Non-Euclidean dissimilarities: causes, embedding and informativeness

Robert P.W. Duin, Elżbieta Pękalska, and Marco Loog

Abstract In many pattern recognition applications object structure is essential for the discrimination purpose. In such cases researchers often use recognition schemes based on template matching which lead to the design of non-Euclidean dissimilarity measures. A vector space derived from the embedding of the dissimilarities is desirable in order to use general classifiers. An isometric embedding of the symmetric non-Euclidean dissimilarities results in a pseudo-Euclidean space. More and better tools are available for the Euclidean spaces but they are not fully consistent with the given dissimilarities.

In this chapter first a review is given of the various embedding procedures for the pairwise dissimilarity data. Next the causes are analyzed for the existence of non-Euclidean dissimilarity measures. Various ways are discussed in which the measures are converted into Euclidean ones. The purpose is to investigate whether the original non-Euclidean measures are informative or not. A positive conclusion is derived as examples can be constructed and found in real data for which the non-Euclidean characteristics of the data are essential for building good classifiers.[1]

Robert P.W. Duin
Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft University of Technology, The Netherlands, e-mail: r.duin@ieee.org

Elżbieta Pękalska
School of Computer Science, University of Manchester, United Kingdom,
e-mail: pekalska@cs.man.ac.uk

Marco Loog
Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft University of Technology, The Netherlands, e-mail: m.loog@tudelft.nl

[1] This chapter is based on previous publications by the authors, [16, 17, 19, 21, 23, 43] and contains text, figures, equations and experimental results taken from these papers.

1 Introduction

Automatic recognition systems work with objects such as images, videos, time signals, spectra and so on. They are built in the process of learning from a set of object examples labeled with the desired pattern classes. Two main steps can be distinguished in this procedure:

Representation: Individual objects are characterized by a set of suitable mathematical descriptors such as vectors, strings of symbols or graphs. A good representation is the one in which objects can easily be related to each other in order to facilitate the next step.

Generalization/Discrimination: The representations of the object examples should enable the mathematical modelling of object classes or class discriminants such that a good class estimate can be found for new, unseen and, thereby, unlabeled objects using the same representation.

The most popular representations, next to strings and graphs, encodes objects as vectors in Euclidean vector spaces. Instead of single vectors, also sets of vectors may be considered for representing individual objects, as studied e.g. in [48, 34, 35, 50]. For some applications, representations defined by strings of symbols and attributed graphs are preferred over vectors as they model the objects more accurately and offer more possibilities to include domain expert knowledge [7].

On the other hand, representations in Euclidean vector spaces are well suited for generalization. Many tools are available to build (learn) models and discriminant functions from sets of object examples (also called training sets) that may be used to classify new objects into the right class. Traditionally, the Euclidean vector space is defined by a set of features. These should ideally characterize the patterns well and be relevant for class differences at the same time. Such features have to be defined by experts exploiting their knowledge of the application.

The use of features has one important drawback. Features often represent the objects just partially because they encode their limited characteristics. Consequently, different objects may have the same representation, i.e. the same feature vector, when they differ by properties that are not expressed in the chosen feature set. This results in class overlap: in some areas of the feature space objects of different classes are represented by the same feature vectors. Consequently, they cannot be distinguished any longer, which leads to an intrinsic classification error, usually called the Bayes error.

An alternative to the feature representation is the dissimilarity representation defined on direct pairwise object comparisons. If the entire objects are taken into account in the comparison, then only identical objects will have a dissimilarity zero (if the dissimilarity measure has the property of 'identity of indiscernibles'). For such a representation class overlap does not exist if the objects are unambiguously labeled, which means that there are no real world objects in the application that belong to multiple classes.

Some dissimilarity measures used in practice do not have the property that a zero dissimilarity can only arise for identical objects. An example is the single-

linkage distance used in clustering: the dissimilarity between two clusters is defined as the distance between the two most neighboring vectors. This distance measure corresponds to defining the smallest distance between the surfaces of two real world objects as the distance between the objects. A zero value however does not imply that the objects are identical; they are just touching.

Distance measures such as the above, and many others, cannot be perfectly embedded in a Euclidean space. This means that there is no set of vectors in a vector space of any dimensionality for which the Euclidean distances between the objects are identical to the given ones. In particular, it holds for non-metric distances, which are just an example from a large set of non-Euclidean distance measures. As we want to include non-metric distances (such as the single-linkage distance) we will use the more general term of dissimilarities instead of distances. They refer to possibly improper distance measures in the mathematical sense. We will still assume that dissimilarities are non-negative and that they have a monotonic relation with object differences: if two given objects are made more different, their dissimilarity increases.

Non-Euclidean symmetric dissimilarity data can be perfectly embedded into pseudo-Euclidean spaces. A proper embedding of non-Euclidean dissimilarities and the training of classifiers in the resulting space are, however, not straightforward. There are computational as well as fundamental problems to be solved. The question thereby arises whether the use of non-Euclidean dissimilarity measures is strictly necessary. Finding the causes of such measures, see Section 2, is a first step to answer this question. This will be more extensively discussed in Section 6. We will investigate whether such measures are really informative and whether it is possible to make Euclidean corrections or approximations by which no information is lost.

Two main vectorial representations of the dissimilarity data, the dissimilarity space and the pseudo-Euclidean embedded space, are presented in Section 3. Section 4 discusses classifiers which can be trained in such spaces. Transformations which make the dissimilarity data Euclidean are briefly presented in Section 5. Next, numerous examples of artificial and real dissimilarity data are collected in Section 7. Oftentimes, they illustrate that linear classifiers in the dissimilarity-derived vector spaces are much more advantageous than the traditional 1-NN rule. Finally, we summarize and discuss our findings in Section 8.

The issue of informativeness of the non-Euclidean measures is the main topic of this chapter. We will present artificial and real world examples for which the use of such measures is really informative. We will, however, also make clear that for any given classifier defined in a non-Euclidean space an equivalent classifier in a Euclidean space can be constructed. It is a challenge to do this such that the training of good classifiers in this Euclidean space is feasible. In addition, we will argue that the dissimilarity space as proposed by the authors [39, 41] is a Euclidean space that preserves all non-Euclidean information and enables the design of well performing classifiers.

2 Causes of non-Euclidean dissimilarities

In this section we shortly explain why non-Euclidean dissimilarities frequently arise in the applications. This results from the analysis of a set of real world objects. Let D be an $N \times N$ dissimilarity matrix describing a set of pairwise dissimilarities between N objects. D is Euclidean if it can be perfectly embedded into a Euclidean space. This means that there exists a Euclidean vector space with N vectors for which all Euclidean distances are identical to the given ones.

There are N^2 free parameters if we want to position N vectors in an N-dimensional space. The dissimilarity matrix D has also N^2 values. D should be symmetric because the Euclidean distance is. Still, there might be no solution possible as the relation between vector coordinates and Euclidean distances is nonlinear. More on the embedding procedures is discussed in Section 3. At this moment we need to remember that the matrix D is Euclidean only if the corresponding vector space exists.

First, it should be emphasized how common non-Euclidean measures are. An extensive overview of such measures is given in [41], but we have often encountered that this fact is not fully recognized. Most researchers wrongly assume that non-Euclidean distances are equivalent to non-metric ones. There are however many metric but non-Euclidean distances, such as the city-block or $\ell - 1$-norm distance.

Almost all probabilistic distance measures are non-Euclidean by nature. This implies that by dealing with object invariants, the dissimilarity matrix derived from the overlap between the probability density functions corresponding to the given objects is non-Euclidean. Also the Mahalanobis class distance as well as the related Fisher criterion are non-Euclidean. Consequently, many non-Euclidean distance measures are used in cluster analysis and in the analysis of spectra in chemometrics and hyperspectral image analysis as spectra can be considered as one-dimensional distributions.

Secondly, what is often overlooked is the following fact. One may compare pairs of real world objects by a (weighted) Euclidean distance, yet the complete set of N objects giving rise to an $N \times N$ dissimilarity matrix D is non-Euclidean. In short, this is caused by the fact that different parts or characteristics of objects are used per pair to define the object differences. Even if the dissimilarity is defined by the weighted sum of differences, as long as there is no single basis of reference for the comparison of **all pairs**, the resulting dissimilarity matrix D will be non-Euclidean. These types of measures often result from matching procedures which minimize the cost or path of transformation between two objects. Fundamental aspects of this important issue are extensively discussed in section 2.2.3.

In shape recognition, various dissimilarity measures are based on the weighted edit distance, on variants of the Hausdorff distance or on non-linear morphing. Usual parameters are optimized within an application w.r.t. the performance based on template matching and other nearest neighbor classifiers [8]. Almost all have non-Euclidean behavior and some are even non-metric [15].

In the design and optimization of the dissimilarity measures for template matching, their Euclidean behavior is not an issue. With the popularity of support vector

machines (SVMs), it has become important to design kernels (similarities) which fulfill the Mercer conditions [13]. This is equivalent to a possibility of an isometric Euclidean embedding of such a kernel (or dissimilarities). Next subsections discuss reasons that give rise to violations of these conditions leading to non-Euclidean dissimilarities or indefinite kernels.

2.1 Non-intrinsic Non-Euclidean Dissimilarities

Below we identify some non-intrinsic causes that give rise to non-Euclidean dissimilarities. In such cases, it is not the dissimilarity measure itself, but the way it is computed or applied that causes the non-Euclidean behavior.

2.1.1 Numeric inaccuracies

Non-Euclidean dissimilarities arise due to the numeric inaccuracies caused by the use of a finite word length. If the intrinsic dimensionality of the data is lower than the sample size, the embedding procedure that relies on an eigendecomposition of a certain matrix, see Section 3, may lead to numerous tiny negative eigenvalues. They should be zero in fact, but become non-zero due to numerical problems. It is thereby advisable to neglect dimensions (features) that correspond to very small positive and negative eigenvalues.

2.1.2 Overestimation of large distances

Complicated measures are used when dissimilarities are derived from raw data such as (objects in) images. They may define the distance between two objects as the length of the path that transforms one object into the other. Examples are the weighted edit distance [4] and deformable templates [33]. In the optimization procedure that minimizes the path length, the procedure may approximate the transformation costs from above. As a consequence, too large distances are found. Even if the objectes are compared by a (weighted) Euclidean distance measure, the resulting set of dissimilarities in D will often become non-Euclidean or even non-metric.

