Journal of Machine Learning Research 2 (2001) 175-211 Submitted 03/01; Published 12/01

A Generalized Kernel Approach to Dissimilarity-based Classification

Elżbieta Pękalska ELA@PH.TN.TUDELFT.NL
Pavel Paclík PAVEL@PH.TN.TUDELFT.NL
Robert P.W. Duin DUIN@PH.TN.TUDELFT.NL
Pattern Recognition Group, Faculty of Applied Sciences
Delft University of Technology
Lorentzweg 1, 2628 CJ Delft, The Netherlands

Editor: Nello Cristianini, John Shawe-Taylor and Robert Williamson

Abstract

Usually, objects to be classified are represented by features. In this paper, we discuss an alternative object representation based on dissimilarity values. If such distances separate the classes well, the nearest neighbor method offers a good solution. However, dissimilarities used in practice are usually far from ideal and the performance of the nearest neighbor rule suffers from its sensitivity to noisy examples. We show that other, more global classification techniques are preferable to the nearest neighbor rule, in such cases.

For classification purposes, two different ways of using generalized dissimilarity kernels are considered. In the first one, distances are isometrically embedded in a pseudo-Euclidean space and the classification task is performed there. In the second approach, classifiers are built directly on distance kernels. Both approaches are described theoretically and then compared using experiments with different dissimilarity measures and datasets including degraded data simulating the problem of missing values.

Keywords: dissimilarity, embedding, pseudo-Euclidean space, nearest mean classifier, support vector classifier, Fisher linear discriminant

1. Introduction

In this paper, we study the design of classifiers, directly based on a given set of dissimilarities between objects using a generalized kernel approach. Kernels are often understood as symmetric, positive definite functions of two variables, and thereby, they express similarity between objects represented in a feature space. Here, we propose to address a kernel in a more general way, i.e. as a proximity measure. The important difference is that a kernel describes now a proximity relation between two objects, which may not be explicitly represented in a feature space, or which may come from two different feature spaces. An example is a dissimilarity measure defined between shapes or contours, or images of different sizes. A suitable choice can be e.g. variants of the Hausdorff distance, comparing two sets of points (Dubuisson and Jain, 1994). In such a case, there is no need to define a common feature space, where the objects are represented. It means that there might be no functional dependence given specifically. We simply assume that a dissimilarity representation is just given.

Since a lot of attention has been paid to similarity kernels, this paper is devoted to classification aspects using dissimilarity (or distance) kernels. Here, we want to emphasize the importance of recognition tasks for which dissimilarity kernels are built directly on images or shapes. Therefore, no feature space is (or needs to be) originally defined, but dissimilarities arise directly from the application. Examples are Jain and Zongker (1997) and Jacobs et al. (2000). Therefore, concerning the notation, if we generally refer to objects, like r, s or x, they will *not* be printed in bold, to emphasize that they are not (or might not) be feature vectors. The distance kernel represents the information in a relative way, i.e. through pairwise dissimilarity relations between objects. The goal now is to learn *only* from such relational data, i.e. without any use (different than computing distances if necessary) of a starting original representation. The question studied here is, therefore, how to learn from data, given only dissimilarity kernels.

Recently, Schölkopf (2000) has proposed to treat kernels as generalized distance measures and also strengthened the link to algorithms based on positive definite kernels, such as the support vector classifier (SVC) (Vapnik, 1995, Schölkopf, 1997) or the kernel principal component analysis (Schölkopf et al., 1999, Schölkopf, 1997). There is, however, an essential difference with our approach. Schölkopf starts his reasoning from a feature space representation, where, in our case, this is not possible; we start a learning task from a dissimilarity kernel. For a more elaborate discussion, see Section 4.4.

Our principal question is how, given a dissimilarity kernel, a recognition problem can be tackled. To this end, we collected a set of methods which can be used by a researcher. In the paper we analyze the problem and illustrate possible solutions. Given a good dissimilarity measure, the k-nearest neighbor (k-NN) classifier is expected to perform well. It is, however, difficult to build such a measure for a complex recognition problem. In case of imperfect dissimilarities, the k-NN rule suffers from its sensitivity to noisy examples, but more global classifiers can perform better. How to design them is the goal of this research.

Two distinct approaches for classification tasks are studied in this paper. In the first one, a dissimilarity representation is isometrically embedded in a feature space, as presented in Section 3. This is always possible for a finite representation, although such a space may be pseudo-Euclidean (Goldfarb, 1985). Objects mapped to such a space preserve the structure of the data revealed in original distances. It means that if the dissimilarity measure defines classes which are bounded and compact, the configuration found in an underlying feature space should reflect those properties. An underlying feature space is constructed in the process of embedding. For this purpose, both the compactness hypothesis and its reverse should hold (see Section 2). If the dimensionality of such a feature space is low, then a classification task can be easily performed.

In the second approach, a dissimilarity kernel is interpreted as a mapping based on a chosen representation set R of objects (Duin, 2000, Pękalska and Duin, 2001), as presented in Section 5. In this formulation, classifiers can be constructed directly on the dissimilarity kernels, as in dissimilarity spaces.

The paper is organized as follows. Section 2 discusses dissimilarities in the light of the compactness hypothesis, in general. Section 3 presents some mathematical formulation of the linear embedding problem. Section 4 gives more details on building classifiers in the embedded space. Section 5 addresses the classification problem by the direct use of dissimilarity kernels. Sections 6 and 7 describe the experiments conducted, and discuss the results. Conclusions are summarized in Section 8.

2. On dissimilarity measures

In general, a classification problem can be solved based on the so-called compactness hypothesis (Arkadev and Braverman, 1966, Duin, 1999), which states that objects that are similar, are also close in their representations. Effectively, this puts a constraint on the dissimilarity measure d, which has to be such that $d(r,s)$ is small if objects r and s are similar, i.e. it should be much smaller for similar objects than for objects that are very different. For feature representations, the above does not hold the other way around: two entirely different objects may have the same feature representation. This does not cause a problem if these feature values are improbable for all or for all but one of the classes. For a dissimilarity kernel, however, the reverse of the compactness hypothesis also holds provided that the dissimilarity measure d poses some continuity. If we demand that $d(r,s) = 0$, if and only if objects r and s are identical, this implies that they belong to the same class. This can be extended somewhat by assuming that all objects s with a small distance to the object r, i.e. for which $d(r,s) < \epsilon$ for a positive ϵ being sufficiently small, are so similar to r, that they belong to the same class. Consequently, the dissimilarities of r and s to all objects x under consideration should be about the same, i.e. $d(r,x) \approx d(s,x)$. We conclude, therefore, that for dissimilarity representations satisfying the above continuity, the reverse of the compactness hypothesis holds: objects that are similar in their representation are also similar in reality and belong, thereby, to the same class (see Duin and Pękalska, 2001). As a result, classes do not overlap and the classification error may become zero for large training sets.

