possibly brains. These models are based on the physical and physiological knowledge of cells and the proteins and minerals that constitute them. Senses themselves usually do not directly generalize from observations. They may be constructed, however, in such a way

that this process is strongly favored on a higher level. For instance, the way the eye (and the retina, in particular) is constructed, is advantageous for the detection of edges and movements as well as for finding interesting details in a global, overall picture. The area of vision thereby profits from this approach. It is studied how nerves process the signals they receive from the senses on a level close to the brain. Somehow this is combined towards a generalization of what is observed by the senses. Models of systems of multiple nerves are called neural networks. They appeared to have a good generalization ability and are thereby also used in technical pattern recognition applications in which the physiological origin is not relevant [4, 62].

Concept modeling. In the external platform, the observations in the starting point are replaced by ideas and concepts. Here one still tries to externally model the given pattern recognition systems, but now in a top-down manner. An example is the field of expert systems: by interviewing experts in a particular pattern recognition task, it is attempted to investigate what rules they use and in what way they are using observations. Also belief networks and probabilistic networks belong to this area as far as they are defined by experts and not learned from observations. This approach can be distinguished from the above system modeling by the fact that it is in no way attempted to model a physical or physiological system in a realistic way. The building blocks are the ideas, concepts and rules, as they live in the mind of the researcher. They are adapted to the application by external inspection of an expert, e.g. by interviewing him. If this is done by the researcher internally by introspection, we have closed the circle and are back to what we have called object modeling, as the individual observations are our internal starting point. We admit that the difference between the two Platonic approaches is minor here (in contrast to the physiological level) as we can also try to interview ourselves to create an objective (!) model of our own concept definitions.

2.4 Examples of Interaction

The four presented approaches are four ways to study the science of pattern recognition. Resulting knowledge is valid for those who share the same starting point. If the results are used for building artificial pattern recognition devices, then there is, of course, no reason to restrict oneself to a particular approach. Any model that works well may be considered. There are, however, certain difficulties in combining different approaches. These may be caused by differences in culture, assumptions or targets. We will present two examples, one for each of the two dichotomies.

Artificial neural networks constitute an alternative technique to be used for generalization within the area of statistical pattern recognition. It has taken, however, almost ten years since their introduction around 1985 before neural networks were fully acknowledged in this field. In that period, the neural network community suffered from lack of knowledge on the competing

classification procedures. One of the basic misunderstandings in the pattern recognition field was caused by its dominating paradigm stating that learning systems should never be larger than strictly necessary, following the Occam's razor principle. It could have not been understood how largely oversized systems such as neural networks would have ever been able to generalize without adapting to peculiarities in the data (the so-called overtraining). At the same time, it was evident in the neural network community that the larger neural network the larger its flexibility, following the analogy that a brain with many neurons would perform better in learning than a brain with a few ones. When this contradiction was finally solved (an example of Kuhn's paradigm shifts [48]), the area of statistical pattern recognition was enriched with a new set of tools. Moreover, some principles were formulated towards understanding of pattern recognition that otherwise would have only been found with great difficulties.

In general, it may be expected that the internal approach profits from the results in the external world. It is possible that thinking, the way we generalize from observations, changes after it is established how this works in nature. For instance, once we have learned how a specific expert solves his problems, this may be used more generally and thereby becomes a rule in structural pattern recognition. The external platform may thereby be used to enrich the internal one.

A direct formal fertilization between the Platonic and Aristotelian approaches is more difficult to achieve. Individual researchers may build some understanding from studying each other's insights, and thereby become mutually inspired. The Platonist may become aware of realizations of his ideas and concepts. The Aristotelian may see some possible generalizations of the observations he collected. It is, however, still one of the major challenges in science to formalize this process.

How should existing knowledge be formulated such that it can be enriched by new observations? Everybody who tries to do this directly encounters the problem that observations may be used to reduce uncertainty (e.g. by the parameter estimation in a model), but that it is very difficult to formalize uncertainty in existing knowledge. Here we encounter a fundamental 'paradox' for a researcher summarizing his findings after years of observations and studies: he has found some answers, but almost always he has also generated more new questions. Growing knowledge comes with more questions. In any formal system, however, in which we manage to incorporate uncertainty (which is already very difficult), this uncertainty will be reduced after having incorporating some observations. We need an automatic hypothesis generation in order to generate new questions. How should the most likely ones be determined? We need to look from different perspectives in order to stimulate the creative process and bring sufficient inspiration and novelty to hypothesis generation. This is necessary in order to make a step towards building a complete theory. This, however, results in the computational complexity mentioned

in the literature [60] when the Platonic structural approach to pattern recognition has to be integrated with the Aristotelian statistical approach.

The same problem may also be phrased differently: how can we express the uncertainty in higher level knowledge in such a way that it may be changed (upgraded) by low level observations? Knowledge is very often structural and has thereby a qualitative nature. On the lowest level, however, observations are often treated as quantities, certainly in automatic systems equipped with physical sensors. And here the Platonic – Aristotelian polarity meets the internal – external polarity: by crossing the border between concepts and observations we also encounter the border between qualitative symbolic descriptions and quantitative measurements.

3 Achievements

In this section we will sketch in broad terms the state of the art in building systems for generalization and recognition. In practical applications it is not the primary goal to study the way of bridging the gap between observations and concepts in a scientific perspective. Still, we can learn a lot from the heuristic solutions that are created to assist the human analyst performing a recognition task. There are many systems that directly try to imitate the decision making process of a human expert, such as an operator guarding a chemical process, an inspector supervising the quality of industrial production or a medical doctor deriving a diagnosis from a series of medical tests. On the basis of systematic interviews the decision making can become explicit and imitated by a computer program: an **expert system** [54]. The possibility to improve such a system by learning from examples is usually very limited and restricted to logical inference that makes the rules as general as possible, and the estimation of the thresholds on observations. The latter is needed as the human expert is not always able to define exactly what he means, e.g. by 'an unusually high temperature'.

In order to relate knowledge to observations, which are measurements in automatic systems, it is often needed to relate **knowledge uncertainty** to imprecise, noisy, or generally invalid measurements. Several frameworks have been developed to this end, e.g. fuzzy systems [74], Bayesian belief networks [42] and reasoning under uncertainty [82]. Characteristic for these approaches is that the given knowledge is already structured and needs explicitly defined parameters of uncertainty. New observations may adapt these parameters by relating them to observational frequencies. The knowledge structure is not learned; it has to be given and is hard to modify. An essential problem is that the variability of the external observations may be probabilistic, but the uncertainty in knowledge is based on 'belief' or 'fuzzy' definitions. Combining them in a single mathematical framework is disputable [39].

In the above approaches either the general knowledge or the concept underlying a class of observations is directly modeled. In **structural pattern**

recognition [26, 65] the starting point is the description of the structure of a single object. This can be done in several ways, e.g. by strings, contour descriptions, time sequences or other order-dependent data. Grammars that are inferred from a collection of strings are the basis of a syntactical approach to pattern recognition [26]. The incorporation of probabilities, e.g. needed for modeling the measurement noise, is not straightforward. Another possibility is the use of graphs. This is in fact already a reduction since objects are decomposed into highlights or landmarks, possibly given by attributes and also their relations, which may be attributed as well. Inferring a language from graphs is already much more difficult than from strings. Consequently, the generalization from a set of objects to a class is usually done by finding typical examples, prototypes, followed by graph matching [5, 78] for classifying new objects.

Generalization in structural pattern recognition is not straightforward. It is often based on the comparison of object descriptions using the entire available training set (the nearest neighbor rule) or a selected subset (the nearest prototype rule). Application knowledge is needed for defining the representation (strings, graphs) as well as for the dissimilarity measure to perform graph matching [51, 7]. The generalization may rely on an analysis of the matrix of dissimilarities, used to determine prototypes. More advanced techniques using the dissimilarity matrix will be described later.