2.1.3 Underestimation of small distances

The underestimation of small distances has the same result as the overestimation of large distances. It may happen when the pairwise comparison of objects is based on different properties for each pair, as it is the case e.g. in studies on consumer preference data. Another example is the comparison of partially occluded objects in computer vision.

2.2 Intrinsic Non-Euclidean Dissimilarities

The causes discussed in the above may be judged as accidental. They result either from computational or observational problems. If better computers and observations were available, they would disappear. Now, we will focuss on dissimilarity measures for which this will not happen. There are three possibilities.

2.2.1 Non-Euclidean Dissimilarities

As already indicated at the start of this section, arguments can be given from the application side to use another metric than the Euclidean one. An example is the l_1-distance between energy spectra as it is related to energy differences. Although the l_2-norm is very convenient for computational reasons and it is rotation invariant in a Euclidean space, other distance measures may naturally arise from the demands in applications, e.g. see [49].

2.2.2 Invariants

A fundamental reason behind non-Euclidean dissimilariities is related to the occurrence of invariants. Frequently, one is not interested in the dissimilarity between given objects A and B, but in the dissimilarity between their equivalence classes i.e. sets of objects $A(\theta)$ and $B(\theta)$ in which θ controls an invariant. One may define the dissimilarity between the A and B as the minimum difference between the sets defined by all their invariants. See Fig. 2.2.2 for an illustration of this idea.

$$d^*(A,B) = \min_{\theta_A} \min_{\theta_B} (d(A(\theta_A), B(\theta_B))) \tag{1}$$

This measure is non-metric: the triangle inequality may be violated as for different pairs of objects different values of θ are found minimizing (1).

2.2.3 Sets of vectors

Complicated objects such as multi-region images may be represented by sets of vectors. Problems like this are investigated in the domain of Multi Instance Learning (MIL), [14] or Bag-of-Words (BoW) classification [54]. Distance measures between such sets have already been studied for a long time in cluster analysis. Many are non-Euclidean or even non-metric, such as the single linkage distance. This measure is defined as the distance between the two most neighboring points of the two clusters being compared. It is non-metric. It even holds that if $d(A,B) = 0$, then it does not follow that $A \equiv B$.

For the single linkage dissimilarity measure it can be understood why the dissimilarity space may be useful. Given a set of such dissimilarities between clouds of

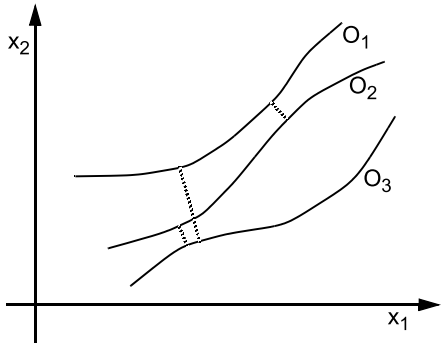

Fig. 1 Vector space with the invariant trajectories for three objects O_1, O_2 and O_3. If the chosen dissimilarity measure is defined as the minimum distance between these trajectories, triangle inequality can easily be violated, i.e. $d(O_1,O_2)+d(O_1,O_3)<d(O_1,O_3)$.

vectors, it can be concluded that two clouds are similar if the two sets of dissimilarities with all other clouds are about equal. If just their mutual dissimilarity is (close to) zero, they may still be very different.

The problem with the single linkage dissimilarity measure between two sets of vectors points to a more general problem in relating sets and even objects. In [35] an attempt has been made to define a proper Mercer kernel between two sets of vectors. Such sets are in that paper compared by the Hellinger distance derived from the Bhattacharyya's affinity between two pdfs $p_A(x)$ and $p_B(x)$ found for the two vector sets A and B:

$$d(A,B)=\left[\int\left(\sqrt{p_A(x)}-\sqrt{p_B(x)}\right)^2\right]^{1/2}. \tag{2}$$

The authors state that by expressing $p(x)$ in any orthogonal basis of functions, the resulting kernel K is automatically positive semidefinite (psd). This is only correct however, if all vector sets $A,B,...$ to which the kernel is applied have the same basis. If different bases are derived in a pairwise comparison of sets, the kernel may become indefinite. This occurs if the two pdfs are estimated in a subspace defined by a PCA computed from the objects of the two classes A and B only.

This makes clear that indefinite relations may arise in any pairwise comparison of real world objects if every pair of objects if first represented in some joint space in which the dissimilarity is computed. These joint spaces may be different for different pairs! Consequently, the total set of dissimilarities will likely have a non-Euclidean behavior, even if each comparison relies on the Euclidean distance, as in (2).

The consequence of this observation is huge for pattern recognition applications. It implies that a representation defined by pairwise dissimilarities between objects can only be Euclidean if a common basis between all objects, including the future test objects, is found for the derivation of such dissimilarities. This is naturally, by definition, the case for feature vector representations, as the joint space for all

objects is already defined by the chosen set of features. For the dissimilarity representation, however, which has the advantage of potentially using the entire objects, the consequence is that no common representation basis can be found before all objects are seen. This contradicts the idea of generalization and discrimination: being able to classify unseen objects.

We emphasize this conclusion as we judge it as very significant: Non-Euclidean object relations naturally arise for real world object recognition as no Euclidean representation can be defined before we have seen (or implicitly considered) all objects, including the ones to be recognized in future. Transductive inference [51] is the solution: include the objects to be classified in the definition of the representation.

3 Vector spaces for the dissimilarity representation

The complete dissimilarity representation is defined as a square matrix with the dissimilarities between all pairs of objects. Traditionally, in the nearest neighbor classification scenario, just the dissimilarities between the test objects and training objects are used. For every test object the nearest neighbors in the set of training objects are first found and used by the nearest neighbor rule. This procedure does not make use of the pairwise relations between the training objects.

The following two approaches construct a new vector space on the basis of the relations within the training set. The resulting vector space is used for training classifiers.

In the first approach, the dissimilarity matrix is considered as a set of vectors, one for every object. They represent the objects in a vector space constructed by the dissimilarity vectors whose coordinates are dissimilarities to the training objects. Usually, this vector space is treated as a Euclidean space and equipped with the standard inner product definition.

In the second approach, an attempt is made to embed the dissimilarity matrix in a Euclidean vector space such that the distances between the extracted vectors are equal to the given dissimilarities. This can only be realized without error, of course, if the original set of dissimilarities is Euclidean. If this is not the case, either an approximate procedure has to be followed or the objects should be embedded into a non-Euclidean vector space. This is a space in which the standard inner product definition and the related distance measure are changed, leading to indefinite inner products and later to indefinite kernels.

It appears that an exact embedding is possible for every symmetric $N \times N$ dissimilarity matrix D with zero self-dissimilarity, i.e. a diagonal all of zeros. The resulting space is the so-called pseudo-Euclidean space.

These two approaches are more formally defined below, using an already published description [21].

3.1 Dissimilarity space

Let $\mathscr{X} = \{x_1, \ldots, x_n\}$ be a training set. Given a dissimilarity function and/or dissimilarity data, we define a data-dependent mapping $D(\cdot, R) : \mathscr{X} \to \mathbb{R}^k$ from $\mathscr{X}$ to the so-called *dissimilarity space* (DS) [20, 28, 45]. The k-element set R consists of objects that are representative for the problem. This set is called the representation set or prototype set and it may be a subset of $\mathscr{X}$. In the dissimilarity space each dimension $D(\cdot, p_i)$ describes a dissimilarity to a prototype p_i from R.

We initially choose $R := \mathscr{X}$. As a result, every object is described by an n-dimensional dissimilarity vector $D(x, \mathscr{X}) = [d(x,x_1) \ \ldots \ d(x,x_n)]^T$. The resulting vector space is endowed with the traditional inner product and the Euclidean metric.

Any dissimilarity measure ρ can be defined in the Dissimilarity Space. One of them is the Euclidean distance:

$$\rho_{DS}(x,y) = (\sum_{i=1}^{n} [d(x,x_i) - d(y,x_i)]^2)^{1/2} \tag{3}$$

This is the distance computed on dissimilarity vectors defined by original dissimilarities. For metric dissimilarity measures ρ it holds asymptotically that the nearest neighbor objects are unchanged by ρ_{DS}. This is however not necessarily true for finite data sets. In that case the nearest neighbors in dissimilarity space might be more appropriate for classifications as the distances are defined in the context of the entire representation set.

The approaches discussed here are originally intended for dissimilarities directly computed between objects and not resulting from feature representations. It is, however, still possible to study dissimilarity representations derived from features which may yield interesting results [42]. In Fig. 2 an example is presented that compares an optimized radial basis SVM with a Fisher linear discriminant computed in the dissimilarity space derived from the Euclidean distances in a feature space. The example shows a large variability of the nearest neighbor distances. As the radial basis kernel used by SVM is constant it cannot be optimal for all regions of the feature space.

The Fisher linear discriminant is computed in the complete dissimilarity space, where the classes are linearly separable. Although the classifier is overtrained (the dissimilarity space is 100-dimensional and the training set has also 100 objects) it gives here the perfect result. It should be realized that this example is specifically constructed to show the possibilities of the dissimilarity space.

3.2 Pseudo-Euclidean space

Before explaining the relation between pseudo-Euclidean spaces and dissimilarity representation, we start with definitions.

A Pseudo-Euclidean Space (PES) $\mathscr{E} = \mathbb{R}^{(\sqrt{\cdot}^{\blacksquare})} = \mathbb{R}^{\surd} \oplus \mathbb{R}^{\blacksquare}$ is a vector space with a non-degenerate indefinite inner product $\langle \cdot, \cdot \rangle_{\mathscr{E}}$ such that $\langle \cdot, \cdot \rangle_{\mathscr{E}}$ is positive definite on $\mathbb{R}^p$ and negative definite on $\mathbb{R}^q$ [25, 41]. The inner product in $\mathbb{R}^{(p,q)}$ is defined (wrt an orthonormal basis) as $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathscr{E}} = \mathbf{x}^T \mathscr{J}_{pq} \mathbf{y}$, where $\mathscr{J}_{pq} = [I_{p \times p}\ 0; 0\ -I_{q \times q}]$ and I is the identity matrix. As a result, $d^2_{\mathscr{E}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^T \mathscr{J}_{pq} (\mathbf{x} - \mathbf{y})$. Obviously, a Euclidean space $\mathbb{R}^p$ is a special case of a pseudo-Euclidean space $\mathbb{R}^{(p,0)}$. An infinite-dimensional extension of a PES is a Kreĭn space. It is a vector space $\mathscr{K}$ equipped with an indefinite inner product $\langle \cdot, \cdot \rangle_{\mathscr{K}} : \mathscr{K} \times \mathscr{K} \to \mathbb{R}$ such that $\mathscr{K}$ admits an orthogonal decomposition as a direct sum, $\mathscr{K} = \mathscr{K}_+ \oplus \mathscr{K}_-$, where $(\mathscr{K}_+, \langle \cdot, \cdot \rangle_+)$ and $(\mathscr{K}_-, -\langle \cdot, \cdot \rangle_-)$ are separable Hilbert spaces with their corresponding positive and negative definite inner products.