In order to interpret further such a hypothesis, let us recall the notion of metric. A distance measure d is called a metric when the following conditions are fulfilled:

- reflectivity, i.e. $d(x,x) = 0$
- positivity, i.e. $d(x,y) > 0$ if x is distinct from y
- symmetry, i.e. $d(x,y) = d(y,x)$
- triangle inequality, i.e. $d(x,y) < d(x,z) + d(z,y)$ for every z

Basically, reflectivity and positivity are crucial to define a proper dissimilarity measure. We do not accept a dissimilarity measure which is zero for two different objects, because it would violate the compactness hypothesis. On the other hand, negative dissimilarities would be difficult to interpret. Therefore, reflectivity and positivity conditions should always be fulfilled. If a distance measure is a metric, then the assumption on the continuity is fulfilled and, thereby, also the reverse of the compactness hypothesis by the means of symmetry and triangle inequality. However, whether the last two conditions are necessary for a dissimilarity kernel is disputable. Non-metric distances often seem to arise when shapes or objects in images are compared by template matching or when other type of distances are built e.g. in computer vision (Dubuisson and Jain, 1994, Jacobs et al., 2000). It has been also argued (Tversky, 1977, Jacobs et al., 2000) that the symmetry constraint might be too strong, especially when dissimilarities come from psychological judgments. Therefore, in some methods, asymmetric distances are also permitted. In this paper, symmetric distance measures not obeying the triangle inequality, and therefore not proper metrics, are included.

3. Linear embedding of dissimilarities

There is a number of ways to embed dissimilarity data in a feature space. Since, we are interested in a faithful configuration, a (non-)linear embedding is performed such that the distances are preserved as well as possible. Since nonlinear projections require more computational effort and, moreover, the way of projecting new points to the existing configuration is not straightforward (or not defined), the linear mappings are preferable. Here, isometric embeddings are considered.

3.1 Embedding of Euclidean distances

Let the representation set $R = \{p_1, p_2, \ldots, p_n\}$ refer to n objects (Duin, 2000, Pękalska and Duin, 2001). Given an Euclidean distance matrix $D \in \mathcal{R}^{n\times n}$ between those objects, a distance preserving mapping onto an Euclidean space can be found. Such a projection is known in the literature as a *classical scaling* or a *metric multidimensional scaling* (Young and Householder, 1938, Borg and Groenen, 1997, Cox and Cox, 1995). In other words, the dimensionality $k \leq n$ and the configuration $X \in \mathcal{R}^{n\times k}$ can be found such that the (squared) Euclidean distances are preserved. Note that having determined one configuration, another one can be found by a rotation or a translation. To remove the last degree of freedom, without loss of generality, the mapping will be constructed such that the origin coincides with the centroid (i.e. the mean vector) of the configuration X.[1]

To define X, the relation between Euclidean distances and inner products are used. First, it can be proven that

$$D^{(2)} = \boldsymbol{b}\,\mathbf{1}^T + \mathbf{1}\,\boldsymbol{b}^T - 2\,B,$$

(see Borg and Groenen, 1997, Appendix A.2), where $D^{(2)}$ is a matrix of square Euclidean distances, B is the matrix of inner products of the underlying configuration X, i.e. $B = X\,X^T$ and $\boldsymbol{b}$ is a vector of the diagonal elements of B. B can also be expressed as:

$$B = -\frac{1}{2}\,J\,D^{(2)}\,J, \tag{1}$$

where J is the centering matrix $J = I - \frac{1}{n}\mathbf{1}\,\mathbf{1}^T \in \mathcal{R}^{n\times n}$ and I is the identity matrix. J projects[2] the data such that the final configuration has zero mean. B is positive definite since it is a Gram matrix (Greub, 1975). Then, the factorization of B by its eigendecomposition can be found as:

$$X\,X^T = B = Q\,\Lambda\,Q^T, \tag{2}$$

where Λ is a diagonal matrix with the diagonal consisting of the first non-negative eigenvalues, ranked in descending order, followed by the zero values, and Q is an orthogonal matrix of the corresponding eigenvectors (Borg and Groenen, 1997). For $k \leq n$ non-zero eigenvalues, a k-dimensional representation X can be then found as:

$$X = Q_k\,\Lambda_k^{\frac{1}{2}}, \qquad Q_k \in \mathcal{R}^{n\times k}, \quad \Lambda_k^{\frac{1}{2}} \in \mathcal{R}^{k\times k}, \tag{3}$$

1. It is also possible that any arbitrary point of X would become the centroid.
2. A more general projection can be achieved, imposing that a weighted mean becomes zero, by which $J = I - \boldsymbol{w}\,\mathbf{1}^T$, where $\boldsymbol{w}$ is a weight vector, such that $\boldsymbol{w}^T\mathbf{1} = 1$ and $B = -\frac{1}{2}\,J^T\,D^{(2)}J$.

where Q_k is a matrix of the first k leading eigenvectors and $\Lambda_k^{\frac{1}{2}}$ contains the square roots of the corresponding eigenvalues. Note that X, determined in this procedure, is unique up to rotation (the centroid is now fixed), since for any orthogonal matrix T, $X\,X^T = (XT)\,(XT)^T$. Note also, that features of X are uncorrelated, since the estimated covariance matrix of X becomes:

$$Cov(X) = \frac{1}{n-1}\,X^T\,X = \frac{1}{n-1}\,\Lambda_k^{\frac{1}{2}}\,Q_k^T\,Q_k\,\Lambda_k^{\frac{1}{2}} = \frac{1}{n-1}\,\Lambda_k, \tag{4}$$

because Q_k is orthogonal.

3.2 Embedding of non-Euclidean distances

The matrix $B = -\frac{1}{2}\,J\,D^{(2)}\,J$ is positive definite if and only if the distance matrix $D \in \mathcal{R}^{n\times n}$ is Euclidean (Borg and Groenen, 1997, Gower, 1986, 1982). Therefore, for a non-Euclidean D, B is not positive definite, i.e. B has *negative* eigenvalues. As a result, X cannot be constructed from B, since it relies on the square roots of eigenvalues; see Formula (3). Two approaches are possible to address this problem in an Euclidean space:

- Only p positive eigenvalues are taken into account, resulting in a p-dimensional configuration $X = Q_p\,\Lambda_p^{\frac{1}{2}}$ $(p < k)$, for which distances approximate the original ones. Since the distances are positive, the largest negative eigenvalues in magnitude, are smaller than the largest positive eigenvalues. Also the sum of the positive eigenvalues is larger than the sum of magnitudes of the negative ones.

  A justification for choosing only positive eigenvalues can be found in Section 3.4, where the issue of noise influence is discussed. We argue that, in general, directly measured distances may be noisy, and therefore, they may not be perfectly Euclidean, which will result in small negative eigenvalues of B; see Figure 1 for an illustration. Therefore, by disregarding them, noise can be diminished.

- It is known that there exists a positive constant $c > |\lambda|$, where λ is the smallest (negative) eigenvalue of B, such that a new square Euclidean distance matrix might be created from $D^{(2)}$ by adding c to off-diagonal elements (Cox and Cox, 1995, Gower, 1986), i.e.:

  $$D_n^{(2)} = D^{(2)} + c\,(\mathbf{1}\,\mathbf{1}^T - I). \tag{5}$$

  Then, X can be again expressed in an Euclidean space. In practice, the eigenvectors remain the same and the value $\frac{c}{2}$ is added to the non-zero eigenvalues, giving the new eigenvalue matrix $\Lambda_k + \frac{c}{2}\,I$. This is equivalent to regularizing the covariance matrix of our configuration X, i.e. $Cov(X) = \frac{1}{n-1}\,(\Lambda_k + \frac{c}{2}\,I)$ and changing X respectively.