The **1-Nearest-Neighbor Rule** (1-NN) is the simplest and most natural classification rule. It should always be used as a reference. It has a good asymptotic performance for metric measures [10, 14], not worse than twice the Bayes error, i.e. the lowest error possible. It works well in practice for finite training sets. Fig. 3 shows how it performs on the Iris data set in comparison to the linear and quadratic classifiers based on the assumption of normal

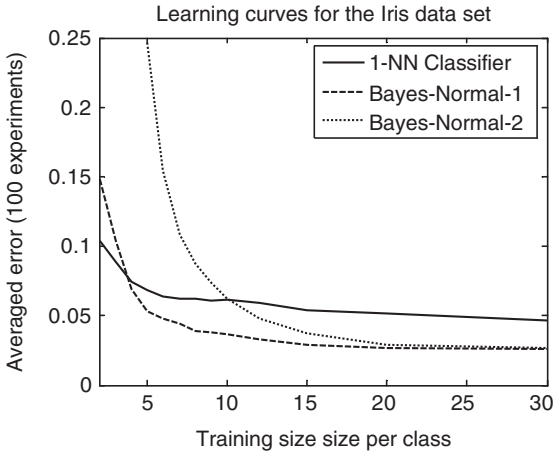


Fig. 3. Learning curves for Iris data

distributions [27]. The k-NN rule, based on a class majority vote over the k nearest neighbors in the training set, is, like the Parzen classifier, even Bayes consistent. These classifiers approximate the Bayes error for increasing training sets [14, 27].

However, such results heavily rely on the assumption that the training examples are identically and independently drawn (iid) from the same distribution as the future objects to be tested. This assumption of a fixed and stationary distribution is very strong, but it yields the best possible classifier. There are, however, other reasons, why it cannot be claimed that pattern recognition is solved by these statistical tools. The 1-NN and k-NN rules have to store the entire training set. The solution is thereby based on a comparison with all possible examples, including ones that are very similar, and asymptotically identical to the new objects to be recognized. By this, a class or a concept is not learned, as the decision relies on memorizing all possible instances. There is simply no generalization.

Other classification procedures, giving rise to two **learning curves** shown in Fig. 3, are based on specific model assumptions. The classifiers may perform well when the assumptions hold and may entirely fail, otherwise. An important observation is that models used in statistical learning procedures have almost necessarily a statistical formulation. Human knowledge, however, certainly in daily life, has almost nothing to do with statistics. Perhaps it is hidden in the human learning process, but it is not explicitly available in the context of human recognition. As a result, there is a need to look for effective model assumptions that are not phrased in statistical terms.

In Fig. 3 we can see that a more complex quadratic classifier performs initially worse than the other ones, but it behaves similarly to a simple linear classifier for large training sets. In general, complex problems may be better solved by complex procedures. This is illustrated in Fig. 4, in which the resulting error curves are shown as functions of complexity and training size. Like in Fig. 3, small training sets require simple classifiers. Larger training sets may be used to train more complex classifiers, but the error will increase, if pushed too far. This is a well-known and frequently studied phenomenon in

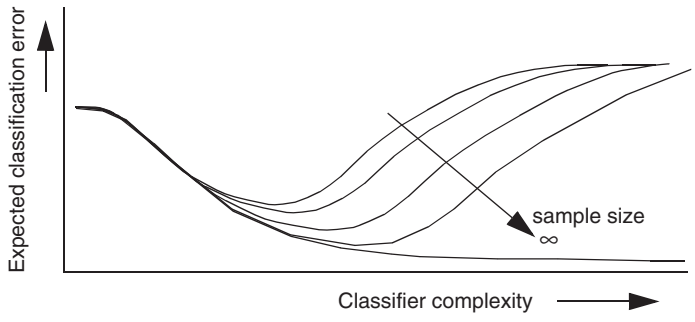


Fig. 4. Curse of dimensionality

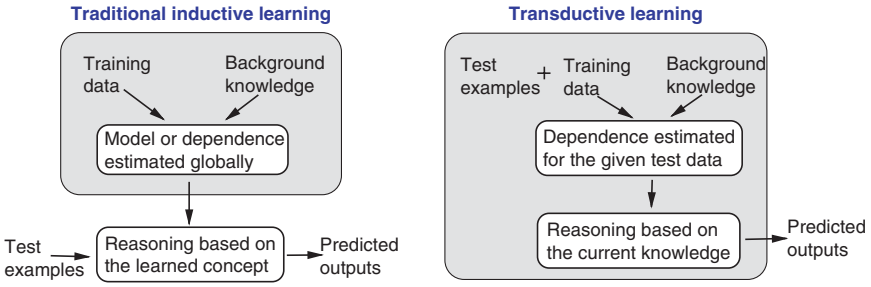


Fig. 5. Inductive (left) and transductive (right) learning paradigms; see also [8]. Background knowledge is here understood in terms of properties of the representations and the specified assumptions on a set of learning algorithms and related parameters

relation to the dimensionality (complexity) of the problem. Objects described by many features often rely on complex classifiers, which may thereby lead to worse results if the number of training examples is insufficient. This is the **curse of dimensionality**, also known as the Rao's paradox or the peaking phenomenon [44, 45]. It is caused by the fact that the classifiers badly generalize, due to a poor estimation of their parameters or their focus/adaptation to the noisy information or irrelevant details in the data. The same phenomenon can be observed while training complex neural networks without taking proper precautions. As a result, they will adapt to accidental data configurations, hence they will *overtrain*. This phenomenon is also well known outside the pattern recognition field. For instance, it is one of the reasons one has to be careful with extensive mass screening in health care: the more diseases and their relations are considered (the more complex the task), the more people will we be unnecessarily sent to hospitals for further examinations.

An important conclusion from this research is that the cardinality of the set of examples from which we want to infer a pattern concept bounds the complexity of the procedure used for generalization. Such a method should be simple if there are just a few examples. A somewhat complicated concept can only be learnt if sufficient prior knowledge is available and incorporated in such a way that the simple procedure is able to benefit from it.

An extreme consequence of the lack of prior knowledge is given by Watanabe as the **Ugly Duckling Theorem** [75]. Assume that objects are described by a set of atomic properties and we consider predicates consisting of all possible logic combinations of these properties in order to train a pattern recognition system. Then, all pairs of objects are equally similar in terms of the number of predicates they share. This is caused by the fact that all atomic properties, their presence as well as their absence, have initially equal weights. As a result, the training set is of no use. Summarized briefly, if we do not know anything about the problem we cannot learn (generalize and/or infer) from observations.

An entirely different reasoning pointing to the same phenomenon is the **No-Free-Lunch Theorem** formulated by Wolpert [81]. It states that all classifiers perform equally well if averaged over all possible classification problems. This also includes a random assignment of objects to classes. In order to understand this theorem it should be realized that the considered set of all possible classification problems includes all possible ways a given data set can be distributed over a set of classes. This again emphasizes that learning cannot be successful without any preference or knowledge.

In essence, it has been established that without prior or background knowledge, no learning, no generalization from examples is possible. Concerning specific applications based on strong models for the classes, it has been shown that additional observations may lower the specified gaps or solve uncertainties in these models. In addition, if these uncertainties are formulated in statistical terms, it will be well possible to diminish their influence by a statistical analysis of the training set. It is, however, unclear what the minimum prior knowledge is that is necessary to make the learning from examples possible. This is of interest if we want to uncover the roots of concept formation, such as learning of a class from examples. There exists one principle, formulated at the very beginning of the study of automatic pattern recognition, which may point to a promising direction. This is the principle of **compactness** [1], also phrased as a compactness hypothesis. It states that we can only learn from examples or phenomena if their representation is such that small variations in these examples cause small deviations in the representation. This demands that the representation is based on a continuous transformation of the real world objects or phenomena. Consequently, it is assumed that a sufficiently small variation in the original object will not cause the change of its class membership. It will still be a realization of the same concept. Consequently, we may learn the class of objects that belong to the same concept by studying the domain of their corresponding representations.

The Ugly Duckling Theorem deals with discrete logical representations. These cannot be solved by the compactness hypothesis unless some metric is assumed that replaces the similarity measured by counting differences in predicates. The No-Free-Lunch Theorem clearly violates the compactness assumption as it makes object representations with contradictory labelings equally probable. In practice, however, we encounter only specific types of problems.