A positive definite kernel function can be interpreted as a generalized inner product in some Hilbert space. This space becomes Euclidean when a kernel matrix is considered. In analogy, an arbitrary symmetric kernel matrix can be interpreted as a generalized inner product in a pseudo-Euclidean space. Such a PES is obviously

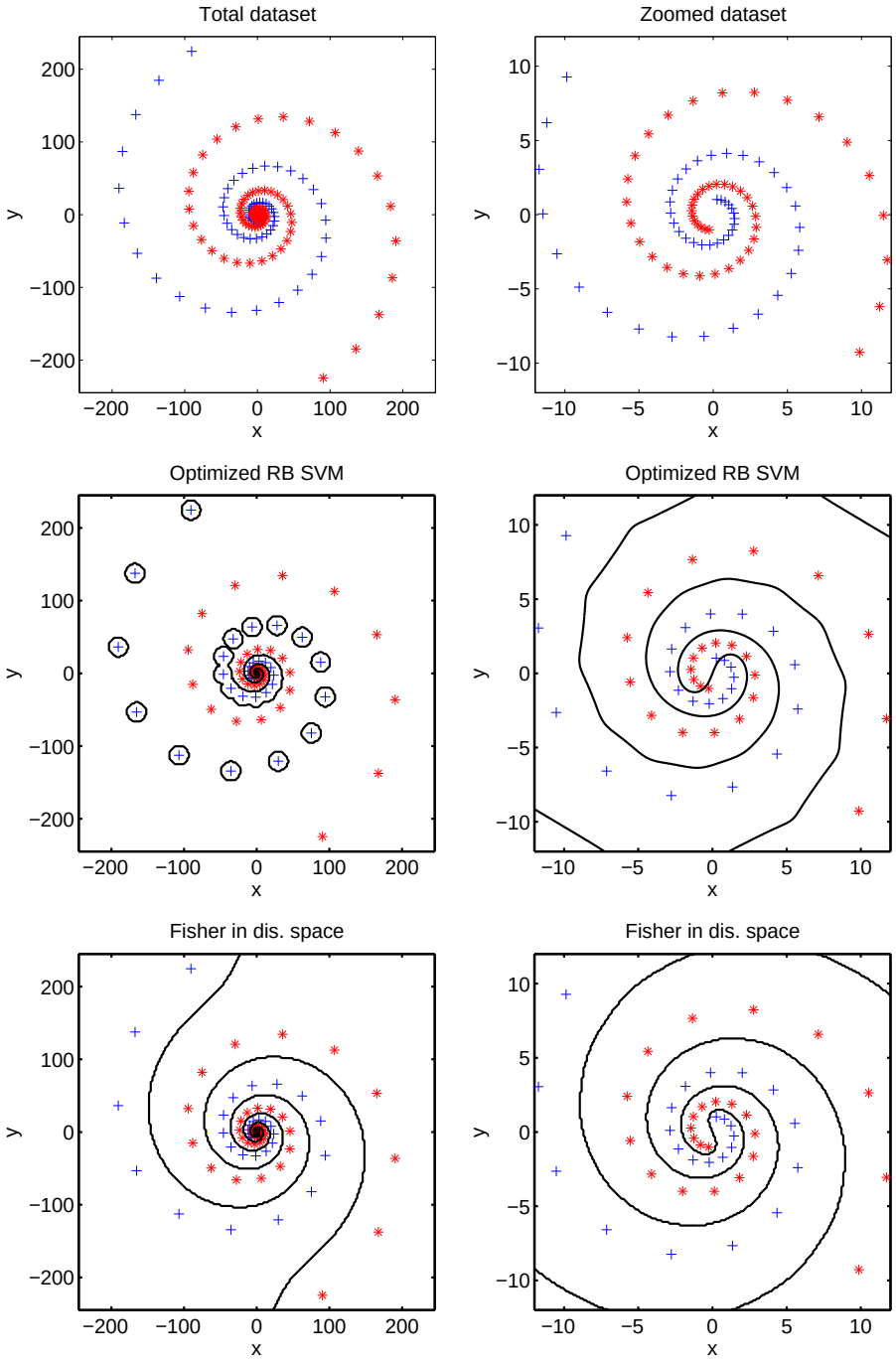

Fig. 2 A spiral example with 100 objects per class. Left column shows the complete data sets, while the right column presents the zoom of the spiral center. 50 objects per class, systematically sampled, are used for training. The middle row shows the training set and SVM with an optimized radial basis function; 17 out of 100 test objects are erroneously classified. The bottom row shows the Fisher Linear Discriminant (without regularization) computed in the dissimilarity space derived from the Euclidean distances. All test objects are correctly classified.

data dependent and can be retrieved via an embedding procedure. Similarly, an arbitrary symmetric dissimilarity matrix with zero self-dissimilarities can be interpreted as a pseudo-Euclidean distance in a proper pseudo-Euclidean space.

Since in practice we deal with finite data, dissimilarity matrices or kernel matrices can be seen as describing relations between vectors in the underlying pseudo-Euclidean spaces. These pseudo-Euclidean spaces can be either determined via an embedding procedure and directly used for generalization, or approached indirectly by the operations on the given indefinite kernel. The section below explains how to find the embedded PES.

3.2.1 Pseudo-Euclidean embedded space

A symmetric dissimilarity matrix $D := D(\mathscr{X}, \mathscr{X})$ can be embedded in a Pseudo-Euclidean Space (PES) $\mathscr{E}$ by an isometric mapping [25, 41]. The embedding relies on the indefinite Gram matrix G, derived as $G := -\frac{1}{2} H D^{\star 2} H$, where $D^{\star 2} = (d_{ij}^2)$ and $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$ is the centering matrix. H projects the data such that X has a zero mean vector. The eigendecomposition of G leads to $G = Q\Lambda Q^T = Q|\Lambda|^{\frac{1}{2}}[\mathscr{J}_{pq}; 0]|\Lambda|^{\frac{1}{2}}Q^T$, where Λ is a diagonal matrix of eigenvalues, first decreasing p positive ones, then increasing q negative ones, followed by zeros. Q is the matrix of eigenvectors. Since $G = X\,\mathscr{J}_{pq}X^T$ by definition of a Gram matrix, $X \in \mathbb{R}^n$ is found as $X = Q_n\,|\Lambda_n|^{\frac{1}{2}}$, where Q_n consists of n eigenvectors ranked according to their eigenvalues Λ_n. Note that X has a zero mean and is uncorrelated, because the estimated pseudo-Euclidean covariance matrix $C = \frac{1}{n-1}X^T X\,\mathscr{J}_{pq} = \frac{1}{n-1}\Lambda_r$ is diagonal. The eigenvalues λ_i encode variances of the extracted features in $\mathbb{R}^{(p,q)}$.

Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. If this space is a PES $\mathbb{R}^{(p,q)}$, $p+q=n$, the pseudo-Euclidean distance is computed as:

$$\begin{aligned}\rho_{PES}(\mathbf{x},\mathbf{y}) &= (\sum_{i=1}^{p}[x_i - y_i]^2 - \sum_{i=p+1}^{p+q}[x_i - y_i]^2)^{1/2} \\ &= (\sum_{i=1}^{n}\delta(i,p)[x_i - y_i]^2)^{1/2},\end{aligned}$$

where $\delta(i,p) = \mathrm{sign}(p - i + 0.5)$. Since the complete pseudo-Euclidean embedding is perfect, $D(x,y) = \rho_{PES}(x,y)$ holds.

Other distance measures may also be defined between vectors in a PES, depending on how this space is interpreted. Two obvious choices are:

$$\rho_{PES+}(\mathbf{x},\mathbf{y}) = (\sum_{i=1}^{p}[x_i - y_i]^2)^{1/2}, \tag{4}$$

which neglects the dimensions corresponding to the negative contributions (derived from negative eigenvalues in the embedding), and

$$\rho_{AES}(\mathbf{x},\mathbf{y}) = (\sum_{i=1}^{n}[x_i - y_i]^2)^{1/2}, \tag{5}$$

which treats the vector space $\mathbb{R}^n$ as Euclidean $\mathbb{R}^{p+q}$. This means that the negative subspace of PES is interpreted as a Euclidean subspace (i.e. the negative signs of eigenvalues are neglected in the embedding procedure).