These two approaches transform the problem, so that a configuration X can be defined again in an Euclidean space. This is especially useful when the negative eigenvalues are relatively small in magnitude, which suggest, on the other hand, that the original distance measure is close to Euclidean. In such cases, those negative eigenvalues can be interpreted as a noise contribution. If the negative eigenvalues are relatively large, then by neglecting them, possibly important information has been rejected. There is still an open question about the consequences on classification tasks of transforming the problem into an Euclidean space, either by neglecting the negative eigenvalues or by directly enlarging $D^{(2)}$ by a constant.

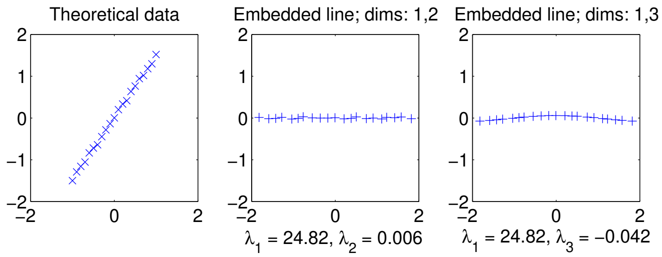


Figure 1: An example on the usefulness of disregarding the negative eigenvalues. Assume that the theoretical, original data is a perfect line, however due to the measurement process, somewhat distorted, as observed in the first plot. The distance kernel D is computed with distances $d_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^{1.004}$, which is nearly Euclidean. During the embedding process 16 eigenvalues are revealed, where 14 are negative (the largest negative in magnitude equals -0.042). This would suggest a possible 16-dimensional configuration, however, one significant positive eigenvalue indicates that the 'real' intrinsic dimensionality of the data is 1 (for an Euclidean distance and a perfect line, the embedded configuration is 1-dimensional). The second plot shows the projection onto first two dimensions (the configuration is retrieved up to a rotation). The last plot presents the projection onto the 1st and the 3rd dimensions, where the 3rd dimension corresponds to the largest (in magnitude) negative eigenvalue (how the embedding is done in such a case is explained by Formula 8). Notice how a tiny change of both the theoretical data and the Euclidean distance of a very simple problem enlarges the number of retrieved dimensions, from 1, in the perfect case, to 16.

Another possibility exists for problems in which the Euclidean space is not 'large enough'. Goldfarb (1984, 1985) proposed to project the data onto a pseudo-Euclidean space, which can be performed for any symmetric distance matrix. The pseudo-Euclidean space can be seen as consisting of two Euclidean spaces, for which the inner product operation is positive definite on the first space and negative definite on the second one (see Greub, 1975, Appendix Appendix A.1). For a simple illustration, see Figure 2. To embed the data, the same reasoning as for an Euclidean space is applied here. The essential difference refers to the notion of an inner product and a distance. Now, $B = -\frac{1}{2}\, J\, D^{(2)}\, J$, is still the matrix of inner products, but it is expressed as (see Appendix A.1):

$$B = X\, M\, X^T, \tag{6}$$

where M is a matrix of the inner product operation in a pseudo-Euclidean space. Following Goldfarb (1985), we can write (compare to Formula 2):

$$X\, M\, X^T = B = Q\, \Lambda\, Q^T = Q\, |\Lambda|^{\frac{1}{2}} \begin{bmatrix} M & \\ & 0 \end{bmatrix} |\Lambda|^{\frac{1}{2}}\, Q^T, \quad \text{where} \quad M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q}, \end{bmatrix} \tag{7}$$

and $p+q = k$. Λ is now based on p positive and q negative eigenvalues, which are presented in the following order: first, positive eigenvalues with decreasing values, then negative ones with decreasing magnitude and finally, zero values. Therefore, X can be now represented

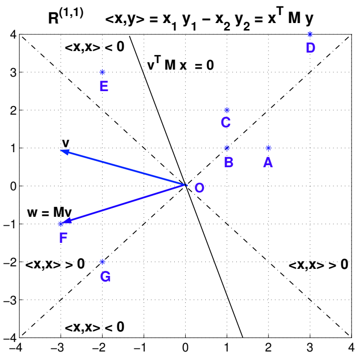


Figure 2: A pseudo-Euclidean space $R^{(1,1)}$, where $d^2(\boldsymbol{x},\boldsymbol{y}) = (\boldsymbol{x}-\boldsymbol{y})^T M(\boldsymbol{x}-\boldsymbol{y})$. Here, the length of any vector of the form $[x_1 \;\; \pm x_1]^T$, is zero. The orthogonal vectors are mirrored w.r.t. the lines $x_2 = x_1$ or $x_2 = -x_1$, e.g. $\langle OA, OC\rangle = 0$. Vector $\boldsymbol{v}$ defines the plane $\boldsymbol{v}^T M\boldsymbol{x} = 0$ in this space. Vector $\boldsymbol{w} = M\,\boldsymbol{v}$, a 'flipped' version of $\boldsymbol{v}$, describes the plane as if in an Euclidean space, i.e. it is perpendicular. This explains that in any pseudo-Euclidean space, the inner product operation can be seen as an Euclidean operation where one vector is 'flipped' by M. In general, distances can be of any sign, e.g.: $d^2(A,C) = d^2(F,G) = 0$, $d^2(A,B) = 1$, $d^2(B,C) = -1$, $d^2(D,A) = -8$, $d^2(F,A) = 21$ and $d^2(E,C) = 8$.

in a pseudo-Euclidean space $\mathcal{R}^k = \mathcal{R}^{(p,q)}$ of, the so-called, signature (p,q) (Goldfarb, 1984), as follows:

$$X = Q_k\,|\Lambda_k|^{\frac{1}{2}}. \tag{8}$$

Note that X has uncorrelated features, since the estimated pseudo-Euclidean covariance matrix (which is not positive definite as in an Euclidean space) is given as (Goldfarb, 1985):

$$Cov(X) = \frac{1}{n-1}\,X^T\,X\,M = \frac{1}{n-1}\,|\Lambda_k| M = \frac{1}{n-1}\,\Lambda_k. \tag{9}$$

This means that X is a result of a mapping in the sense of the PCA projection and the whole embedding procedure can be also interpreted as a sort of a kernel-PCA (Schölkopf et al., 1999, Schölkopf, 2000) approach, where the kernel B is a reproducing kernel for the pseudo-Euclidean feature space (see Section 4.4).

Computing square distances in a pseudo-Euclidean space $\mathcal{R}^{(p,q)}$ can be interpreted as computing the square Euclidean distance in a 'positive' space $\mathcal{R}^p$ and subtracting the square Euclidean distance found in a 'negative' space $\mathcal{R}^q$ (see Appendix A.1). The distances computed only in the 'positive' space are overestimated, therefore, the purpose of the 'negative' space is to correct them, e.g. make them be non-Euclidean. Since pseudo-Euclidean spaces, we are going to use, will result from the embedding process of positive distances, the contribution of the 'negative' space $\mathcal{R}^q$ to the overall distances is smaller than of the space $\mathcal{R}^p$. In such a case, due to the construction of X, X has relatively small feature values corresponding to the space $\mathcal{R}^q$. Practice confirms that many measured distances are close to the Euclidean one, giving relatively small negative eigenvalues in the embedding procedure. Note also, that the inner product $\boldsymbol{v}^T M\,\boldsymbol{x}$ can be interpreted as a traditional, Euclidean inner product $\boldsymbol{w}^T\boldsymbol{x}$, where $\boldsymbol{w} = M\boldsymbol{v}$. Therefore, having found a pseudo-Euclidean configuration X, a linear classifier $y = \boldsymbol{v}^T M \boldsymbol{x} + v_0$ can be found by addressing it as in a standard, Euclidean case, i.e. $y = \boldsymbol{w}^T \boldsymbol{x} + v_0$.