Building proper **representations** has become an important issue in pattern recognition [20]. For a long time this idea has been restricted to the reduction of overly large feature sets to the sizes for which generalization procedures can produce significant results, given the cardinality of the training set. Several procedures have been studied based on feature selection as well as linear and nonlinear feature extraction [45]. A pessimistic result was found that about any hierarchical ordering of (sub)space separability that fulfills the necessary monotonicity constraints can be constructed by an example based on normal distributions only [11]. Very advanced procedures are needed to find such 'hidden' subspaces in which classes are well separable [61]. It has to

be doubted, however, whether such problems arise in practice, and whether such feature selection procedures are really necessary in problems with finite sample sizes. This doubt is further supported by an insight that feature reduction procedures should rely on global and not very detailed criteria if their purpose is to reduce the high dimensionality to a size which is in agreement with the given training set.

Feed-forward neural networks are a very general tool that, among others, offer the possibility to train a single system built between sensor and classification [4, 41, 62]. They thereby cover the representation step in the input layer(s) and the generalization step in the output layer(s). These layers are simultaneously optimized. The number of neurons in the network should be sufficiently large to make the interesting optima tractable. This, however, brings the danger of overtraining. There exist several ways to prevent that by incorporating some regularization steps in the optimization process. This replaces the adaptation step in Fig. 1. A difficult point here, however, is that it is not sufficiently clear how to choose regularization of an appropriate strength. The other important application of neural networks is that the use of various regularization techniques enables one to control the nonlinearity of the resulting classifier. This gives also a possibility to use not only complex, but also moderately nonlinear functions. Neural networks are thereby one of the most general tools for building pattern recognition systems.

In statistical learning, Vapnik has rigorously studied the problem of adapting the complexity of the generalization procedure to a finite training set [72, 73]. The resulting **Vapnik-Chervonenkis** (VC) dimension, a complexity measure for a family of classification functions, gives a good insight into the mechanisms that determine the final performance (which depends on the training error and the VC dimension). The resulting error bounds, however, are too general for a direct use. One of the reasons is that, like in the No-Free-Lunch Theorem, the set of classification problems (positions and labeling of the data examples) is not restricted to the ones that obey the compactness assumption.

One of the insights gained by studying the complexity measures of polynomial functions is that they have to be as simple as possible in terms of the number of their free parameters to be optimized. This was already realized by Cover in 1965 [9]. Vapnik extended this finding around 1994 to arbitrary non-linear classifiers [73]. In that case, however, the number of free parameters is not necessarily indicative for the complexity of a given family of functions, but the VC dimension is. In Vapnik's terms, the VC dimension reflects the flexibility of a family of functions (such as polynomials or radial basis functions) to separate arbitrarily labeled and positioned n-element data in a vector space of a fixed dimension. This VC dimension should be finite and small to guarantee the good performance of the generalization function.

This idea was elegantly incorporated to the **Support Vector Machine** (SVM) [73], in which the number of parameters is as small as a suitably determined subset of the training objects (the support vectors) and into

independent of the dimensionality of the vector space. One way to phrase this principle is that the structure of the classifier itself is simplified as far as possible (following the Occam's razor principle). So, after a detor along huge neural networks possibly having many more parameters than training examples, pattern recognition was back to the *small-is-beautiful* principle, but now better understood and elegantly formulated.

The use of **kernels** largely enriched the applicability of the SVM to nonlinear decision functions [66, 67, 73]. The kernel approach virtually generates nonlinear transformations of the combinations of the existing features. By using the representer theorem, a linear classifier in this nonlinear feature space can be constructed, because the kernel encodes generalized inner products of the original vectors only. Consequently, well-performing nonlinear classifiers built on training sets of almost any size in almost any feature space can be computed by using the SVM in combination with the 'kernel trick' [66].

This method has still a few limitations, however. It was originally designed for separable classes, hence it suffers when high overlap occurs. The use of slack variables, necessary for handling such an overlap, leads to a large number of support vectors and, consequently, to a large VC dimension. In such cases, other learning procedures have to be preferred. Another difficulty is that the class of admissible kernels is very narrow to guarantee the optimal solution. A kernel K has to be (conditionally) positive semidefinite (cpd) functions of two variables as only then it can be interpreted as a generalized inner product in reproducing kernel Hilbert space induced by K. Kernels were first considered as functions in Euclidean vector spaces, but they are now also designed to handle more general representations. **Special-purpose kernels** are defined in a number of applications such as text processing and shape recognition, in which good features are difficult to obtain. They use background knowledge from the application in which similarities between objects are defined in such a way that a proper kernel can be constructed. The difficulty is, again, the strong requirement of kernels as being cpd.

The next step is the so-called **dissimilarity representation** [56] in which general proximity measures between objects can be used for their representation. The measure itself may be arbitrary, provided that it is meaningful for the problem. Proximity plays a key role in the quest for an integrated structural and statistical learning model, since it is a natural bridge between these two approaches [6, 56]. Proximity is the basic quality to capture the characteristics of a set objects forming a group. It can be defined in various ways and contexts, based on sensory measurements, numerical descriptions, sequences, graphs, relations and other non-vectorial representations, as well as their combinations. A representation based on proximities is, therefore, universal.

Although some foundations are laid down [56], the ways for effective learning from general proximity representations are still to be developed. Since measures may not belong to the class of permissable kernels, the traditional SVM, as such, cannot be used. There exist alternative interpretations of

indefinite kernels and their relation to pseudo-Euclidean and Krein spaces [38, 50, 55, 56, 58], in which learning is possible for **non-Euclidean representations**. In general, proximity representations are embedded into suitable vector spaces equipped with a generalized inner product or norm, in which numerical techniques can either be developed or adapted from the existing ones. It has been experimentally shown that many classification techniques may perform well for general dissimilarity representations.

4 Perspectives

Pattern recognition deals with discovering, distinguishing, detecting or characterizing patterns present in the surrounding world. It relies on extraction and representation of information from the observed data, such that after integration with background knowledge, it ultimately leads to a formulation of new knowledge and concepts. The result of learning is that the knowledge already captured in some formal terms is used to describe the present interdependencies such that the relations between patterns are better understood (interpreted) or used for generalization. The latter means that a concept, e.g. of a class of objects, is formalized such that it can be applied to unseen examples of the same domain, inducing new information, e.g. the class label of a new object. In this process new examples should obey the same deduction process as applied to the original examples.

In the next subsections we will first recapitulate the elements of logical reasoning that contribute to learning. Next, this will be related to the Platonic and Aristotelian scientific approaches discussed in Section 2. Finally, two novel pattern recognition paradigms are placed in this view.

4.1 Learning by Logical Reasoning

Learning from examples is an active process of concept formation that relies on abstraction (focus on important characteristics or reduction of detail) and analogy (comparison between different entities or relations focussing on some aspect of their similarity). Learning often requires dynamical, multi-level (seeing the details leading to unified concepts, which further build higher level concepts) and possibly multi-strategy actions (e.g. in order to support good predictive power as well as interpretability). A learning task is basically defined by input data (design set), background knowledge or problem context and a learning goal [52]. Many inferential strategies need to be synergetically integrated to be successful in reaching this goal. The most important ones are inductive, deductive and abductive principles, which are briefly presented next. More formal definitions can be sought in the literature on formal logic, philosophy or e.g. in [23, 40, 52, 83].

Inductive reasoning is the *synthetic* inference process of arriving at a conclusion or a general rule from a limited set of observations. This relies on

a formation of a concept or a model, given the data. Although such a derived inductive conclusion cannot be proved, its reliability is supported by empirical observations. As along as the related deductions are not in contradiction with experiments, the inductive conclusion remains valid. If, however, future observations lead to contradiction, either an adaption or a new inference is necessary to find a better rule. To make it more formal, induction learns a general rule R (concerning A and B) from numerous examples of A and B. In practice, induction is often realized in a quantitative way. Its strength relies then on probability theory and the law of large numbers, in which given a large number of cases, one can describe their properties in the limit and the corresponding rate of convergence.