To inspect the amount of non-Euclidean influence in the derived PES, we define Negative EigenFraction (NEF) as:

$$NEC = \sum_{j=p+1}^{p+q} |\lambda_j| / \sum_{i=1}^{p+q} |\lambda_i| \in [0,1] \tag{6}$$

Fig. 3 shows how NEF varies as a function of p of the Minkowski-p dissimilarity measure (k-dimensional spaces) for a two-dimensional example:

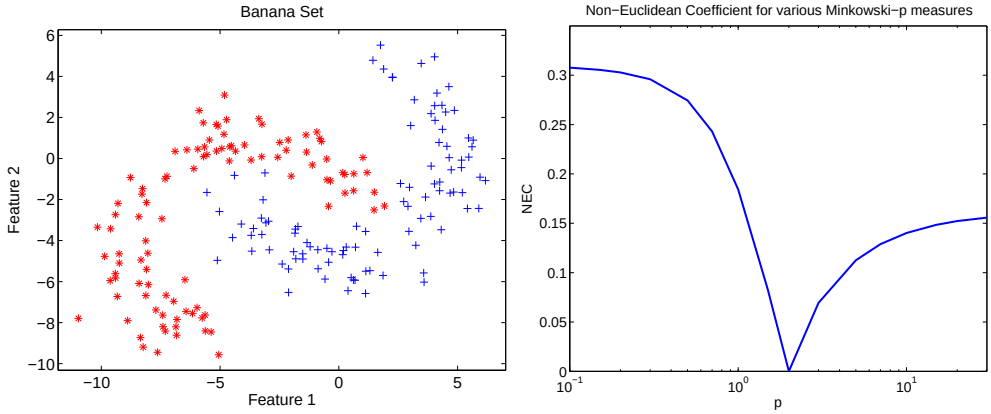

Fig. 3 A two-dimensional data set (left) and the NEF as a function of p for various Minkowski-p dissimilarity measures.

$$\rho_{Min_p}(\mathbf{x},\mathbf{y}) = (\sum_{i=1}^{k}[x_i - y_i]^p)^{1/p} \tag{7}$$

This dissimilarity measure is Euclidean for $p = 2$ and metric for $p > 1$. The measure is non-Euclidean for all $p \neq 2$. The value of NEF may vary considerably with a changing dimensionality. This phenomenon is illustrated in Fig. 4 for 100 points generated by a standard Gaussian distribution for various values of p. The one-dimensional dissimilarities obviously fit perfectly to a Euclidean space. For a high dimensionality, the sets of dissimilarities become again better embeddable in a Euclidean space.

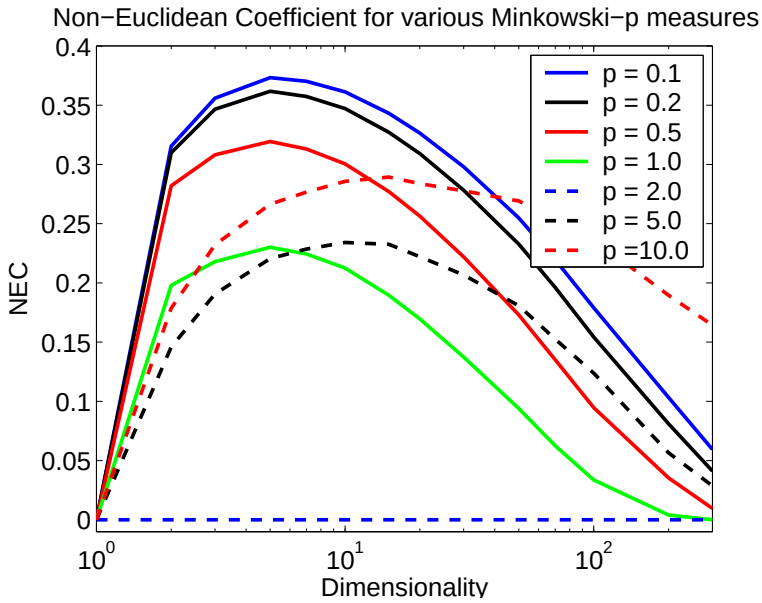

Fig. 4 The Non-Euclidean Coefficient for various Minkowski-p dissimilarity measures as a function of the dimensionality of a set of 100 points generated by a standard Gaussian distribution.

3.3 Discussion on dissimilarity-based vector spaces

Now we want to make some remarks on the two procedures for deriving vector spaces from dissimilarity matrices, as discussed in previous subsection.

The dissimilarity space interprets the dissimilarity vectors, defined by the dissimilarities from objects to particular prototypes from the representation set, as features. The true characteristics behind the used dissimilarity measure is not used when a general classifier is applied in a dissimilarity space. Special classifiers are needed to make use of that information. The good side of this 'disadvantage' is that the dissimilarity space can be used for any dissimilarity representation, including ones that are negative, asymmetric or weird, otherwise.

The embedding procedure is more restrictive. The dissimilarities are assumed to be symmetric and become zero for identical objects. A pseudo-Euclidean space is needed for a perfect embedding in case of non-Euclidean data sets. A pseudo-Euclidean space is however "broader" than the original distance measure in the sense that it allows negative square distances. Moreover, the requirements of a proper metric or well-defined distances obeying the triangle inequality are not of use as they do not guarantee a Euclidean embedding.

A severe drawback of both procedures is that they initially generate vector spaces that have as many objects as dimensions. Specific classifiers or dimension reduction procedures are thereby needed. For the dissimilarity representation this is more

feasible than for the feature representation: features can vary greatly in their discriminative power, range, costs or characteristics. Some features may be very good, others might be useless, or only useful in relation with particular other features.

This is not true for dissimilarities. The initial representation is based on objects which have similar characteristics. It is not beneficial to use two objects that are much alike as it leads to highly correlated dissimilarity vectors. Systematic, or even random procedures that reduce the initial representation set (in fact prototype selection) can be very effective [40] for this reason.

A relevant topic in the comparison of both procedures is the representation of new objects in a given space derived from dissimilarities between an earlier set of objects ("projection"). For the dissimilarity space this is simple. It is defined by the dissimilarities with the representation set used to define the space. A "projection" into a pseudo-Euclidean space is not straight forward. The space itself is found by the eigenvalue decomposition. Traditionally new objects are projected into such a space by determining the point with the shortest distance. For pseudo-Euclidean spaces however this is not appropriate as distances can be negative. The projection point can thereby be chosen such that it has an arbitrarily large negative distance. The consequence is that in case new objects are considered the space has to rebuild from the combined set of old and new objects. This is directly related to the final observation made in Section 2.2.3 about the need to use transductive inference for non-Euclidean data.

4 Classifiers

We will discuss here a few well-known classifiers and their behavior in various spaces. This is a summary of our experiences based on numerous studies and applications. See [41, 19, 22] and their references.

In order to make a choice between the embedded pseudo-Euclidean space and the dissimilarity space for classifier training one should take into account the essential differences between these spaces. Pseudo-Euclidean embedding aims to preserve the given distances, while the dissimilarity space is not concerned about it. In addition, there is a nonlinear transformation between these spaces: the dissimilarity space can be defined by computing the distances to the prototypes in the embedded space. As a consequence, a linear classifier in the embedded space is a nonlinear classifier in the dissimilarity space. The reverse holds as well, but it should be kept in mind that the dissimilarity space is more general. As it is also defined for arbitrary, even asymmetric, dissimilarities, classifiers will relate to possible objects that do not exist in the embedded space.

It is outside the scope of this chapter, but the following observation might be helpful for some readers. If the dissimilarities are not constructed by a procedure on a structural representation of objects, but are derived as Euclidean distances in a feature space, then the pseudo-Euclidean embedding effectively reconstructs the original Euclidean feature space (except for orthonormal transformations). So in that case a linear classifier in the dissimilarity space is a nonlinear classifier in the embedded space, which is the same nonlinear classifier in the feature space. Such a classifier, computed in a dissimilarity space, can perform very well [18, 22].

4.1 Nearest neighbor classifier

The k-nearest neighbor (k-NN) classifier in an embedded (pseudo-)Euclidean space is based on the distances computed in this space. By definition these are the original dissimilarities (provided that the test examples are embedded together with the training objects). So without the process of embedding, this classifier can directly be applied to a given dissimilarity matrix but is simultaneously also a classifier for the embedded space. This is the classifier traditionally used by many researchers in the area of structural pattern recognition. The study of the dissimilarity representation arose because this classifier did not make use of the dissimilarities between the objects in the training set. Classification is entirely based on the dissimilarities of a test object to the objects in the training (or representation) set only.

The k-NN rule computed in the dissimilarity space relies on a Euclidean distance between the dissimilarity vectors, hence the nearest neighbors are determined by using all dissimilarities of a given object to the representation objects. As explained in Section 3.1, it is already mentioned that the distances between similar objects are small in the two spaces for large training sets and the metric distance. So, it is expected that learning curves are asymptotically identical. However, for small

training sets the k-NN classifier in the dissimilarity space performs usually better than the direct k-NN rule as it uses more information.

4.2 Parzen density classifiers

The class densities computed by the Parzen kernel density procedure are based on pairwise distance computations between objects. The applicability of this classifier as well as its performance is thereby related to those of the k-NN rule. The major difference is that this classifier is smoother, depending on the choice of the smoothing parameter (kernel) and that its optimization involves the entire training set.

4.3 Normal density Bayes classifiers

Bayes classifiers assume that classes can be described by probability density functions. The expected classification error is minimized by using class priors and the Bayes' rule. In case of normal density functions either a linear classifier (Linear Discriminant Analysis, LDA) arises on the basis of equal class covariances, or a quadratic classifier is obtained for the general case (Quadratic Discriminant Analysis, QDA). These two classifiers are the best possible in case of (nearly) normal class distributions and a sufficiently large training set. As mean vectors and covariance matrices can be computed in a pseudo-Euclidean space, see [26, 41], these classifiers can be re-defined there as well if we forget the starting point of normal distributions. The reason is that normal distributions are not well defined in pseudo-Euclidean spaces; it is not clear what a normal distribution is unless we refer to associated Euclidean spaces.

In a dissimilarity space the assumption of normal distributions works often very well. This is due to the fact that in many cases dissimilarity measures are based on, or related to sums of numerical differences. Under certain conditions large sums of random variables tend to be normally distributed. It is not perfectly true for distances as we often get Weibull [9] or χ^2 distributions, but the approximations are sufficient for a good performance of LDA and QDA. The effect is emphasized if the classification procedure involves the computation of linear subspaces, e.g. by PCA. Thanks to projections the aspect of normality is emphasized even more.

4.4 Fisher's linear discriminant

In a Euclidean space the Fisher linear discriminant (FLD) is defined as the linear classifier that maximizes the Fisher criterion, i.e. the ratio of the between-class variance to the within-class variance. For a two-class problem, the solution is equivalent

to LDA (up to an added constant), even though no assumption is made about normal distributions. Since variance and covariance matrices are well defined in pseudo-Euclidean spaces, the Fisher criterion can be used to derive the FLD classifier there. Interestingly, FLD in a pseudo-Euclidean space coincides with FLD in the associated Euclidean space. FLD is a linear classifier in a pseudo-Euclidean space, but can be rewritten to FLD in the associated space; see also [44, 31].

In a dissimilarity space, which is Euclidean by definition, FLD coincides with LDA for a two-class problem. The performances of these classifiers may differ for multi-class problems as the implementations of FLD and LDA will usually vary then. Nevertheless, FLD performs very well. Due to the nonlinearity of the dissimilarity measure, FLD in a dissimilarity space corresponds to a nonlinear classifier in the embedded pseudo-Euclidean space.