3.3 Embedding new points

Having found a configuration X in a pseudo-Euclidean space that preserves all pairwise distances $D(R,R)$, new objects can be can added to this space via the linear projection. Given the square distance matrix $D_n^{(2)} \in \mathcal{R}^{s\times n}$, expressing dissimilarities between s novel objects and all objects of the representation set R, a configuration X_n is to be determined in a pseudo-Euclidean space $\mathcal{R}^k = \mathcal{R}^{(p,q)}$. First, the matrix $B_n \in \mathcal{R}^{s\times n}$ of inner products relating all new objects to all objects from R should be found, which becomes (see Appendix A.3):

$$B_n = -\frac{1}{2}(D_n^{(2)}\, J - U\, D^{(2)}\, J), \tag{10}$$

where J is the centering matrix and $U = \frac{1}{n}\, \mathbf{1}^T \mathbf{1} \in \mathcal{R}^{s\times n}$. Since B_n can be expressed as:

$$X_n\, M\, X^T = B_n, \quad \text{with}$$

$$M = \begin{cases} I \in \mathcal{R}^{k\times k} & \text{if the space } \mathcal{R}^k \text{ is Euclidean,} \\ \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix} \in \mathcal{R}^{k\times k} & \text{if the space } \mathcal{R}^k \text{ is pseudo-Euclidean,} \end{cases} \tag{11}$$

therefore, X_n is found as the mean-square error solution to $X_n\, M\, X^T = B_n$, i.e. $X_n = B_n\, X\, (X^T\, X)^{-1}\, M$. Knowing that $X^T\, X = |\Lambda|$ and $X = Q_k\, |\Lambda_k|^{\frac{1}{2}}$, X_n is alternatively presented as:

$$X_n = B_n\, X\, |\Lambda|^{-1}\, M \;\text{ or }\; X_n = B_n\, Q_k\, |\Lambda_k|^{-\frac{1}{2}}\, M.$$

3.4 Reduction of dimensionality

By adding one object to the representation set R, (and, therefore, to the dissimilarity kernel $D(R,R)$), in practice one point is added to a finite pseudo-Euclidean space, but the dimensionality k of the vector representation might increase by more than one, contrary to the Euclidean case (Goldfarb, 1985). This means that both outliers and noise can contribute significantly to the resulting dimensionality k. In practice, when new points are added, they are projected onto the space determined by the starting configuration X. Therefore, the reliability of X, i.e. whether $D(R,R)$ describes a sufficiently well sampled representation set, plays an essential role in the process of representing new data, and consequently, the classification performance.

Originally, the (pseudo-)Euclidean configuration X is found such that the distances are preserved exactly and the dimensionality of X is determined by the number of non-zero eigenvalues of B. However, there might be many relatively small non-zero eigenvalues as compared to the large ones. Knowing that dissimilarities are noisy measurements, the small eigenvalues correspond to non-significant directions of X. In such a framework, neglecting small eigenvalues stands for reducing noise contribution (see Figure 3 for an illustration) or for finding a representation with the intrinsic dimension.

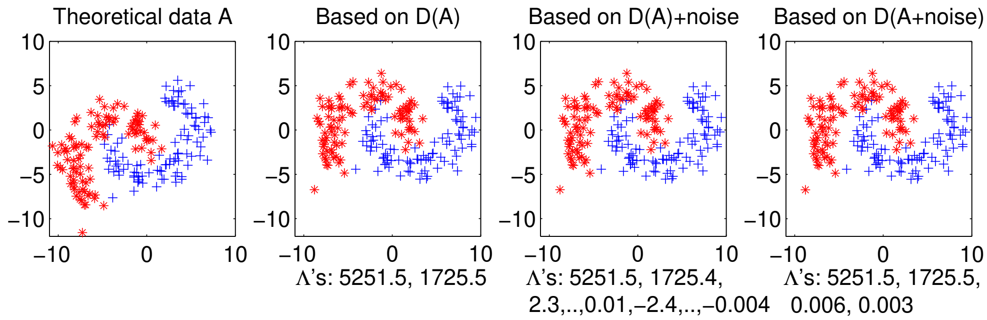


Figure 3: Noise influence on eigenvalues of B. The first, leftmost plot presents the 2D theoretical banana data (consisting of 200 points), for which the Euclidean distance matrix D has been computed. The second plot shows the result of the embedding of D into a 2D space (note that the retrieved configuration is up to a rotation). The third plot presents the projection onto the first 2 dimensions of the 199D data obtained via embedding of distorted distances $\tilde{D}$, (where $\tilde{d}_{ij} = d_{ij} + s_{ij}$ and $s_{ij} \sim N(0, 0.001)$), which become non-Euclidean. The last plot presents the projection onto the first 2 dimensions of the 4D data obtained via embedding of $D(\tilde{A})$, where $\tilde{A}$ consists of the theoretical data A to which 2 noisy features were added. Note that the first 2 largest eigenvalues, as given under the graphs, are about the same for non-distorted as well as for distorted data, which gives practically the same results in all cases. Therefore, by rejecting relatively small eigenvalues, noise is diminished.

In such a case, distances will be preserved approximately. One has, however, a control over the dimensionality of the reduced vector representation. Basically, the dimensionality reduction can be achieved by the orthogonal projection, governed by the PCA. The particular construction of $X = Q_k\,|\Lambda_k|^{\frac{1}{2}}$ and the fact that X is an uncorrelated vector representation, i.e. $Cov(X) = \frac{1}{n-1}\,\Lambda_k$, stand for X being given in the form of the orthogonal PCA projection; see Formula (9). It means that the reduction of dimensionality is performed in a simple way by neglecting directions corresponding to eigenvalues small in magnitude. The reduced representation (being an orthogonal projection) is then determined by the p' significant positive eigenvalues and q' significant (in magnitude) negative eigenvalues. Therefore, $X' \in \mathcal{R}^{n\times k'}$, $k' < k$, is found as $X' = Q_{k'}\,|\Lambda_{k'}|^{\frac{1}{2}}$, where $k' = p' + q'$ and $\Lambda_{k'}$ is a diagonal matrix of first, decreasing positive eigenvalues and then increasing negative eigenvalues, and $Q_{k'}$ is the matrix of corresponding eigenvectors.

4. Classification in a pseudo-Euclidean space

The symmetric dissimilarity kernel D can be seen as a description of an underlying, lower-dimensional vector representation X. If X is determined in an Euclidean space, then any traditional classifier can be constructed in such a space. If X happens to be a pseudo-Euclidean representation, then the conventional classifiers should be redefined. Here, we limit ourselves to simple, linear classification rules: nearest mean classifier, Fisher linear discriminant and support vector classifier, but, more advanced classifiers can be built as well.