Deductive reasoning is the *analytic* inference process in which existing knowledge of known facts or agreed-upon rules is used to derive a conclusion. Such a conclusion does not yield 'new' knowledge since it is a logical consequence of what has already been known, but implicitly (it is not of a greater generality than the premises). Deduction, therefore, uses a logical argument to make explicit what has been hidden. It is also a valid form of proof provided that one starts from true premises. It has a predictive power, which makes it complementary to induction. In a pattern recognition system, both evaluation and prediction rely on deductive reasoning. To make it more formal, let us assume that A is a set of observations, B is a conclusion and R is a general rule. Let B be a logical consequence of A and R, i.e. $(A \wedge R) \models B$, where $\models$ denotes entailment. In a deductive reasoning, given A and using the rule R, the consequence B is derived.

Abductive reasoning is the *constructive* process of deriving the most likely or best explanations of known facts. This is a creative process, in which possible and feasible hypotheses are generated for a further evaluation. Since both abduction and induction deal with incomplete information, induction may be viewed in some aspects as abduction and vice versa, which leads to some confusion between these two [23, 52]. Here, we assume they are different. Concerning the entailment $(A \wedge R) \models B$, having observed the consequence B in the context of the rule R, A is derived to *explain* B.

In all learning paradigms there is an interplay between inductive, abductive and deductive principles. Both deduction and abduction make possible to conceptually understand a phenomenon, while induction verifies it. More precisely, abduction generates or reformulates new (feasible) ideas or hypotheses, induction justifies the validity of these hypothesis with observed data and deduction evaluates and tests them. Concerning pattern recognition systems, *abduction* explores data, transforms the representation and suggests feasible classifiers for the given problem. It also generates new classifiers or reformulates the old ones. Abduction is present in an initial exploratory step or in the Adaptation stage; see Fig. 1. Induction trains the classifier in the Generalization stage, while deduction predicts the final outcome (such as label) for the test data by applying the trained classifier in the Evaluation stage.

Since abduction is hardly emphasized in learning, we will give some more insights. In abduction, a peculiarity or an artifact is observed and a hypothesis is then created to explain it. Such a hypothesis is suggested based on existing knowledge or may extend it, e.g. by using analogy. So, the abductive process is creative and works towards new discovery. In data analysis, visualization facilitates the abductive process. In response to visual observations of irregularities or bizarre patterns, a researcher is inspired to look for clues that can be used to explain such an unexpected behavior. Mistakes and errors can therefore serve the purpose of discovery when strange results are inquired with a critical mind. Note, however, that this process is very hard to implement into automatic recognition systems as it would require to encode not only the detailed domain knowledge, but also techniques that are able to detect 'surprises' as well as strategies for their possible use. In fact, this requires a conscious interaction. Ultimately, only a human analyst can interactively respond in such cases, so abduction can be incorporated into semi-automatic systems well. In traditional pattern recognition systems, abduction is usually defined in the terms of data and works over pre-specified set of transformations, models or classifiers.

4.2 Logical Reasoning Related to Scientific Approaches

If pattern recognition (learning from examples) is merely understood as a process of concept formation from a set of observations, the inductive principle is the most appealing for this task. Indeed, it is the most widely emphasized in the literature, in which 'learning' is implicitly understood as 'inductive learning'. Such a reasoning leads to inferring new knowledge (rule or model) which is hopefully valid not only for the known examples, but also for novel, unseen objects. Various validation measures or adaptation steps are taken to support the applicability of the determined model. Additionally, care has to be taken that the unseen objects obey the same assumptions as the original objects used in training. If this does not hold, such an empirical generalization becomes invalid. One should therefore exercise in critical thinking while designing a complete learning system. It means that one has to be conscious which assumptions are made and be able to quantify their sensibility, usability and validity with the learning goal.

On the other hand, deductive reasoning plays a significant role in the Platonic approach. This top-down scenario starts from a set of rules derived from expert knowledge on problem domain or from a degree of belief in a hypothesis. The existing prior knowledge is first formulated in appropriate terms. These are further used to generate inductive inferences regarding the validity of the hypotheses in the presence of observed examples. So, deductive formalism (description of the object's structure) or deductive predictions (based on the Bayes rule) precede inductive principles. A simple example in the Bayesian inference is the well-known Expectation-Maximization (EM) algorithm used in problems with incomplete data [13]. The EM algorithm iterates between the

E-step and M-step until convergence. In the E-step, given a current (or initial) estimate of the unknown variable, a conditional expectation is found, which is maximized in the M-step and derives a new estimate. The E-step is based on deduction, while the M-step relies on induction. In the case of Bayesian nets, which model a set of concepts (provided by an expert) through a network of conditional dependencies, predictions (deductions) are made from the (initial) hypotheses (beliefs over conditional dependencies) using the Bayes theorem. Then, inductive inferences regarding the hypotheses are drawn from the data. Note also that if the existing prior knowledge is captured in some rules, learning may become a simplification of these rules such that their logical combinations describe the problem.

In the Aristotelian approach to pattern recognition, observation of particulars and their explanation are essential for deriving a concept. As we already know, abduction plays a role here, especially for data exploration and characterization to explain or suggest a modification of the representation or an adaptation of the given classifier. Aristotelian learning often relies on the Occam's razor principle which advocates to choose the simplest model or hypothesis among otherwise equivalent ones and can be implemented in a number of ways [8].

In summary, the Platonic scenario is dominantly inductive-deductive, while the Aristotelian scenario is dominantly inductive-abductive. Both frameworks have different merits and shortcomings. The strength of the Platonic approach lies in the proper formulation and use of subjective beliefs, expert knowledge and possibility to encode internal structural organization of objects. It is model-driven. In this way, however, the inductive generalization becomes limited, as there may be little freedom in the description to explore and discovery of new knowledge. The strength of the Aristotelian approach lies in a numerical induction and a well-developed mathematical theory of vector spaces in which the actual learning takes place. It is data-driven. The weakness, however, lies in the difficulty to incorporate the expert or background knowledge about the problem. Moreover, in many practical applications, it is known that the implicit assumptions of representative training sets, identical and identically distributed (iid) samples as well as stationary distributions do not hold.

4.3 Two New Pattern Recognition Paradigms

Two far-reaching novel paradigms have been proposed that deal with the drawbacks of the Platonic and Aristotelian approaches. In the Aristotelian scenario, Vapnik has introduced *transductive* learning [73], while in the Platonic scenario, Goldfarb has advocated a new structural learning paradigm [31, 32]. We think these are two major perspectives of the science of pattern recognition.

Vapnik [73] 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.' In the traditional Aristotelian scenario, the learning task is often transformed to the problem of function estimation, in which a decision function is determined *globally* for the entire domain (e.g. for all possible examples in a feature vector space). This is, however, a solution to a more general problem than necessary to arrive at a conclusion (output) for *specific* input data. Consequently, the application of this common-sense principle requires a reformulation of the learning problem such that novel (unlabeled) examples are considered in the context of the given training set. This leads to the transductive principle which aims at estimating the output for a *given* input only when required and may differ from an instance to instance. The training sample, considered either globally, or in the local neighborhoods of test examples, is actively used to determine the output. As a result, this leads to confidence measures of single predictions instead of globally estimated classifiers. It provides ways to overcome the difficulty of iid samples and stationary distributions. More formally, in a transductive reasoning, given an entailment $A \models (B \cup C)$, if the consequence B is observed as the result of A, then the consequence C becomes more likely.

The truly transductive principle requires an active synergy of inductive, deductive and abductive principles in a conscious decision process. We believe it is practised by people who analyze complex situations, deduce and validate possible solutions and make decisions in novel ways. Examples are medical doctors, financial advisers, strategy planners or leaders of large organizations. In the context of automatic learning, transduction has applications to learning from partially labeled sets and otherwise missing information, information retrieval, active learning and all types of diagnostics. Some proposals can be found e.g. in [34, 46, 47, 73]. Although many researchers recognize the importance of this principle, many remain also reluctant. This may be caused by unfamiliarity with this idea, few existing procedures, or by the accompanying computational costs as a complete decision process has to be constantly inferred anew.