4.5 Logistic classifier

The logistic classifier is based on a model of the class posterior probabilities as a function of the distance to the classifier [1]. The distance between a vector and a linear hyperplane in a pseudo-Euclidean space however is an unsuitable concept for classification as it can have any value $(-\infty, \infty)$ for vectors on the same side of this hyperplane. We are not aware of a definition and an implementation of the logistic classifier for pseudo-Euclidean spaces. Alternatively, the logistic classifier can be constructed in the associated Euclidean space.

In a dissimilarity space, the logistic classifier performs well, although in practice normal density based classifiers work often better. It relaxes the demands for normality as made by LDA. It is also more robust in case of high-dimensional spaces.

4.6 Support vector machine (SVM)

The linear kernel in a pseudo-Euclidean space is indefinite (non-Mercer). The quadratic optimization procedure used to optimize a linear SVM may thereby fail [30]. An SVM can however be constructed if the contribution of the positive subspace of the Euclidean space is much stronger than that of the negative subspace. Mathematically, it means that the measure is only slightly deviating from the Euclidean behavior and the solution of the SVM optimization is found in the positive definite neighborhood. Various researchers have reported good results in applying this classifier, e.g. see [6]. Although the solution is not guaranteed and the algorithm (in this case LIBSVM, [11]) does not stop at the global optimum, a good classifier can be constructed.

In case of a dissimilarity space the (linear) SVM is particularly useful for computing classifiers in the complete space in which the representations set equals thr trainsing set, $R := \mathscr{X}$, see Section 3.1. The given training set $\mathscr{X}$ defines therefore a

separable problem. The SVM classifier is well defined. It does not overtrain or only overtrains just slightly. The advantage of this procedure is that it does not demand a reduction of the representation set. By a suitable normalization of the dissimilarity matrix (such that the average dissimilarity is one) we found stable and good results in many applications by setting the trade-off parameter C in the SVM procedure [12] to $C = 100$. Hereby, additional cross-validation loops are avoided to optimize this parameter. As a result, in an application one can choose to focus on optimizing the dissimilarity measure.

5 Transformations

We will summarize the problem of building vector spaces from non-Euclidean dissimilarities as discussed so far:

- Non-Euclidean dissimilarities naturally arise in comparing real world objects for recognition purposes, see Section 2.2 and in particular Section 2.2.3.
- The pseudo-Euclidean space, Section 3.2 offers a proper isometric embedding for non-Euclidean data while the dissimilarity space, Section 3.1 postulates an Euclidean space in which just under some conditions, asymptotically, the nearest neighbor relations may be consistent with the given ones.
- The definition of classifiers in the pseudo-Euclidean space is not straightforward and many of the standard tools developed for statistical pattern recognition and machine learning are not valid or need to be redesigned. The dissimilarity space however is a standard vector representation that can be used as the traditional feature space. See Section 4.
- The representation of new objects for classification purposes is for the pseudo-Euclidean space not well defined and for the dissimilarity space straightforward, see Section 3.3. The only proper existing solution for the pseudo-Euclidean space is to include these objects in construction of the space, at the cost of retraining the classifiers. This type of transductive learning [51], is fundamentally related to non-Euclidean object dissimilarities, see Section 2.2.3. For the dissimilarity spaces transduction can be easily realized (at the cost of retraining the classifiers) or skipped (at the cost of accuracy).

Given the above, the dissimilarity space is preferred in most applications. It is easy to define and to handle. There is a one-to-one relation with the constituting dissimilarity matrix between the given objects. Any change in this matrix is reflected in a change of the representation. Moreover, this change is continuous. It can thereby be stated that there is no loss of information. The pseudo-Euclidean embedding on the other hand is of fundamental interest as it directly reflects the non-Euclidean aspects of the data. It is thereby a perfect place to study the question whether the non-Euclideaness contributes to the recognition performance or disturbs it.

One way to do this is to investigate transformations of the pseudo-Euclidean space that shrink or remove the non-Euclideaness. We discuss shortly a number of possibilities. See [23],[21] for more information.

5.1 The Dissimilarity Space (DS)

The original pseudo-Euclidean space, based on all eigenvectors, offers an isometric embedding of the given dissimilarities. So if we compute the distances in this space between all object, the original dissimilarity matrix is obtained and thereby the dissimilarity space. If the pseudo-Euclidean space is first transformed, e.g. by rescaling or by deleting some axes (eigenvectors of the original embedding) then in

a similar way a dissimilarity space can be obtained. This Euclidean space reflects all information of such a transformed pseudo-Euclidean space. As the transformation is continuous, then for any classifier in the pseudo-Euclidean space there exists a classifier in the dissimilarity space that yields the same classification. The transformation, however, is non-linear. So a linear classifier in the pseudo-Euclidean space is non-linear in the dissimilarity space and the other way around. Consequently, classifiers classifiers trained in these spaces before and after transformation yield different performances.

5.2 The Positive part of the Pseudo Euclidean Space (PES+)

The most obvious correction for a pseudo-Euclidean space $\mathbb{R}^{(p,q)}$ is to neglect the negative definite subspace. This results in a p-dimensional Euclidean space $\mathbb{R}^p$ with many-to-one mappings. Consequently, it is possible that the class overlap for the training set increases. It may, however, be worthwhile if the negative eigenvalues in the embedding procedure are mainly the result of noise and are not informative for the class separation. In that case this correction may improve the classification.

5.3 The Negative part of the Pseudo Euclidean Space (PES-)

In case the positive definite subspace of the pseudo-Euclidean space $\mathbb{R}^{(p,q)}$ is neglected a q-dimensional Euclidean space $\mathbb{R}^q$ is obtained. It is expected for real world applications that this space will show a bad class separation. As in this space however all information is collected that makes the dissimilarities non-Euclidean, any separation will indicate that such useful information exists.

5.4 The Associated Euclidean Space (AES)

Since $\mathbb{R}^{(p,q)}$ is a vector space, we can equip it with the traditional inner product, which leads to the so-called associated Euclidean space $\mathbb{R}^{p+q}$. It means that the vector coordinates are identical to those of PES, but now we use the norm and distance measure that are Euclidean. This is consistent with the natural topology of a vector space. This solution is identical to the one obtained by classical scaling based on the magnitudes of eigenvalues [27, 41].

5.5 Dissimilarity Enlargement by a Constant (DEC)

Instead of modifying the embedding procedure, the dissimilarity matrix may be adapted such that it is embeddable into a Euclidean space. A simple way to avoid the negative eigenvalues is to increase all off-diagonal elements of the dissimilarity matrix such that $d_c^2(x_i,x_j) = d^2(x_i,x_j) + 2c, \forall_{i \neq j}$. The value of c is chosen such that $c \geq -\lambda_{min}$, where λ_{min} is the smallest negative eigenvalue in the pseudo-Euclidean embedding of D. As a result, all eigenvalues are increased by c [41].

In our experiments we set $c = -\lambda_{min}$. Since the eigenvalues reflect the variances of the embedded data, the dimensions of the resulting Euclidean space are unevenly scaled by $\sqrt{\lambda_i + c}$. Note that the dimension with the largest negative contribution in PES has now a zero variance. In this way, dimensions related to noisy negative eigenvalues are more pronounced [41].

6 Are non-Euclidean dissimilarity measures informative?

The question about informativeness of non-Euclidean dissimilarity measures is different than the question whether non-Euclidean measures are better than Euclidean ones. The later question cannot be answered in general. After studying a set of individual problems compared for a large set of dissimilarity measures it might be found that for some problems the best measure is non-Euclidean. Such a result however is always temporary. A new Euclidean measure that outperforms the earlier ones may be invented later.

The question of informativeness on the other hand may be answered in an absolute sense. Even if a particular measure is not the best one, its non-Euclidean contribution can be judged as informative if the performance deteriorates by removing it. Should this result also be found by a classifier constructed in the non-Euclidean space? If a Euclidean correction can be found for an initially non-Euclidean representation that enables the construction of a good classifier, is the non-Euclidean dissimilarity measure then informative? We answer this question positively as any transformation can be included in the classifier and thereby effectively a classifier for the non-Euclidean representation has been found.

We will therefore state that the non-Euclidean character of a dissimilarity measure is non-informative if the classification result improves by removing its non-Euclidean contribution. The answer may be classifier dependent.

The traditional way of removing the non-Euclidean contribution is by neglecting the negative eigenvectors that define dimensions of the pseudo-Euclidean embedding. This is the PES+ defined in Section 5. The PES-, can be used as a check to see whether there is any class separability in the negative part of the embedded space. The below experiments are entirely based on the dissimilarity spaces of the various spaces, see Section 5.

We analyze a set of public domain dissimilarity matrices used in various applications, as well as a few artificially generated ones. See Table 1 for some properties: *size* (number of objects), (number of) *classes*, *non-metric* (fraction of triangle violations, if zero the dataset is metric), *NEF* (negative eigenfraction, see Section 3.2.1) and *Rand Err* (classification error by random assignment). Every dissimilarity matrix is made symmetric by averaging with its transpose and normalized by the average off-diagonal dissimilarity. We compute the linear SVM in the dissimilarity spaces based on the original pseudo-Euclidean space (PES), the positive space (PES+) and the negative space (PES-). Error estimates are based on the leave-one-out crossvalidation. These experiments are done in a transductive way: test objects are included in the derivation of the embedded space as well as the dissimilarity representations.