4.1 Generalized nearest mean classifier

Nearest mean classifier (NMC) is the simplest linear classifier which assigns an unknown object to a class of the nearest mean. In a (pseudo-)Euclidean space $\mathcal{R}^k$, such a decision rule is based on the (pseudo-)Euclidean distance. Given D, assume a 2-class problem with the classes ω_1 and ω_2. The vector representation $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is such that it preserves the originally considered squared distances $D^{(2)}$. Let $\overline{\boldsymbol{x}}_{(i)}$ be the mean vector of the class ω_i. For a new object z represented in this space, as $\boldsymbol{z}_x$, the classification rule is defined as:

$$\begin{array}{ll} \text{Assign } z \text{ to } \omega_1 & \text{iff } d^2(\boldsymbol{z}_x, \overline{\boldsymbol{x}}_{(1)}) < d^2(\boldsymbol{z}_x, \overline{\boldsymbol{x}}_{(2)}) \\ \text{Assign } z \text{ to } \omega_2 & \text{otherwise} \end{array}$$

where $d^2(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x} - \boldsymbol{y}||^2 = \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{x} - \boldsymbol{y} \rangle = (\boldsymbol{x} - \boldsymbol{y})^T M (\boldsymbol{x} - \boldsymbol{y})$ and M is defined by Formula (11). Embedding a dissimilarity kernel D can be avoided when only class mean vectors and distances have to be computed. Therefore, we propose an alternative approach. A similar classification process can be carried out, however, without performing the exact mapping. As a result, the generalized nearest mean classifier (GNMC) will be obtained.

Assume first that a class ω is represented by a distance matrix $D(R, R)$ based on the representation set $R = \{p_1, \ldots, p_n\}$. Let a new object z be represented by distances to the set R. Then, the proximity of z to the class ω is measured by the function f_ω defined as:

$$\begin{aligned} f_\omega(z) &= \tfrac{1}{n} \textstyle\sum_{j=1}^n d^2(z, p_j) - V_d(R), \\ V_d(R) &= \tfrac{1}{2n^2} \textstyle\sum_{j=1}^n \sum_{k=1}^n d^2(p_j, p_k), \end{aligned} \tag{12}$$

where $V_d(R)$ is a generalized variability of the underlying feature space. It can be shown (see Appendix A.4) that the following holds:

$$\begin{aligned} f_\omega(z) &= ||\boldsymbol{z}_x - \overline{\boldsymbol{x}}||^2 = d^2(\boldsymbol{z}_x, \overline{\boldsymbol{x}}), \\ V_d(X) &= \tfrac{1}{n} \textstyle\sum_{j=1}^n ||\boldsymbol{x}_j||^2 - ||\overline{\boldsymbol{x}}||^2, \end{aligned} \tag{13}$$

where $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is found by a linear embedding, as introduced in Section 3 and $\boldsymbol{z}_x$ is the representation of the object z in space $\mathcal{R}^k$.

From dependencies given in (13), two important observations can be made. The first one refers to V_d expressing a variability in X. If X is a 1-dimensional vector, then V_d coincides with the variance of X. For X being a higher dimensional Euclidean representation, V_d is equal to the sum of variances, i.e. trace $(cov(X))$. For a pseudo-Euclidean space, V_d stands for a generalized variability, based on the pseudo-Euclidean covariance matrix; see Formula (9). The second observation refers to the function $f_\omega(z)$ which measures the distance of the point $\boldsymbol{z}_x$ to the mean of the class ω, both represented in the space $\mathcal{R}^k$. The interesting point is that such a distance can be computed without performing the embedding process explicitly, since it operates only on the given distances D, as presented in (12).

This result allows us to define a generalized nearest mean classifier as follows:

$$\text{Assign } z \text{ to } \omega_j : f_{\omega_j}(z) = \min_i \{f_{\omega_i}(z)\}, \tag{14}$$

where f_{ω_i} is defined as in (12). In other words, z is assigned to a class of the nearest mean (centroid), where each centroid is described in an underlying space defined by the within-class distances. Note that in the formulation given by (14), the classification rule holds for any number of classes.

The NMC and the GNMC in a pseudo-Euclidean space are not, in general, identical classifiers. The NMC finds a linear embedding onto a feature space $\mathcal{R}^k$ based on the whole distance matrix $D^{(2)}$. Therefore, the dimensionality of such a space is determined by both the within-class and between-class distances. The GNMC operates only on the within-class distances. Although the embedding is not performed directly, the GNMC works in the underlying feature spaces $\mathcal{R}^k_{\omega_i}$, for each class separately. It may happen that the signatures of feature spaces $\mathcal{R}^k_{\omega_i}$ are not the same. In such a case, the performances of the the NMC and the GNMC differ, because the NMC unifies pseudo-Euclidean space and the signature for all classes, while the GNMC treats them separately, which allows to describe them properly. Since the GNMC makes use of the distinct signatures, its accuracy is expected to be higher for problems in which the classes behave differently.

4.2 Fisher linear discriminant

The linear classifier (or a separating hyperplane) in a pseudo-Euclidean space $\mathcal{R}^k = \mathcal{R}^{(p,q)}$ is defined as follows (Greub, 1975, Goldfarb, 1985):

$$f(\boldsymbol{x}) = \boldsymbol{v}^T M\, \boldsymbol{x} + v_0, \quad \text{where } \; M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix} \tag{15}$$

To construct the Fisher linear discriminant (FLD), the notion of a pseudo-Euclidean covariance matrix is needed. For the representation X, it is defined as (Goldfarb, 1985):

$$cov(X) = \frac{1}{n-1} \left[\sum_{i=1}^{n} (\boldsymbol{x}_i - \overline{\boldsymbol{x}})\, (\boldsymbol{x}_i - \overline{\boldsymbol{x}})^T \right] M, \quad \text{where } \; \overline{\boldsymbol{x}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i.$$

Making use of the above definition and following Goldfarb (1985), the FLD, obtained by maximizing the ratio of between-scatter to within-scatter (Fisher criterion) (Fukunaga, 1990), for a 2-class problem is given by:

$$\begin{aligned} \boldsymbol{v} &= M\, C_W^{-1}\, (\boldsymbol{m}_1 - \boldsymbol{m}_2), n \\ v_0 &= -\tfrac{1}{2}\, (\boldsymbol{m}_1 + \boldsymbol{m}_2)^T \underbrace{M\, M}_{=I}\, C_W^{-1}\, (\boldsymbol{m}_1 - \boldsymbol{m}_2), \end{aligned}$$

where $C_W\, M$ is the pooled within-class covariance matrix in a pseudo-Euclidean space (C_W is the pooled within-class covariance matrix as computed for the Euclidean case) and $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$ stand for the class means. The FLD in a pseudo-Euclidean space can be constructed as the hyperplane $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + v_0$, where $\boldsymbol{w} = M\boldsymbol{v}$ and $\boldsymbol{w}^T \boldsymbol{x}$ refers to an Euclidean inner product. See Figure 4 as an illustration of a simple problem.