In the Platonic scenario, Goldfarb and his colleagues have developed structural inductive learning, realized by the so-called evolving transformation systems (ETS) [31, 32]. Goldfarb first noticed the intrinsic and impossible to overcome inadequacy of vector spaces to truly learn from examples [30]. The reason is that such quantitative representations loose all information on object structure; there is no way an object can be generated given its numeric encoding. The second crucial observation was that all objects in the universe have a formative history. This led Goldfarb to the conclusion that an object representation should capture the object's formative evolution, i.e. the way the object is created through a sequence of suitable transformations in time. The creation process is only possible through structural operations. So, *'the resulting representation embodies temporal structural information in the form of a formative, or generative, history'* [31]. Consequently, objects are treated as evolving structural processes and a class is defined by structural processes,

which are 'similar'. This is an inductive structural/symbolic class representation, the central concept in ETS. This representation is learnable from a (small) set of examples and has the capability to generate objects from the class.

The generative history of a class starts from a single progenitor and is encoded as a multi-level hierarchical system. On a given level, the basic structural elements are defined together with their structural transformations, such that both are used to constitute a new structural element on a higher level. This new element becomes meaningful on that level. Similarity plays an important role, as it is used as a basic quality for a class representation as a set of similar structural processes. Similarity measure is learned in training to induce the optimal finite set of weighted structural transformations that are necessary on the given level, such that the similarity of an object to the class representation is large. 'This mathematical structure allows one to capture dynamically, during the learning process, the compositional structure of objects/events within a given inductive, or evolutionary, environment' [31].

Goldfarb's ideas bear some similarity to the ones of Wolfram, presented in his book on 'a new kind of science' [80]. Wolfram considers computation as the primary concept in nature; all processes are the results of cellular-automata[6] type of computational processes, and thereby inherently numerical. He observes that repetitive use of simple computational transformations can cause very complex phenomena, especially if computational mechanisms are used at different levels. Goldfarb also discusses dynamical systems, in which complexity is built from simpler structures, through hierarchical folding up (or enrichment). The major difference is that he considers structure of primary interest, which leads to evolving temporal structural processes instead of computational ones.

In summary, Goldfarb proposes a revolutionary paradigm: an ontological model of a class representation in an epistemological context, as it is learnable from examples. This is a truly unique unification. We think it is the most complete and challenging approach to pattern recognition to this date, a breakthrough. By including the formative history of objects into their representation, Goldfarb attributes them some aspects of human consciousness. The far reaching consequence of his ideas is a generalized measurement process that will be one day present in sensors. Such sensors will be able to measure 'in structural units' instead of numerical units (say, meters) as it is currently done. The inductive process over a set of structural units lies at the foundation of new inductive informatics. The difficulty, however, is that the current formalism in mathematics and related fields is not yet prepared for adopting these far-reaching ideas. We, however, believe, they will pave the road and be found anew or rediscovered in the next decennia.

[6] Cellular automata are discrete dynamical systems that operate on a regular lattice in space and time, and are characterized by 'local' interactions.

5 Challenges

A lot of research effort is needed before the two novel and far-reaching paradigms are ready for practical applications. So, this section focuses on several challenges that naturally come in the current context and will be summarized for the design of automatic pattern recognition procedures. A number of fundamental problems, related to the various approaches, have already been identified in the previous sections and some will return here on a more technical level. Many of the points raised in this section have been more extensively discussed in [17]. We will emphasize these which have only been touched or are not treated in the standard books [15, 71, 76] or in the review by Jain et al. [45]. The issues to be described are just a selection of the many which are not yet entirely understood. Some of them may be solved in the future by the development of novel procedures or by gaining an additional understanding. Others may remain an issue of concern to be dealt with in each application separately. We will be systematically describe them, following the line of advancement of a pattern recognition system; see also Fig. 1:

- **Representation and background knowledge.** This is the way in which individual real world objects and phenomena are numerically described or encoded such that they can be related to each other in some meaningful mathematical framework. This framework has to allow the generalization to take place.
- **Design set.** This is the set of objects available or selected to develop the recognition system.
- **Adaptation.** This is usually a transformation of the representation such that it becomes more suitable for the generalization step.
- **Generalization.** This is the step in which objects of the design set are related such that classes of objects can be distinguished and new objects can be accurately classified.
- **Evaluation.** This is an estimate of the performance of a developed recognition system.

5.1 Representation and Background Knowledge

The problem of representation is a core issue for pattern recognition [18, 20]. Representation encodes the real world objects by some numerical description, handled by computers in such a way that the individual object representations can be interrelated. Based on that, later a generalization is achieved, establishing descriptions or discriminations between classes of objects. Originally, the issue of representation was almost neglected, as it was reduced to the demand of having discriminative features provided by some expert. Statistical learning is often believed to start in a given feature vector space. Indeed, many books on pattern recognition disregard the topic of representation, simply by assuming that objects are somehow already represented [4, 62]. A systematic study on

representation [20, 56] is not easy, as it is application or domain-dependent (where the word *domain* refers to the nature or character of problems and the resulting type of data). For instance, the representations of a time signal, an image of an isolated 2D object, an image of a set of objects on some background, a 3D object reconstruction or the collected set of outcomes of a medical examination are entirely different observations that need individual approaches to find good representations. Anyway, if the starting point of a pattern recognition problem is not well defined, this cannot be improved later in the process of learning. It is, therefore, of crucial importance to study the representation issues seriously. Some of them are phrased in the subsequent sections.

The use of vector spaces. Traditionally, objects are represented by vectors in a feature vector space. This representation makes it feasible to perform some generalization (with respect to this linear space), e.g. by estimating density functions for classes of objects. The object structure is, however, lost in such a description. If objects contain an inherent, identifiable structure or organization, then relations between their elements, like relations between neighboring pixels in an image, are entirely neglected. This also holds for spatial properties encoded by Fourier coefficients or wavelets weights. These original structures may be partially rediscovered by deriving statistics over a set of vectors representing objects, but these are not included in the representation itself. One may wonder whether the representation of objects as vectors in a space is not oversimplified to be able to reflect the nature of objects in a proper way. Perhaps objects might be better represented by convex bodies, curves or by other structures in a metric vector space. The generalization over sets of vectors, however, is heavily studied and mathematically well developed. How to generalize over a set of other structures is still an open question.

The essential problem of the use of vector spaces for object representation is originally pointed out by Goldfarb [30, 33]. He prefers a structural representation in which the original object organization (connectedness of building structural elements) is preserved. However, as a generalization procedure for structural representations does not exist yet, Goldfarb starts from the evolving transformation systems [29] to develop a novel system [31]. As already indicated in Sec. 4.3 we see this as a possible direction for a future breakthrough.

Compactness. An important, but seldom explicitly identified property of representations is compactness [1]. In order to consider classes, which are bounded in their domains, the representation should be constrained: objects that are similar in reality should be close in their representations (where the closeness is captured by an appropriate relation, possibly a proximity measure). If this demand is not satisfied, objects may be described capriciously and, as a result, no generalization is possible. This compactness assumption puts some restriction on the possible probability density functions used to describe classes in a representation vector space. This, thereby, also narrows

the set of possible classification problems. A formal description of the probability distribution of this set may be of interest to estimate the expected performance of classification procedures for an arbitrary problem.

In Sec. 3, we pointed out that the lack of a formal restriction of pattern recognition problems to those with a compact representation was the basis of pessimistic results like the No-Free-Lunch Theorem [81] and the classification error bounds resulting from the VC complexity measure [72, 73]. One of the main challenges for pattern recognition to find a formal description of compactness that can be used in error estimators the average over the set of possible pattern recognition problems.