The four Chickenpieces datasets are the averages of 11 dissimilarity matrices derived from a weighted edit distance between blobs [4]. FlowCyto is the average of four specific histogram dissimilarities including an automatic calibration correction. WoodyPlants is a subset of the shape dissimilarities between leaves of woody plants [32]. We used classes with more than 50 objects. Catcortex is based on the connection strength between 65 cortical areas of a cat, [28]. Protein measures pro-

Table 1 Classification errors of the linear SVM for several representations using the leave-one-out crossvalidation.

	size	classes	Non-Metric	NEF	Rand Err	PES -¿ DS	PES+ -¿ DS	PES- -¿ DS
Chickenpieces45	446	5	0	0.156	0.791	**0.022**	0.132	0.175
Chickenpieces60	446	5	0	0.162	0.791	**0.020**	0.067	0.173
Chickenpieces90	446	5	0	0.152	0.791	**0.022**	0.052	0.148
Chickenpieces120	446	5	0	0.130	0.791	**0.034**	0.108	0.148
WoodyPlants50	791	14	5e-4	0.229	0.928	**0.075**	0.076	0.442
CatCortex	65	4	2e-3	0.208	0.738	**0.046**	0.077	0.662
Protein	213	4	0	0.001	0.718	0.005	**0.000**	0.634
Balls3D	200	2	3e-4	0.001	0.500	**0.470**	0.495	0.000
GaussM1	500	2	0	0.262	0.500	**0.202**	**0.202**	0.228
GaussM02	500	2	5e-4	0.393	0.500	0.204	**0.174**	0.252
CoilYork	288	4	8e-8	0.258	0.750	**0.267**	0.313	0.618
CoilDelftSame	288	4	0	0.027	0.750	**0.413**	0.417	0.597
CoilDelftDiff	288	4	8e-8	0.128	0.750	**0.347**	0.358	0.691
NewsGroups	600	4	4e-5	0.202	0.733	**0.198**	0.213	0.435
BrainMRI	124	2	5e-5	0.112	0.499	0.226	**0.218**	0.556
Pedestrians	689	3	4e-8	0.111	0.348	**0.010**	0.015	0.030

tein sequence differences using an evolutionary distance measure [29]. Balls3D is an artificial dataset based on the surface distances of randomly positioned balls of two classes having a slightly different radius. GaussM1 and GaussM02 are based on two 20-dimensional normally distributed sets of objects for which dissimilarities are computed using the ℓ_p-norm (Minkovsky) distances with $p = 1$ (metric, non-Euclidean) and $p = 0.2$ (non-metric). The three Coil datasets are based on the same sets of SIFT points in the COIL images compared by different graph distances. BrainMRI is the average of 182 dissimilarity measures obtained from MRI brain images. Pedestrians is a set of dissimilarities between detected objects (possibly pedestrians) in street images of the classes 'pedestrian', 'car' and 'other'. They are based on cloud distances between sets of feature points derived from single images.

The table shows examples of non-Euclidean datasets for which the non-Euclideaness is informative, as well datasets for which it is non-informative. In all cases where the error of the PES- is significantly better than the error of random assignment the negative space is informative. It contributes clearly to the classification performance based on the entire space for the chickenpieces datasets as in these cases the error for just the positive space, PES+ is clearly worse than for the entire space, PES. BrainMRI is an example of a dataset for which the non-Euclideaness is non-informative as the negative part of the space does not contribute. The artificial dataset Balls3D has been successfully constructed such that all information is in the negative part of the space: classes can be entirely separated by PES- and the positive part, PES+ can be better removed.

7 Examples

In this section we will discuss a few examples that are typical for the use of dissimilarities in structural pattern recognition problems.They have been published by us before [19] and are repeated here as they may serve well as an illustration.

7.1 Shapes

A simple and clear example of a structural pattern recognition problem is the recognition of blobs: 2D binary structures. An example is given in Fig. 5. It is an object out of the five-class chickenpieces dataset consisting of 445 images [2]. One of the best structural recognition procedure uses a string representation of the contour described by a set of segments of the same length [5]. The string elements are the consecutive angles of these segments. The weighted edit distances between all pairs of contours are used to compute the pairwise dissimilarities. This measure is non-Euclidean.

A $(\gamma,Ł)$ family of problems is considered depending on the specific choice for the cost of one editing operation γ as well as for the segment's length L used in the contour description. As a result, the classification performance depends on the parameters used, as shown in Fig 5, right. 10-fold cross-validation errors are shown there for the 1-NN rule directly applied on the dissimilarities as well as the results for the linear SVM computed by LIBSVM, [11], in the dissimilarity space. In addition, the results are presented for the average of the 11 dissimilarity matrices. We can observe that the linear classifier in the dissimilarity space (SVM-1) improves the traditional 1-NN results and that combining of the dissimilarities improves the results further on.

7.2 Histograms and spectra

Histograms and spectra offer very simple examples of data representations that are judged by human experts on their shape. In addition, also the sampling of the bins or wavelengths may serve as a useful vector representation for an automatic analysis. This is thanks to the fact that the domain is bounded and that spectra are often aligned. Below we give an example in which the dissimilarity representation outperforms the straightforward vector representation based on sampling because the first can correct for a wrong calibration (resulting in an imperfect alignment) in a pairwise fashion. Another reason to prefer dissimilarities for histograms and spectra over sampled vectorial data is that a dissimilarity measure encodes shape information. See the papers by Porro [47, 46] for more details.

We will consider now a dataset of 612 FL3-A DNA flowcytometer histograms from breast cancer tissues in a resolution of 256 bins. The initial data were ac-

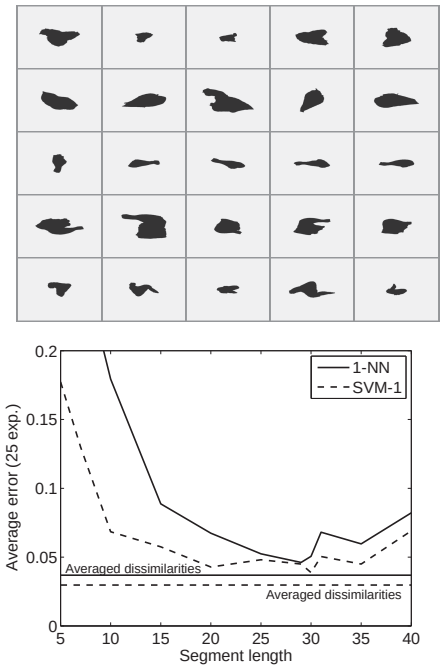

Fig. 5 Left: some examples of the chickenpieces dataset. Right: the error curves as a function of the segment length L.

quired by M. Nap and N. van Rodijnen of the Atrium Medical Center in Heerlen, The Netherlands, during 2000-2004, using the four tubes 3-6 of a DACO Galaxy flow cytometer. Histograms are labeled into three classes: aneuploid (335 patients), diploid (131) and tetraploid (146). We averaged the histograms of the four tubes thereby covering the DNA contents of about 80000 cells per patient. We removed the first and the last bin of every histogram as here outliers are collected, thereby obtaining 254 bins per histogram. Examples of histograms are shown in Fig. 6. The following representations are used:

Histograms. Objects (patients) are represented by the normalized values of the histograms (summed to one) described by a 254-dimensional vector. This representation is similar to the pixel representation used for images as it is based on just a sampling of the measurements.
Euclidean distances. These dissimilarities are computed as the Euclidean distances in the vector space mentioned above. Every object is represented by by a vector of distances to the objects in the training set.
Calibrated distances. As the histograms may suffer from an incorrect calibration in the horizontal direction (DNA content) for every pairwise dissimilarity we compute the multiplicative correction factor for the bin positions that minimizes their dissimilarity. Here we used the ℓ_1 distance. This representation makes use

of the shape structure of the histograms and removes an invariant (the wrong original calibration).

A linear SVM with a fixed trade-off parameter C is used in learning. The learning curves for the three representations are shown in the bottom right of Fig. 6. They illustrate how for this classifier the dissimilarity representation leads to better results than the vector representation based on the histogram sampling. The use of the background knowledge in the definition of the dissimilarity measure improves the results further on.

7.3 Images

The recognition of objects on the basis of the entire image can only be done if these images are aligned. Otherwise, earlier pre-procession or segmentation is necessary. This problem is thereby a 2-dimensional extension of the histogram and spectra recognition task.

We will show an example of digit recognition by using a part of the classic NIST database of handwritten numbers [52] on the basis of random subsets of 500 digits for the ten classes 0-9. The images were resampled to 32×32 pixels in such a way that the digits fit either horizontally or vertically. Fig. 7 shows a few examples: black is '1' and white is '0'. The dataset is repeatedly split into training and test sets and hold-out classification is applied. In every split the ten classes are evenly represented.
The following representations are used:

Features. We used 10 moments: the seven rotations invariant moments and the moments $[00], [01], [10]$, measuring the total number of black pixels and the centers of gravity in the horizontal and vertical directions.
Pixels. Every digit is represented by a vector of the intensity values in $32 * 32 = 1024$ dimensional vector space.
Dissimilarities to the training object. Every object is represented by the Euclidean distances to all objects in the training set.
Dissimilarities to blurred digits in the training set. As the pixels in the digit images are spatially connected blurring may emphasize this. In this way the distances between slightly rotated, shifted or locally transformed but otherwise identical digits become small.

The results are shown in Fig. 7 on the right. They show that the pixel representation is superior for large training sets. This is to be expected as this representation stores asymptotically the universe of possible digits. For small training sets a suitable set of features may perform better. The moments we use here are very general features. Better ones can be found for digit description. As explained before a feature-based description reduces the (information on the) object: it may be insensitive for some object modifications. For sufficiently large representation sets the dissimilarity representation may see all object differences and may thereby perform better.

7.4 Sequences

The recognition of sequences of observations is in particular difficult if the sequences of a given class vary in length, but capture the same 'story' (information) from the beginning to the end. Some may run faster, or even run faster over just a part of the story and slow down elsewhere. A possible solution is to rely on Dynamic Time Warping (DTW) that relates the sequences in a nonlinear way, yet obeys the order of the events. Once two sequences are optimally aligned, the distance between them may be computed.

An example in which the above has been applied successfully is the recognition of 3-dimensional gestures from the sign language [37] based on an statistically optimized DTW procedure [3]. We took a part of a dataset of this study: the 20 classes (signs) that were most frequently available. Each of these classes has 75 examples. The entire dataset thereby consists of a 1500×1500 matrix of DTW-based dissimilarities. The leave-one-out 1-NN error for this dataset is 0.041, which is based on the computation of 1499 DTW dissimilarities per test object. In Fig. 8, left, a scatterplot is shown of the first two PCA components showing that some classes can already be distinguished with these two features (linear combinations of dissimilarities).

We studied dissimilarity representations consisting of just one randomly drawn example per class. The resulting dissimilarity space has thereby 20 dimensions. New objects have to be compared with just these 20 objects. This space is now filled with randomly selected training sets, containing between 2 and 50 objects per class. Remaining objects are used for testing. Two classifiers are studied, the linear SVM (using the LIBSVM package [11]) with a fixed trade-off parameter $C = 100$ (we used normalized dissimilarity matrices with the average dissimilarities set to 100) and LDA. The experiment was repeated 25 times and the results averaged out.