4.3 Support vector classifier

Let n points $\boldsymbol{x}_i$, $i = 1, 2, \ldots, n$ be given in an Euclidean space $\mathcal{R}^k$. Each point $\boldsymbol{x}_i$ belongs to one of two classes as described by the corresponding label $y_i \in \{-1, 1\}$. The goal for non-overlapping classes is to find the optimal hyperplane: $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0$, which maximizes the margin, i.e. $\frac{2}{||\boldsymbol{w}||^2}$ (Vapnik, 1995, Burges, 1998) (or, alternatively, minimizes $||\boldsymbol{w}||^2$). For non-linearly separable classes, non-negative slack variables ξ_i are introduced, allowing for

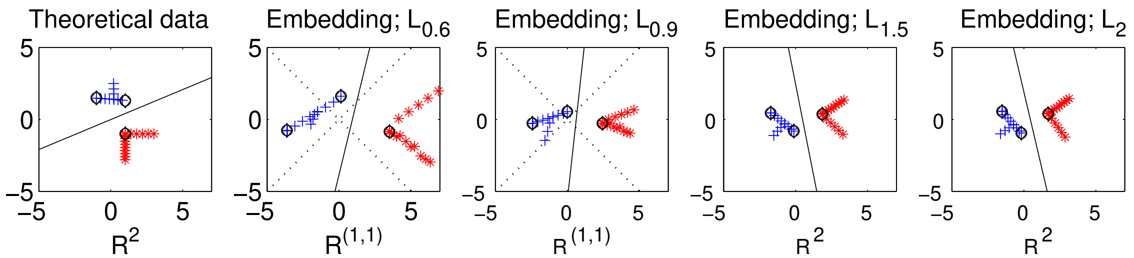


Figure 4: An illustration of the decision boundary of the FLD in an embedded space. The leftmost plot presents a 2D theoretical, artificial data. There are 3 training points, marked by circles. Only 3 objects are taken for training, because then the data can be perfectly embedded in not more than 2 dimensions. The remaining points, marked by '+' and '*' belong to the examples of testing data, which illustrate how the new objects are projected on the retrieved (pseudo-)Euclidean space. The following plots show the embedding results of the L_p distance D, where $d_{ij} = (\sum_{k=1}^{2} |x_{ik} - x_{jk}|^p)^{1/p}$, for $p = \{0.6, 0.9, 1.5, 2\}$. For positive p smaller than 1, the L_p distance is not metric and in such cases the distances between close objects are emphasized. In all subplots, the FLD, determined by the 3 circles, in the original (first subplot) or the embedded spaces is drawn. For $p = 2$, the original, theoretical data is retrieved up to a rotation.

classification errors, so that the soft margin linear support vector classifier (SVC) (Vapnik, 1995, Burges, 1998) is found as the solution of the quadratic programming procedure:

$$\begin{array}{lll} \text{Minimize} & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\,\sum_{i=1}^{n}\xi_i & \\ \text{s.t.} & y_i\,(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, n \\ & \xi_i \geq 0 \end{array} \tag{16}$$

The term $\sum_{i=1}^{n}\xi_i$ is an upper bound on the misclassification of the training samples and C can be regarded as a regularization parameter, a trade-off between the number of errors and the width of the margin. The dual programming formulation is given as follows:

$$\begin{array}{ll} \text{Maximize} & -\frac{1}{2}\,\boldsymbol{\alpha}^T\,\mathrm{diag}(\boldsymbol{y})\,K\,\mathrm{diag}(\boldsymbol{y})\,\boldsymbol{\alpha} + \boldsymbol{\alpha}^T\mathbf{1} \\ \text{s.t.} & \boldsymbol{\alpha}^T\,\boldsymbol{y} = 0 \\ & 0 \leq \alpha_i \leq C, \qquad i = 1, 2, \ldots, n \end{array} \tag{17}$$

where K is an $n \times n$ kernel matrix such that $K_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$. After solving the problem (17), the weight vector $\boldsymbol{w}$ is found to be a linear combination of the data vectors, giving $\boldsymbol{w} = \sum_{i=1}^{n}\alpha_i\,y_i\,\boldsymbol{x}_i$. Since many α_i become zero, the data points $\boldsymbol{x}_i$ with positive α_i, are the so-called support vectors. Only they contribute to the hyperplane equation. As a result, the discrimination function can be presented in terms of inner products as follows:

$$f(\boldsymbol{x}) = \sum_{\alpha_i > 0} \alpha_i\,y_i\,\langle \boldsymbol{x}, \boldsymbol{x}_i \rangle + w_0. \tag{18}$$

Note that in the training stage, the kernel matrix K becomes $K = X\,X^T$, which equals B, the matrix of inner products (see Formula 1), i.e. $K = B$.

Since the linear SVC is based only on inner products, and by the linear relation (1) between $D^{(2)}$ and B $(= K)$, the SVC can be easily constructed in the underlying feature space without performing the embedding explicitly, provided that the distances are Euclidean. For novel objects, represented by the distances to the representation set R, the SVC can be immediately tested by using B_n, the matrix of inner products between new objects and objects embedded originally, as provided by Formula (10).

For a non-Euclidean dissimilarity matrix D, the matrix of inner products is not positive definite, resulting in a pseudo-Euclidean space, The configuration X is given by (8), i.e. $X = Q\,|\Lambda|^{\frac{1}{2}}$, for which a linear classifier is defined by (15). If we now assign $\boldsymbol{w}^T = \boldsymbol{v}^T M$, then the classifier $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + v_0$ can be treated in an Euclidean space. The operation of $\boldsymbol{v}^T M$ is seen as flipping the values of the $\boldsymbol{w}$ vector in all 'negative' directions of the pseudo-Euclidean space. As suggested by Graepel et al. (1999a), this is equivalent to flipping the negative eigenvalues to positive ones, and considering the inner product $B' = Q\,|\Lambda|\,Q^T$ in an Euclidean space as being positive definite.

Summarizing, for a dissimilarity kernel D, the SVC classifier can be built in the underlying feature space as follows. First, the matrix B is computed according to (1). If B is not positive definite, then the matrix B' is computed as $B' = Q\,|\Lambda|\,Q^T$, otherwise $B' = B$. It describes a positive definite kernel, used to construct the SVC according to (18). Note also that any polynomial SVC classifier (Vapnik, 1995) can be built by using B' directly.

4.4 Discussion on the kernel trick for distances

Recently, Schölkopf (2000) has considered kernels as generalized dissimilarity measures. His reasoning starts from the observation that a Mercer kernel K (Vapnik, 1995), i.e. a positive definite kernel, can be seen as a (nonlinear) generalization of the similarity measure based on inner products. This is possible because such a kernel can be expressed as an inner product operation in some high-dimensional feature space $\mathcal{G}$, i.e. $K(\boldsymbol{x},\boldsymbol{y}) = \langle\phi(\boldsymbol{x}),\phi(\boldsymbol{y})\rangle$, where ϕ is a mapping, $\phi : \mathcal{F} \to \mathcal{G}$ and $\phi(\boldsymbol{x})$ is the image of $\boldsymbol{x}$ in the space $\mathcal{G}$. Following the same idea, Schölkopf considers a generalization of the squared Euclidean distance in the space $\mathcal{G}$, by using, the so-called *kernel trick*:

$$||\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})||^2 = K(\boldsymbol{x},\boldsymbol{x}) + K(\boldsymbol{y},\boldsymbol{y}) - 2\,K(\boldsymbol{x},\boldsymbol{y}),$$

which allows to express this distance only by using the kernel, without explicitly performing the mapping. Next, Schölkopf argues that a larger class of kernels, namely the *conditionally positive definite* kernels, can be used. A symmetric function $K : \mathcal{F}\times\mathcal{F} \to \mathcal{R}$ which for all $m \in \mathcal{N}$, all vectors $\boldsymbol{c} \in \mathcal{R}^{m\times 1}$ and all $\boldsymbol{x}_i \in \mathcal{F}$ fulfills:

$$\boldsymbol{c}^T K\,\boldsymbol{c} \geq 0, \quad \text{where} \;\; K_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{19}$$

is called a positive definite (p.d.) kernel. When the inequality (19) is satisfied for $\boldsymbol{c}$ such that $\boldsymbol{c}^T\mathbf{1} = 0$, the kernel is called conditionally positive definite (c.p.d.). A relation between the p.d. and c.p.d. kernels can be established as ('†' stands for a conjugate transpose):

$$\tilde{K} = \frac{1}{2}\,(I - \mathbf{1}\boldsymbol{w}^\dagger)\,K\,(I - \boldsymbol{w}\mathbf{1}^\dagger), \quad \text{where} \;\; \boldsymbol{w}\mathbf{1}^\dagger = 1 \tag{20}$$

i.e. $\tilde{K} \in \mathcal{R}^{n\times n}$ is p.d. if and only if K is c.p.d.