Representation types. There exists numerous ways in which representations can be derived. The basic 'numerical' types are now distinguished as:

- *Features.* Objects are described by characteristic attributes. If these attributes are continuous, the representation is usually compact in the corresponding feature vector space. Nominal, categorical or ordinal attributes may cause problems. Since a description by features is a reduction of objects to vectors, different objects may have identical representations, which may lead to class overlap.
- *Pixels or other samples.* A complete representation of an object may be approximated by its sampling. For images, these are pixels, for time signals, these are time samples and for spectra, these are wavelengths. A pixel representation is a specific, boundary case of a feature representation, as it describes the object properties in each point of observation.
- *Probability models.* Object characteristics may be reflected by some probabilistic model. Such models may be based on expert knowledge or trained from examples. Mixtures of knowledge and probability estimates are difficult, especially for large models.
- *Dissimilarities, similarities or proximities.* Instead of an absolute description by features, objects are relatively described by their dissimilarities to a collection of specified objects. These may be carefully optimized prototypes or representatives for the problem, but also random subsets may work well [56]. The dissimilarities may be derived from raw data, such as images, spectra or time samples, from original feature representations or from structural representations such as strings or relational graphs. If the dissimilarity measure is nonnegative and zero only for two identical objects, always belonging to the same class, the class overlap may be avoided by dissimilarity representations.
- *Conceptual representations.* Objects may be related to classes in various ways, e.g. by a set of classifiers, each based on a different representation, training set or model. The combined set of these initial classifications or clusterings constitute a new representation [56]. This is used in the area of combining clusterings [24, 25] or combining classifiers [49].

In the structural approaches, objects are represented in qualitative ways. The most important are strings or sequences, graphs and their collections and hierarchical representations in the form of ontological trees or semantic nets.

Vectorial object descriptions and proximity representations provide a good way for generalization in some appropriately determined spaces. It is, however, difficult to integrate them with the detailed prior or background knowledge that one has on the problem. On the other hand, probabilistic models and, especially, structural models are well suited for such an integration. The later, however, constitute a weak basis for training general classification schemes. Usually, they are limited to assigning objects to the class model that fits best, e.g. by the nearest neighbor rule. Other statistical learning techniques are applied to these if given an appropriate proximity measure or a vectorial representation space found by graph embeddings [79].

It is a challenge to find representations that constitute a good basis for modeling object structure and which can also be used for generalizing from examples. The next step is to find representations not only based on background knowledge or given by the expert, but to learn or optimize them from examples.

5.2 Design Set

A pattern recognition problem is not only specified by a representation, but also by the set of examples given for training and evaluating a classifier in various stages. The selection of this set and its usage strongly influence the overall performance of the final system. We will discuss some related issues.

Multiple use of the training set. The entire design set or its parts are used in several stages during the development of a recognition system. Usually, one starts from some exploration, which may lead to the removal of wrongly represented or erroneously labeled objects. After gaining some insights into the problem, the analyst may select a classification procedure based on the observations. Next, the set of objects may go through some preprocessing and normalization. Additionally, the representation has to be optimized, e.g. by a feature/object selection or extraction. Then, a series of classifiers has to be trained and the best ones need to be selected or combined. An overall evaluation may result in a re-iteration of some steps and different choices.

In this complete process the same objects may be used a number of times for the estimation, training, validation, selection and evaluation. Usually, an expected error estimation is obtained by a cross-validation or hold-out method [32, 77]. It is well known that the multiple use of objects should be avoided as it biases the results and decisions. Re-using objects, however, is almost unavoidable in practice. A general theory does not exist yet, that predicts how much a training set is 'worn-out' by its repetitive use and which suggests corrections that can diminish such effects.

Representativeness of the training set. Training sets should be representative for the objects to be classified by the final system. It is common to take a randomly selected subset of the latter for training. Intuitively, it seems to be useless to collect many objects represented in the regions where classes do not overlap. On the contrary, in the proximity of the decision boundary, a higher sampling rate seems to be advantageous. This depends on the complexity of the decision function and the expected class overlap, and is, of course, inherently related to the chosen procedure.

Another problem are the unstable, unknown or undetermined class distributions. Examples are the impossibility to characterize the class of non-faces in the face detection problem, or in machine diagnostics, the probability distribution of all casual events if the machine is used for undetermined production purposes. A training set that is representative for the class distributions cannot be found in such cases. An alternative may be to sample the *domain* of the classes such that all possible object occurrences are approximately covered. This means that for any object that could be encountered in practice there exists a sufficiently similar object in the training set, defined in relation to the specified class differences. Moreover, as class density estimates cannot be derived for such a training set, class posterior probabilities cannot be estimated. For this reason such a type of domain based sampling is only appropriate for non-overlapping classes. In particular, this problem is of interest for non-overlapping (dis)similarity based representations [18].

Consequently, we wonder whether it is possible to use a more general type of sampling than the classical iid sampling, namely the domain sampling. If so, the open questions refer to the verification of dense samplings and types of new classifiers that are explicitly built on such domains.

5.3 Adaptation

Once a recognition problem has been formulated by a set of example objects in a convenient representation, the generalization over this set may be considered, finally leading to a recognition system. The selection of a proper generalization procedure may not be evident, or several disagreements may exist between the realized and preferred procedures. This occurs e.g. when the chosen representation needs a non-linear classifier and only linear decision functions are computationally feasible, or when the space dimensionality is high with respect to the size of the training set, or the representation cannot be perfectly embedded in a Euclidean space, while most classifiers demand that. For reasons like these, various adaptations of the representation may be considered. When class differences are explicitly preserved or emphasized, such an adaptation may be considered as a part of the generalization procedure. Some adaptation issues that are less connected to classification are discussed below.

Problem complexity. In order to determine which classification procedures might be beneficial for a given problem, Ho and Basu [43] proposed

to investigate its complexity. This is an ill-defined concept. Some of its aspects include data organization, sampling, irreducibility (or redundancy) and the interplay between the local and global character of the representation and/or of the classifier. Perhaps several other attributes are needed to define complexity such that it can be used to indicate a suitable pattern recognition solution to a given problem; see also [2].

Selection or combining. Representations may be complex, e.g. if objects are represented by a large amount of features or if they are related to a large set of prototypes. A collection of classifiers can be designed to make use of this fact and later combined. Additionally, also a number of representations may be considered simultaneously. In all these situations, the question arises on which should be preferred: a selection from the various sources of information or some type of combination. A selection may be random or based on a systematic search for which many strategies and criteria are possible [49]. Combinations may sometimes be fixed, e.g. by taking an average, or a type of a parameterized combination like a weighted linear combination as a principal component analysis; see also [12, 56, 59].

The choice favoring either a selection or combining procedure may also be dictated by economical arguments, or by minimizing the amount of necessary measurements, or computation. If this is unimportant, the decision has to be made according to the accuracy arguments. Selection neglects some information, while combination tries to use everything. The latter, however, may suffer from overtraining as weights or other parameters have to be estimated and may be adapted to the noise or irrelevant details in the data. The sparse solutions offered by support vector machines [67] and sparse linear programming approaches [28, 35] constitute a way of compromise. How to optimize them efficiently is still a question.

Nonlinear transformations and kernels. If a representation demands or allows for a complicated, nonlinear solution, a way to proceed is to transform the representation appropriately such that linear aspects are emphasized. A simple (e.g. linear) classifier may then perform well. The use of kernels, see Sec. 3, is a general possibility. In some applications, indefinite kernels are proposed as being consistent with the background knowledge. They may result in non-Euclidean dissimilarity representations, which are challenging to handle; see [57] for a discussion.

5.4 Generalization

The generalization over sets of vectors leading to class descriptions or discriminants was extensively studied in pattern recognition in the 60's and 70's of the previous century. Many classifiers were designed, based on the assumption of normal distributions, kernels or potential functions, nearest neighbor rules, multi-layer perceptrons, and so on [15, 45, 62, 76]. These types of studies were later extended by the fields of multivariate statistics, artificial neural networks

and machine learning. However, in the pattern recognition community, there is still a high interest in the classification problem, especially in relation to practical questions concerning issues of combining classifiers, novelty detection or the handling of ill-sampled classes.

Handling multiple solutions. Classifier selection or classifier combination. Almost any more complicated pattern recognition problem can be solved in multiple ways. Various choices can be made for the representation, the adaptation and the classification. Such solutions usually do not only differ in the total classification performance, they may also make different errors. Some type of combining classifiers will thereby be advantageous [49]. It is to be expected that in the future most pattern recognition systems for real world problems are constituted of a set of classifiers. In spite of the fact that this area is heavily studied, a general approach on how to select, train and combine solutions is still not available. As training sets have to be used for optimizing several subsystems, the problem how to design complex systems is strongly related to the above issue of multiple use of the training set.