The learning curves in Fig. 8, right, show the constant value of the 1-NN classifier performance using the dissimilarities to the single training examples per class only, and the increasing performances of the two classifiers for a growing number of training objects. Their average errors for 50 training objects per class is 0.07. Recall that this is still based on the computation of just 20 DTW dissimilarities per object as we work in the related 20-dimensional dissimilarity space. Our experiments show that LDA reaches an error of 0.035 for a representation set of three objects per class, i.e. 60 objects in total. Again, the training set size is 50 examples per class, i.e. 1000 examples in total. For testing new objects one needs to compute a weighted sum (linear combination) of 60 dissimilarity values giving the error of 0.035 instead of computing and ordering 1500 dissimilarities to all training objects for the 1-NN classifier leading to an error of 0.041.

7.5 Graphs

Graphs[2] are the main representation for describing structure in observed objects. In order to classify new objects, the pairwise differences between graphs have to be computed by using a graph matching technique. The resulting dissimilarities are usually related to the cost of matching and may be used to define a dissimilarity representation. We present here classification results obtained with a simple set of graphs describing four objects in the Coil database [38] described by 72 images for every object. The graphs are the Delaunay triangulations derived from corner points found in these images; see [53]. They are unattributed. Hence, the graphs describe the structure only. We used three dissimilarity measures:

CoilDelftSame Dissimilarities are found in a 5D space of eigenvectors derived from the two graphs by the JoEig approach; see [36]

CoilDelftDiff Graphs are compared in the eigenspace with a dimensionality determined by the smallest graph in every pairwise comparison by the JoEig approach; see [36]

CoilYork Dissimilarities are found by graph matching, using the algorithm of Gold and Ranguranjan; [24]

All dissimilarity matrices are normalized such that the average dissimilarity is 1. In addition to the three dissimilarity datasets we used also their averaged dissimilarity matrix.

In a 10-fold cross-validation experiment, with $R := T$, we use four classifiers: the 1-NN rule on the given dissimilarities and the 1-NN rule in the dissimilarity space (listed as 1-NND in Table 7.5), LDA on a PCA-derived subspace covering 99% of the variance and the linear SVM with a fixed trade-off parameter $C = 1$. All experiments are repeated 25 times. Table 7.5 reports the mean classification errors and the standard deviations of these means in between brackets. Some interesting observations are:

- The CoilYork dissimilarity measure is apparently much better than the two CoilDelft measures.
- The classifiers in the dissimilarity space however are not useful for the CoilYork measure, but they are for the CoilDelft measures. Apparently these two ways of computing dissimilarities are essentially different.
- Averaging all three measures significantly improves the classifier performance in the resulting dissimilarity space, even outperforming the original best CoilYork result. It is striking that this does not hold for the 1-NN rule applied to the original dissimilarities.

[2] Results presented in this section are based on a joint research with Prof. Richard Wilson, University of York, UK, and Dr. Wan-Jui Lee, Delft University of Technology, The Netherlands

Table 2 10-fold cross-validation errors averaged over 25 repetitions.

dataset	1-NN	1-NND	PCA-LDA	SVM-1
CoilDelftDiff	0.477 (0.002)	0.441 (0.003)	0.403 (0.003)	0.395 (0.003)
CoilDelftSame	0.646 (0.002)	0.406 (0.003)	0.423 (0.003)	0.387 (0.003)
CoilYork	0.252 (0.003)	0.368 (0.004)	0.310 (0.004)	0.326 (0.003)
Averaged	0.373 (0.002)	0.217 (0.003)	0.264 (0.003)	0.238 (0.002)

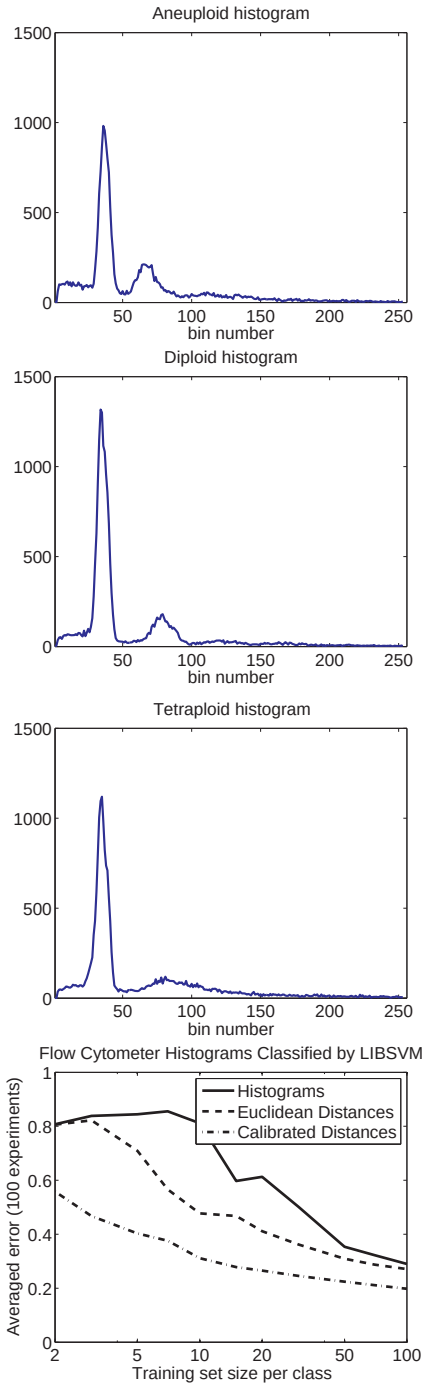

Fig. 6 Examples of some flowcytometer histograms: aneuploid, diploid and tetraploid. Bottom right shows the learning curves.

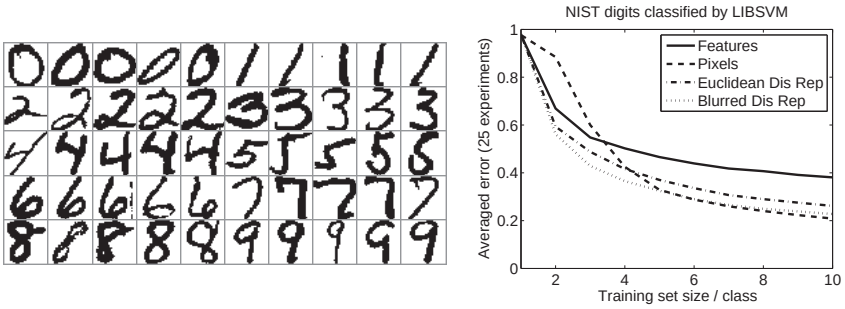

Fig. 7 Left: examples of the images used for the digit recognition experiment. Right: the learning curves.

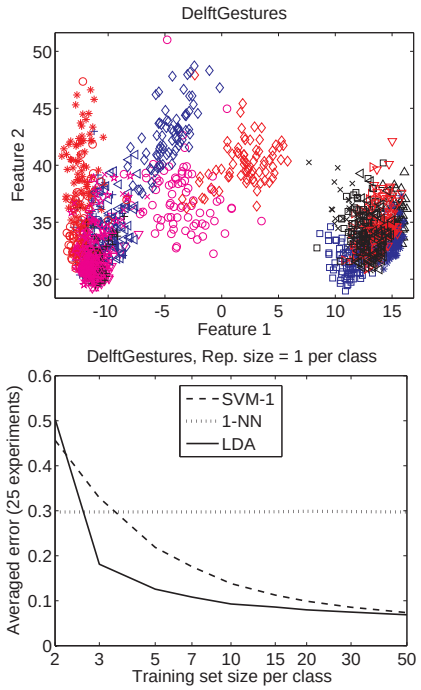

Fig. 8 PCA and learning curves for the 20-class Delft Gesture Dataset.

8 Discussion

The dissimilarity representation discussed in this chapter is in particular useful for applications in structural pattern recognition as it is a way the represent objects in their entirety. This may result in non=Euclidean or even non-metric dissimilarities. We have presented ways how to handle them, analyzed possible causes of the non-Euclideaness and answered the question whether such dissimilarity measures can be informative. Finally we presented a set of examples on real world data.

We will repeat and emphasize some significant observations and additionally touch a few topics that could not be treated.

In our analysis on the causes of non-Euclidean dissimilarities we made the observation that may be caused naturally in the process of comparing real world objects (Section /refsec:causes, in particular when vector spaces are defined on just a subset of the objects of interest. This implies that objects to be classified may have to be included in the analysis together with the training set (Section 2.2.3), so called transductive inference or transductive learning [51].

The non-Euclideaness is a problem when it is attempted to build vector spaces from given dissimilarity data. This bridges the fields of structural and statistical pattern recognition [17],[19], [22], [6]. Before this problem was faced researchers just used dissimilarities for template matching or approximated the non-Euclidean dissimilarities by Euclidean ones. In this chapter examples are given that show that the non-Euclidean part of the data (reflected in the so-called negative part of the pseudo-Euclidean space used for an isometrical embedding the dissimilarities, Section 3.2.1) can be informative for the classification, see Section 6.

In Section 7 a number of real world examples has been given that show that the dissimilarity approach can contribute significantly to the solution of pattern recognition problems. The use of the dissimilarity space is thereby advantageous. It avoids the computational complexity of embedding dissimilarities in a pseudo-Euclidean space as well as the Euclidean correction of this space or the problems of constructing classifiers. We judge the study of pseudo-Euclidean embedding especially of interest for studying the informativeness of the non-Euclidean characteristics. The dissimilarity space preserves all non-Euclidean information but is itself Euclidean, Section 5.

There are many interesting issues related to the dissimilarity approach discussed in this chapter. A number of them are discussed elsewhere or hardly investigated so far. A first, obvious question is that relating all objects to all other objects results into a computational explosion. Moreover, it may seem that there is not really a need to determine the dissimilarities to vary similar objects, which will become the case for growing training sets. Prototype selection is thereby of interest to reduce to size of the representation set. See [10] for some results and earlier references. Directly related to this is the question whether dissimilarities are useful for very large training sets. How to find the optimal set of prototypes for such cases? Is it possible to guarantee some asymptotically optimal result?