In the simplest case, $c_i = \frac{1}{n}$ for all i, (20) is then equal to Formula (1), if we read $B = \tilde{K}$ and $D^{(2)} = -K$. This means that D is an Euclidean distance matrix, if and only if $-D^{(2)}$ is a c.p.d. kernel.

For K, being a c.p.d. kernel, with $K(\boldsymbol{x}, \boldsymbol{x}) = 0$ for all $\boldsymbol{x} \in \mathcal{F}$, there exists a Hilbert space $\mathcal{H}$ of real-valued functions on $\mathcal{F}$ and a mapping $\phi : \mathcal{F} \to \mathcal{H}$, such that

$$||\phi(\boldsymbol{x}) - \phi(\boldsymbol{y})||^2 = -K(\boldsymbol{x}, \boldsymbol{y}). \tag{21}$$

This supports the fact that an Euclidean distance kernel D can be embedded in an Euclidean space, which is an example of a Hilbert space. This is justified by $K = -D^{(2)}$ being a c.p.d. kernel. This means that in this paper a particular case of the mapping ϕ is considered.

More generally, Schölkopf proves that for a real-valued, symmetric kernel $\tilde{K}$, there exists a linear space $\mathcal{H}$ (which might not be a Hilbert space) of real-valued functions on $\mathcal{F}$, endowed with a symmetric, bilinear form $Q(\cdot, \cdot)$ and a mapping $\phi : \mathcal{F} \to \mathcal{H}$, such that

$$\tilde{K}(\boldsymbol{x}, \boldsymbol{y}) = Q(\phi(\boldsymbol{x}), \phi(\boldsymbol{y})).$$

$\tilde{K}$ is then a reproducing kernel for a feature space $\mathcal{H}$ (Wahba, 1999).

According to the definition, a pseudo-Euclidean space (Greub, 1975)is equipped with a non-degenerate, indefinite, symmetric bilinear form $Q = \langle \cdot, \cdot \rangle$, seen as a generalized inner product (see Appendix A.1). This justifies that for a non-Euclidean distance kernel D, $\tilde{K} = B = \frac{1}{2}\, J\, K\, J$, where $K = -D^{(2)}$ (see Formula 1), is a reproducing kernel for a pseudo-Euclidean feature space $\mathcal{H}$.

Schölkopf argues also that the c.p.d. kernels K are ‘ a natural choice whenever we are dealing with a translation invariant problem’, where the SVC or the kernel-PCA are of an example. From Formula (21) it follows that $-K$ is the squared Euclidean distance in some Hilbert space.

In summary, Schölkopf provides a new framework for distance based algorithms. The squared Euclidean distance can now be realized in another feature space by using a suitable kernel function, which should be conditionally positive definite. Using Formula (20), a c.p.d. kernel can be transformed into a p.d. kernel, to which, again, the kernel algorithms could be applied. In this way, Schölkopf's work provides a mathematical context of our approach of embedding distances. It gives more information on relations between the c.p.d. kernels and p.d. kernels, or in other words, it names the class of c.p.d. kernels that can be isometrically embedded in an Euclidean space. However, Schölkopf starts from a given feature space and a known dissimilarity measure. We assume that a distance kernel is given implicitly by a dataset, maybe not even knowing what type of a measure it is or how it was computed. One of our approaches relies on a linear embedding, which is a particular case of a mapping ϕ considered by Schölkopf. While Schölkopf focuses mostly on a mathematical formulations, we try to study how the methods work in practice.

5. Classification on dissimilarities

The second approach, mentioned in the introduction, addresses the dissimilarity kernel as a mapping defined by the representation set $R = \{p_1, \ldots, p_n\}$. A mapping $D(z, R) : \mathcal{F} \to \mathcal{R}^n$ is defined as $D(z, R) = [d(z, p_1)\, d(z, p_2)\, \ldots\, d(z, p_n)]^T$. Notice that $\mathcal{F}$ expresses an original

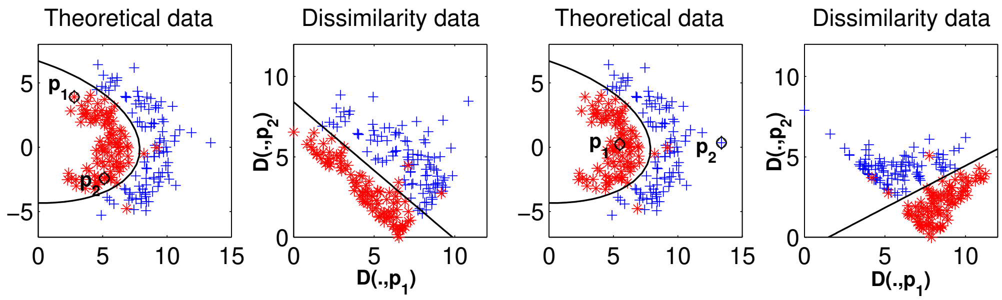


Figure 5: A simple illustration of a 2D dissimilarity space. The first and third plots show the theoretical, artificial data, with a quadratic classifier. The L_p distance D, where $d_{ij} = (\sum_{k=1}^{2} |x_{ik} - y_{jk}|^{5/2})^{2/5}$ was computed for this data. The representation set consist of two objects, i.e. $R = [p_1, p_2]$. The second and the fourth plots present the dissimilarity spaces $D(\cdot, R)$, where the representative objects are marked by circles on the first and third plots. Note that if R is well chosen, a linear classifier on a dissimilarity kernel $D(\cdot, R)$ separates the data very well.

feature space of objects, which might not be given explicitly. The dimensionality of such a dissimilarity space is controlled by the size of R. Using this formulation, classifiers can be constructed directly on the dissimilarity kernels, as in the dissimilarity space.

A justification to construct classifiers in dissimilarity space is as follows. The property that distances should be small for similar objects, i.e. belonging to the same class, and larger for objects of different classes, gives a possibility for a discrimination and, thereby, $D(\cdot, p_i)$, defined by the distances to the representative p_i, can be interpreted as a feature. If, p_i is a characteristic object of a particular class, then the discrimination power of $D(\cdot, p_i)$ can be large. On the other hand, if p_i is a non-typical object of its class, then the $D(\cdot, p_i)$ may discriminate poorly.