Classifier typology. Any classification procedure has its own explicit or built-in assumptions with respect to data inherent characteristics and the class distributions. This implies that a procedure will lead to relatively good performance if a problem fulfils its exact assumptions. Consequently, any classification approach has its problem for which it is the best. In some cases such a problem might be far from practical application. The construction of such problems may reveal which typical characteristics of a particular procedure are. Moreover, when new proposals are to be evaluated, it may be demanded that some examples of its corresponding typical classification problem are published, making clear what the area of application may be; see [19].

Generalization principles. The two basic generalization principles, see Section 4, are probabilistic inference, using the Bayes-rule [63] and the minimum description length principle that determines the most simple model in agreement with the observations (based on Occam's razor) [37]. These two principles are essentially different[7]. The first one is sensitive to multiple copies of an existing object in the training set, while the second one is not. Consequently, the latter is not based on densities, but just on object differences or distances. An important issue is to find in which situations each of these principle should be recommended and whether the choice should be made in the beginning, in the selection of the design set and the way of building a representation, or it should be postpone until a later stage.

The use of unlabeled objects and active learning. The above mentioned principles are examples of statistical inductive learning, where a classifier is

[7] Note that Bayesian inference is also believed to implement the Occam's razor [8] in which preference for simpler models is encoded by encouraging particular prior distributions. This is, however, not the primary point as it is in the minimum description length principle.

induced based on the design set and it is later applied to unknown objects. The disadvantage of such approach is that a decision function is in fact designed for all possible representations, whether valid or not. Transductive learning, see Section 4.3, is an appealing alternative as it determines the class membership only for the objects in question, while relying on the collected design set or its suitable subset [73]. The use of unlabeled objects, not just the one to be classified, is a general principle that may be applied in many situations. It may improve a classifier based on just a labeled training set. If this is understood properly, the classification of an entire test set may yield better results than the classification of individuals.

Classification or class detection. Two-class problems constitute the traditional basic line in pattern recognition, which reduces to finding a discriminant or a binary decision function. Multi-class problems can be formulated as a series of two-class problems. This can be done in various ways, none of them is entirely satisfactory. An entirely different approach is the description of individual classes by so-called one-class classifiers [69, 70]. In this way the focuss is given to class description instead of to class separation. This brings us to the issue of the structure of a class.

Traditionally classes are defined by a distribution in the representation space. However, the better such a representation, the higher its dimensionality, the more difficult it is to estimate a probability density function. Moreover, as we have seen above, it is for some applications questionable whether such a distribution exist. A class is then a part of a possible non-linear manifold in a high-dimensional space. It has a structure instead of a density distribution. It is a challenge to use this approach for building entire pattern recognition systems.

5.5 Evaluation

Two questions are always apparent in the development of recognition systems. The first refers to the overall performance of a particular system once it is trained, and has sometimes a definite answer. The second question is more open and asks which good recognition procedures are in general.

Recognition system performance. Suitable criteria should be used to evaluate the overall performance of the entire system. Different measures with different characteristics can be applied, however, usually, only a single criterion is used. The basic ones are the average accuracy computed over all validation objects or the accuracy determined by the worst-case scenario. In the first case, we again assume that the set of objects to be recognized is well defined (in terms of distributions). Then, it can be sampled and the accuracy of the entire system is estimated based on the evaluation set. In this case, however, we neglect the issue that after having used this evaluation set together with the training set, a better system could have been found. A more interesting point is how to judge the performance of a system if the distribution of objects

is ill-defined or if a domain based classification system is used as discussed above. Now, the largest mistake that is made becomes a crucial factor for this type of judgements. One needs to be careful, however, as this may refer to an unimportant outlier (resulting e.g. from invalid measurements).

Practice shows that a single criterion, like the final accuracy, is insufficient to judge the overall performance of the whole system. As a result, multiple performance measures should be taken into account, possibly at each stage. These measures should not only reflect the correctness of the system, but also its flexibility to cope with unusual situations in which e.g. specific examples should be rejected or misclassification costs incorporated.

Prior probability of problems. As argued above, any procedure has a problem for which it performs well. So, we may wonder how large the class of such problems is. We cannot state that any classifier is better than any other classifier, unless the distribution of problems to which these classifiers will be applied is defined. Such distributions are hardly studied. What is done at most is that classifiers are compared over a collection of benchmark problems. Such sets are usually defined ad hoc and just serve as an illustration. The collection of problems to which a classification procedure will be applied is not defined. As argued in Section 3, it may be as large as all problems with a compact representation, but preferably not larger.

6 Discussion and Conclusions

Recognition of patterns and inference skills lie at the core of human learning. It is a human activity that we try to imitate by mechanical means. There are no physical laws that assign observations to classes. It is the human consciousness that groups observations together. Although their connections and interrelations are often hidden, some understanding may be gained in the attempt of imitating this process. The human process of learning patterns from examples may follow along the lines of trial and error. By freeing our minds of fixed beliefs and petty details we may not only understand single observations but also induce principles and formulate concepts that lie behind the observed facts. New ideas can be born then. These processes of abstraction and concept formation are necessary for development and survival. In practice, (semi-)automatic learning systems are built by imitating such abilities in order to gain understanding of the problem, explain the underlying phenomena and develop good predictive models.

It has, however, to be strongly doubted whether statistics play an important role in the human learning process. Estimation of probabilities, especially in multivariate situations is not very intuitive for majority of people. Moreover, the amount of examples needed to build a reliable classifier by statistical means is much larger than it is available for humans. In human recognition, proximity based on relations between objects seems to come before features

are searched and may be, thereby, more fundamental. For this reason and the above observation, we think that the study of proximities, distances and domain based classifiers are of great interest. This is further encouraged by the fact that such representations offer a bridge between the possibilities of learning in vector spaces and the structural description of objects that preserve relations between objects inherent structure. We think that the use of proximities for representation, generalization and evaluation constitute the most intriguing issues in pattern recognition.

The existing gap between structural and statistical pattern recognition partially coincides with the gap between knowledge and observations. Prior knowledge and observations are both needed in a subtle interplay to gain new knowledge. The existing knowledge is needed to guide the deduction process and to generate the models and possible hypotheses needed by induction, transduction and abduction. But, above all, it is needed to select relevant examples and a proper representation. If and only if the prior knowledge is made sufficiently explicit to set this environment, new observations can be processed to gain new knowledge. If this is not properly done, some results may be obtained in purely statistical terms, but these cannot be integrated with what was already known and have thereby to stay in the domain of observations. The study of automatic pattern recognition systems makes perfectly clear that learning is possible, only if the Platonic and Aristotelian scientific approaches cooperate closely. This is what we aim for.

References

[1] A.G. Arkadev and E.M. Braverman. *Computers and Pattern Recognition*. Thompson, Washington, DC, 1966.
[2] M. Basu and T.K. Ho, editors. *Data Complexity in Pattern Recognition*. Springer, 2006.
[3] R. Bergmann. *Developing Industrial Case-Based Reasoning Applications*. Springer, 2004.
[4] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
[5] H. Bunke. Recent developments in graph matching. In *International Conference on Pattern Recognition*, volume 2, pages 117–124, 2000.
[6] 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*, pages 1–11, 2001.
[7] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.
[8] V.S. Cherkassky and F. Mulier. *Learning from data: Concepts, Theory and Methods*. John Wiley & Sons, Inc., New York, NY, USA, 1998.