At this point it is relevant to realize the following. If objects show a zero distance if and only if they are identical and if they are labeled unambiguously then classes

do not overlap and a zero-error classifier is possible. What is the best way to reach this? Most classifiers assume class overlap. The study of classifiers that make use of the fact that classes do not overlap didn't make much progress after the definition of the original perceptron rule. The assumption of non-overlapping classes may also have a significant impact on the collection of training data and the definition of classifier performance. If classes do not overlap there is no need use a statistical approach based on density distributions. The definition of class domains may be sufficient. Training sets should in that case be representative for the domains and not for the distributions. This implies that it will be allowed to ask application experts for typical examples instead of selecting an i.i.d dataset representative for the data distribution.

For most practical applications there will be many ways to define dissimilarity measures that are zero if and only if the objects are identical. Combining such measures usually improves the results. In particular a straightforward averaging as applied in Section 7 is very interesting as is does not introduce additional parameters but just combines different types of information resulting in dissimilarity matrices of the same size and spaces of the same dimensionality in which data is better separable.

A new and significant application domain, next to structural pattern recognition, is the design of classification procedures for sets points in a feature space representing different parts of objects to be recognized, see Section 2.2.3. It is a generalization of the Multi-Instance Learning (MIL) problem and the bag-of-words classifiers. The proper design of dissimilarity measures between two sets of feature vectors representing two objects, adapted to the characteristics of the problem at hand is a fascinating issue [14], [35], [50], [54].

Once the basic tools for dissimilarity based classification are established, the next question will be to define the basic set of dissimilarity measures for various data types like for the above mentioned sets of feature vectors. For every more general domain of objects like images, spectra, time signals a set of basic dissimilarity measures should be available to define an initial solution for most problems. Like for the areas of feature extraction and classifiers the optimal approach should be tuned to the application, but the availability of a set of tools and examples may contribute to good solution of the pattern recognition problem at hand.

References

1. J. A. Anderson. Logistic discrimination. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics 2: Classification, Pattern Recognition and Reduction of Dimensionality*, pages 169–191, Amsterdam, 1982. North Holland.
2. G. Andreu, A. Crespo, and J. M. Valiente. Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition. In *Proceedings of ICNN'97, International Conference on Neural Networks*, volume II, pages 1341–1346. IEEE Service Center, Piscataway, NJ, 1997.
3. Claus Bahlmann and Hans Burkhardt. The writer independent online handwriting recognition system frog on hand and cluster generative statistical dynamic time warping. *IEEE Trans. Pattern Anal. Mach. Intell*, 26(3):299–310, 2004.
4. H. Bunke and U. Bühler. Applications of approximate string matching to 2D shape recognition. *Pattern recognition*, 26(12):1797–1812, 1993.
5. H. Bunke and U. Buhler. Applications of approximate string matching to 2D shape recognition. *Pattern Recognition*, 26(12):1797–1812, December 1993.
6. H. Bunke and K. Riesen. Graph classification based on dissimilarity space embedding. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 996–1007, 2008.
7. H. Bunke and A. Sanfeliu, editors. *Syntactic and Structural Pattern Recognition Theory and Applications*. World Scientific, 1990.
8. H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, 1998.
9. G. J. Burghouts, A. W. M. Smeulders, and J. M. Geusebroek. The distribution family of similarity distances. In *Advances in Neural Information Processing Systems*, volume 20, 2007.
10. Yenisel Plasencia Calana, Edel B. García Reyes, Mauricio Orozco-Alzate, and Robert P. W. Duin. Prototype selection for dissimilarity representation by a genetic algorithm. In *ICPR 2010*, pages 177–180, 2010.
11. C.-C. Chang and C.-J. Lin. LIBSVM: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/˜cjlin/libsvm.
12. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
13. N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, UK, 2000.
14. T. G Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.
15. M.P. Dubuisson and A.K. Jain. Modified Hausdorff distance for object matching. In *Int. Conference on Pattern Recognition*, volume 1, pages 566–568, 1994.
16. R. P. W. Duin and E. Pękalska. Non-Euclidean dissimilarities: Causes and informativeness. In E.R. Hancock et al., editor, *Proc. SSPR & SPR 2010 (LNCS)*, volume 6218, pages 324–333. Springer-Verlag, 2010.
17. Robert P. W. Duin. Non-euclidean problems in pattern recognition related to human expert knowledge. In Joaquim Filipe and José Cordeiro, editors, *ICEIS*, volume 73 of *Lecture Notes in Business Information Processing*, pages 15–28. Springer, 2010.
18. Robert P. W. Duin, Marco Loog, Elzbieta Pekalska, and David M. J. Tax. Feature-based dissimilarity space classification. In Devrim Ünay, Zehra Çataltepe, and Selim Aksoy, editors, *ICPR Contests*, volume 6388 of *Lecture Notes in Computer Science*, pages 46–55. Springer, 2010.
19. Robert P. W. Duin and Elzbieta Pekalska. The dissimilarity representation for structural pattern recognition. In *CIARP, (LNCS)*, volume 7042, pages 1–24. Springer, 2011.
20. R.P.W. Duin, D. de Ridder, and D.M.J. Tax. Experiments with object based discriminant functions; a featureless approach to pattern recognition. *Pattern Recognition Letters*, 18(11-13):1159–1166, 1997.
21. R.P.W. Duin and E. Pękalska. On refining dissimilarity matrices for an improved nn learning. In *ICPR*, pages 1–4, 2008.

22. R.P.W. Duin and E. Pękalska. The dissimilarity space: between structural and statistical pattern recognition. *Pattern Recognition Letters*, 33:826–832, 2012.
23. R.P.W. Duin, E. Pękalska, A. Harol, W.-J. Lee, and H. Bunke. On euclidean corrections for non-euclidean dissimilarities. In *SSPR/SPR*, pages 551–561, 2008.
24. Steven Gold and Anand Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(4):377–388, 1996.
25. L. Goldfarb. A new approach to pattern recognition. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 241–402. Elsevier, 1985.
26. L. Goldfarb. A new approach to pattern recognition. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 241–402. Elsevier, 1985.
27. J.C. Gower. Metric and Euclidean Properties of Dissimilarity Coefficients. *J. of Classification*, 3:5–48, 1986.
28. T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In *Advances in Neural Information System Processing 11*, pages 438–444, 1999.
29. 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 *ICANN*, pages 304–309, 1999.
30. B. Haasdonk. Feature space interpretation of SVMs with indefinite kernels. *IEEE TPAMI*, 25(5):482–492, 2005.
31. B. Haasdonk and E. Pękalska. Indefinite kernel fisher discriminant. In *ICPR*, pages 1–4, 2008.
32. D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with Non-Metric Distances: Image Retrieval and Class Representation. *IEEE TPAMI*, 22(6):583–600, 2000.
33. Anil K. Jain and Douglas E. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Trans. Pattern Anal. Mach. Intell*, 19(12):1386–1391, 1997.
34. T. Joachims. A probabilistic analysis of the rocchio algorithm with TFIDF for text categorization. In *Proceedings of International Conference on Machine Learning*, pages 143–151, 1997.
35. Risi Imre Kondor and Tony Jebara. A kernel between sets of vectors. In *ICML*, pages 361–368, 2003.
36. W. J. Lee and R. P. W. Duin. An inexact graph comparison approach in joint eigenspace. In *Structural, Syntactic, and Statistical Pattern Recognition*, pages 35–44, 2008.
37. J. F. Lichtenauer, E. A. Hendriks, and M. J. T. Reinders. Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(11):2040–2046, November 2008.
38. S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-100). In *Columbia University*, 1996.
39. E. Pękalska and R. P. W. Duin. Dissimilarity representations allow for building good classifiers. *Pattern Recognition Letters*, 23(8):943–956, June 2002.
40. E. Pękalska, R. P. W. Duin, and P. Paclik. Prototype selection for dissimilarity-based classifiers. *Pattern Recognition*, 39(2):189–208, February 2006.
41. E. Pękalska and R.P.W Duin. *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. World Scientific, Singapore, 2005.
42. E. Pękalska and R.P.W. Duin. Dissimilarity-based classification for vectorial representations. In *ICPR (3)*, pages 137–140, 2006.
43. E. Pękalska and R.P.W. Duin. Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(6):729–744, Nov. 2008.
44. E. Pękalska and B. Haasdonk. Kernel discriminant analysis with positive definite and indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1017–1032, 2009.
45. E. Pękalska, P. Paclík, and R.P.W. Duin. A Generalized Kernel Approach to Dissimilarity Based Classification. *J. of Machine Learning Research*, 2(2):175–211, 2002.

46. Diana Porro-Muñoz, Robert P. W. Duin, Isneri Talavera-Bustamante, and Mauricio Orozco-Alzate. Classification of three-way data by the dissimilarity representation. *Signal Processing*, 91(11):2520–2529, 2011.
47. Diana Porro-Muoz, Isneri Talavera, Robert P. W. Duin, Noslen Hernndez, and Mauricio Orozco-Alzate. Dissimilarity representation on functional spectral data for classification. *Journal of Chemometrics*, pages n/a–n/a, 2011.
48. Noor Azah Samsudin and Andrew P. Bradley. Nearest neighbour group-based classification. *Pattern Recognition*, 43(10):3458–3467, 2010.
49. Nicu Sebe, Michael S. Lew, and Dionysius P. Huijsmans. Toward improved ranking metrics. *IEEE Trans. Pattern Anal. Mach. Intell*, 22(10):1132–1143, 2000.
50. D.M.J. Tax, Marco Loog, Robert P. W. Duin, Veronika Cheplygina, and Wan-Jui Lee. *Bag dissimilarities for multiple instance learning*, volume LNCS 7005 of *Lecture Notes in Computer Science*, pages 222–234. Springer, 2011.
51. V. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.
52. C.L. Wilson and M.D. Garris. Handprinted character database 3. Technical report, National Institute of Standards and Technology, February 1992.
53. B. Xiao and E. R. Hancock. Geometric characterisation of graphs. In *CIAP*, pages 471–478, 2005.
54. Jun Yang, Yu-Gang Jiang, Alexander G. Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In James Ze Wang, Nozha Boujemaa, Alberto Del Bimbo, and Jia Li, editors, *Multimedia Information Retrieval*, pages 197–206. ACM, 2007.