[9] T.M. Cover. Geomerical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14:326–334, 1965.
[10] T.M. Cover and P.E. Hart. Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.
[11] T.M. Cover and J.M. van Campenhout. On the possible orderings in the measurement selection problem. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-7(9):657–661, 1977.
[12] I.M. de Diego, J.M. Moguerza, and A. Muñoz. Combining kernel information for support vector classification. In *Multiple Classifier Systems*, pages 102–111. Springer-Verlag, 2004.
[13] 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.
[14] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, 1996.
[15] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, Inc., 2nd edition, 2001.
[16] R.P.W. Duin. Four scientific approaches to pattern recognition. In *Fourth Quinquennial Review 1996-2001. Dutch Society for Pattern Recognition and Image Processing*, pages 331–337. NVPHBV, Delft, 2001.
[17] R.P.W. Duin and E. Pękalska. Open issues in pattern recognition. In *Computer Recognition Systems*, pages 27–42. Springer, Berlin, 2005.
[18] 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.
[19] R.P.W. Duin, E. Pękalska, and D.M.J. Tax. The characterization of classification problems by classifier disagreements. In *International Conference on Pattern Recognition*, volume 2, pages 140–143, Cambridge, United Kingdom, 2004.
[20] 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.
[21] S. Edelman. *Representation and Recognition in Vision*. MIT Press, Cambridge, 1999.
[22] B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, London, 1993.
[23] P. Flach and A. Kakas, editors. *Abduction and Induction: essays on their relation and integration*. Kluwer Academic Publishers, 2000.
[24] A. Fred and A.K. Jain. Data clustering using evidence accumulation. In *International Conference on Pattern Recognition*, pages 276–280, Quebec City, Canada, 2002.

[25] A. Fred and A.K. Jain. Robust data clustering. In *Conf. on Computer Vision and Pattern Recognition*, pages 442 –451, Madison - Wisconsin, USA, 2002.
[26] K.S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice-Hall, 1982.
[27] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
[28] G.M. Fung and O.L. Mangasarian. A Feature Selection Newton Method for Support Vector Machine Classification. *Computational Optimization and Aplications*, 28(2):185–202, 2004.
[29] L. Goldfarb. On the foundations of intelligent processes – I. An evolving model for pattern recognition. *Pattern Recognition*, 23(6):595–616, 1990.
[30] 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.
[31] L. Goldfarb and D. Gay. What is a structural representation? Fifth variation. Technical Report TR05-175, University of New Brunswick, Fredericton, Canada, 2005.
[32] L. Goldfarb and O. Golubitsky. What is a structural measurement process? Technical Report TR01-147, University of New Brunswick, Fredericton, Canada, 2001.
[33] L. Goldfarb and J. Hook. Why classical models for pattern recognition are not pattern recognition models. In *International Conference on Advances in Pattern Recognition*, pages 405–414, 1998.
[34] T. Graepel, R. Herbrich, and K. Obermayer. Bayesian transduction. In *Advances in Neural Information System Processing*, pages 456–462, 2000.
[35] 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.
[36] U. Grenander. *Abstract Inference*. John Wiley & Sons, Inc., 1981.
[37] P. Grünwald, I.J. Myung, and Pitt M., editors. *Advances in Minimum Description Length: Theory and Applications*. MIT Press, 2005.
[38] B. Haasdonk. Feature space interpretation of SVMs with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):482–492, 2005.
[39] I. Hacking. *The emergence of probability*. Cambridge University Press, 1974.
[40] G. Harman and S. Kulkarni. *Reliable Reasoning: Induction and Statistical Learning Theory*. MIT Press, to appear.
[41] S. Haykin. *Neural Networks, a Comprehensive Foundation, second edition*. Prentice-Hall, 1999.
[42] D. Heckerman. A tutorial on learning with Bayesian networks. In M. Jordan, editor, *Learning in Graphical Models*, pages 301–354. MIT Press, Cambridge, MA, 1999.

[43] T.K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, 2002.
[44] 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.
[45] 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.
[46] T. Joachims. Transductive inference for text classification using support vector machines. In I. Bratko and S. Dzeroski, editors, *International Conference on Machine Learning*, pages 200–209, 1999.
[47] T. Joachims. Transductive learning via spectral graph partitioning. In *International Conference on Machine Learning*, 2003.
[48] T.S. Kuhn. *The Structure of Scientific Revolutions*. University of Chicago Press, 1970.
[49] L.I. Kuncheva. *Combining Pattern Classifiers. Methods and Algorithms*. Wiley, 2004.
[50] J. Laub and K.-R. Müller. Feature discovery in non-metric pairwise data. *Journal of Machine Learning Research*, pages 801–818, 2004.
[51] 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.
[52] R.S. Michalski. Inferential theory of learning as a conceptual basis for multistrategy learning. *Machine Learning*, 11:111–151, 1993.
[53] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
[54] Richard E. Neapolitan. *Probabilistic reasoning in expert systems: theory and algorithms*. John Wiley & Sons, Inc., New York, NY, USA, 1990.
[55] C.S. Ong, S. Mary, X.and Canu, and Smola A.J. Learning with non-positive kernels. In *International Conference on Machine Learning*, pages 639–646, 2004.
[56] E. Pękalska and R.P.W. Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, Singapore, 2005.
[57] E. Pękalska, R.P.W. Duin, S. Günter, and H. Bunke. On not making dissimilarities Euclidean. In *Joint IAPR International Workshops on SSPR and SPR*, pages 1145–1154. Springer-Verlag, 2004.
[58] E. Pękalska, P. Paclík, and R.P.W. Duin. A Generalized Kernel Approach to Dissimilarity Based Classification. *Journal of Machine Learning Research*, 2:175–211, 2002.
[59] E. Pękalska, M. Skurichina, and R.P.W. Duin. Combining Dissimilarity Representations in One-class Classifier Problems. In *Multiple Classifier Systems*, pages 122–133. Springer-Verlag, 2004.

[60] L.I. Perlovsky. Conundrum of combinatorial complexity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):666–670, 1998.
[61] P. Pudil, J. Novovićova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.
[62] B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.
[63] C.P. Robert. *The Bayesian Choice*. Springer-Verlag, New York, 2001.
[64] K.M. Sayre. *Recognition, a study in the philosophy of artificial intelligence*. University of Notre Dame Press, 1965.
[65] M.I. Schlesinger and Hlavác. *Ten Lectures on Statistical and Structural Pattern Recognition*. Kluwer Academic Publishers, 2002.
[66] B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, 2002.
[67] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, UK, 2004.
[32] M. Stone. Cross-validation: A review. *Mathematics, Operations and Statistics*, (9):127–140, 1978.
[69] D.M.J. Tax. *One-class classification. Concept-learning in the absence of counter-examples*. PhD thesis, Delft University of Technology, The Netherlands, 2001.
[70] D.M.J. Tax and R.P.W. Duin. Support vector data description. *Machine Learning*, 54(1):45–56, 2004.
[71] F. van der Heiden, R.P.W. Duin, D. de Ridder, and D.M.J. Tax. *Classification, Parameter Estimation, State Estimation: An Engineering Approach Using MatLab*. Wiley, New York, 2004.
[72] V. Vapnik. *Estimation of Dependences based on Empirical Data*. Springer Verlag, 1982.
[73] V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.
[74] L.-X. Wang and J.M. Mendel. Generating fuzzy rules by learning from examples. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(6):1414–1427, 1992.
[75] S. Watanabe. *Pattern Recognition, Human and Mechanical*. John Wiley & Sons, 1985.
[76] A. Webb. *Statistical Pattern Recognition*. John Wiley & Sons, Ltd., 2002.
[77] S.M. Weiss and C.A. Kulikowski. *Computer Systems That Learn*. Morgan Kaufmann, 1991.
[78] R.C. Wilson and E.R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):634–648, 1997.
[79] R.C. Wilson, B. Luo, and E.R. Hancock. Pattern vectors from algebraic graph theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1112–1124, 2005.
[80] S. Wolfram. *A new kind of science*. Wolfram Media, 2002.
[81] D.H. Wolpert. *The Mathematics of Generalization*. Addison-Wesley, 1995.

[82] R.R. Yager, M. Fedrizzi, and J. (Eds) Kacprzyk. *Advances in the Dempster-Shafer Theory of Evidence*. Wesley, 1994.

[83] C.H. Yu. Quantitative methodology in the perspectives of abduction, deduction, and induction. In *Annual Meeting of American Educational Research Association*, San Francisco, CA, 2006.