Defining a well-discriminating dissimilarity measure for a non-trivial recognition problem is difficult. On the other hand, when a good distance measure is derived, we almost solved our classification problem. It is still a challenge, especially if such a measure should preferably incorporate invariances. Building such a measure is equivalent to defining good features for a traditional classification problem. If a good measure can be found, then the k-nearest neighbor (k-NN) method is expected to perform well provided that D is metric or nearly metric. The decision of the k-NN is based on local neighborhoods only and it is, in general, sensitive to noise. It means that k nearest neighbors found might not include the best representatives of a class to which an object should be assigned. Moreover, the k-NN does not work for an asymmetric distance measure or might perform very badly for dissimilarities which strongly do not obey the triangle inequality. In such cases, a better generalization can be achieved by a classifier built in a dissimilarity space.

It might be better to include all the distances to determine the good representatives in a training process. For instance, a linear classifier in a dissimilarity space is a weighted linear combination of dissimilarities between an object and the representation set R. The weights are optimized on the training set, and large weights (in magnitude) emphasize objects which play an essential role during discrimination. By doing this, a more global classifier can be built, by which its sensitivity to noisy representative examples is reduced. Our experience confirms that a linear or quadratic classifier can often generalize better than the k-NN rule, especially for a small representation set R (see (Pękalska and Duin, 2001)).

A linear classifier built on the dissimilarity kernel D is given by:

$$f(D(x,R)) = \sum_{j=1}^{n} w_j\, d(x,x_j) + w_0 = \boldsymbol{w}^T\, D(x,R) + w_0 \tag{22}$$

and a linear classifier on the dissimilarity kernel $D^{(2)}$ is expressed as:

$$f(D^{(2)}(x,R)) = \sum_{j=1}^{n} w_j\, d^2(x,x_j) + w_0 = \boldsymbol{w}^T\, D^{(2)}(x,R) + w_0. \tag{23}$$

There is an essential difference between those two separating hyperplanes. Because of the linear relation between the square distance matrix $D^{(2)}$ and the matrix of inner products, the classifier (23) built on $D^{(2)}$ is, in fact, a quadratic classifier in the underlying space. The linear classifier (22), constructed on dissimilarity kernel D, is, in general, non-quadratic, nonlinear classifier in the underlying feature space, since there is a nonlinear relation between the inner products and the kernel D.

5.1 The Fisher Linear Discriminant (FLD)

In general, any traditional classifier operating on feature spaces might be used on dissimilarity kernels. Since most of the commonly-used dissimilarity measures are based on sums of differences between measurements, the choice of the linear Bayesian classifier, assuming normal densities, is a natural consequence of the central limit theorem applied to them. In principle, the quadratic Bayesian classifier could be even better, but it requires much more training objects for estimation of the class covariance matrices. It is known that for 2-class problems with equally probable classes, this classifier is equivalent to the Fisher linear discriminant (FLD), obtained by maximizing the so-called Fisher criterion, i.e. the ratio of between-scatter to within-scatter (Duda et al., 2001, Fukunaga, 1990). Therefore, we refer to this separating hyperplane as to the FLD.

The FLD can be now constructed on D in the form of (22). As a starting point, the representation set R, consisting of n objects, and the training set T coincide, i.e. $T = R$. In such a case, we have to deal with a small sample size classification problem, i.e. with n vectors $D(x_i, R)$ in n dimensions. We have recently proposed to use a reduced representation set R' of the size $r < n$, which is especially of importance when the distances are expensive to compute (Pękalska and Duin, 2001). The easiest way to choose R' is by a random selection. Also r objects can be chosen such that the minimum distance between any of them is maximized. Another possibility is based on a greedy approach. Starting from a randomly chosen object, in an iterative procedure, an object is added, which is the most dissimilar to all objects already chosen. It might be done globally or for each class separately. In case the most dissimilar objects are chosen they are likely to be outliers or positioned on the boundary. There exist many other ways to determine a reduced representation set R', but they will not be investigated here.

After R' has been established, a linear classifier is built on $D(T, R')$. For a new object, only distances to the set R' have to be computed. For a 2-class problem, with the prior probabilities p_{ω_1} and p_{ω_2}, the linear Bayesian classifier (Fukunaga, 1990) is constructed on the dissimilarity kernel D as follows:

$$f(D(x, R')) = \boldsymbol{w}^T\, D(x, R')^T + w_0,$$

where $\boldsymbol{w} = C_W^{-1}\,(\boldsymbol{m}_1 - \boldsymbol{m}_2)$ and $w_0 = -\frac{1}{2}\,(\boldsymbol{m}_1 + \boldsymbol{m}_2)^T\, C_W^{-1}\,(\boldsymbol{m}_1 - \boldsymbol{m}_2) + log(\frac{p_{\omega_1}}{p_{\omega_2}})$. C_W is the pooled within-class covariance matrix and $\boldsymbol{m}_i$, $i = 1, 2$, stands for the class mean in the dissimilarity space $D(\cdot, R)$.

When T and R' are identical (or $R' = R$), C_W becomes singular, and the linear classifier cannot be built. Regularization can be used instead, yielding an approximated covariance matrix $(1 - \lambda)\, C_W + \lambda\, I$. A classifier based on the regularized covariance matrix is called the regularized linear discriminant (RLD).

5.2 Linear programming (LP) machines

With a properly defined objective function and constraints for a dissimilarity kernel, a separating hyperplane can be obtained by solving a linear programming problem. Assume a 2-class problem, with classes ω_1 and ω_2 of the cardinality n_1 and n_2, respectively, and the labels $y_i = \{1, -1\}$. Let f be the separating hyperplane, built for the complete representation set R (i.e. $R = T$) as given by (22). Then, the simple optimization problem, minimizing the number of misclassification errors ξ_j, can be defined as:

$$\begin{aligned} &\text{Minimize} && \textstyle\sum_{i=1}^n s_i\, \xi_i \\ &\text{s.t.} && y_i\, f(D(\boldsymbol{x}_i, R)) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ &&& \xi_i \geq 0 \end{aligned} \tag{24}$$

where $s_i = 1$ for all i, or $s_i = \frac{1}{n_i}$ depending on a class label y_i. It is argued by Bennett and Mangasarian (1992) that the latter formulation can guarantee a nontrivial solution even when mean vectors of two classes happen to be the same. Such a problem can be solved by the standard optimization methods, such as simplex algorithm or interior-point methods. Since no other constraints are included, the hyperplane is constructed in an n-dimensional dissimilarity space $D(\cdot, R)$. It is possible, however, to impose a sparse solution, by minimizing the norm L_1 of the weight vector $\boldsymbol{w}$ of the hyperplane (22) i.e. $||w_j||_1 = \sum_{j=1}^n |w_j|$. In order to formulate such a minimization task in terms of a LP problem (i.e. to eliminate the absolute value $|w_j|$ from the objective function), w_j is expressed by non-negative variables α_j and β_j as $w_j = \alpha_j - \beta_j$. The minimization problem becomes thereby:

$$\begin{aligned} &\text{Minimize} && \textstyle\sum_{i=1}^n (\alpha_i + \beta_i) + C\, \sum_{i=1}^n \xi_i \\ &\text{s.t.} && y_i\, f(D(\boldsymbol{x}_i, R)) \geq 1 - \xi_i, \qquad i = 1, \ldots, n \\ &&& \alpha_i,\, \beta_i,\, \xi_i \geq 0 \end{aligned} \tag{25}$$

A more flexible formulation of a classification problem has been proposed by (Graepel et al., 1999b). Now, the problem is to minimize $||\boldsymbol{w}||_1 - \mu\, \rho$, which basically means that the margin

ρ becomes a variable of the optimization problem. Note that $\rho = 1$ for the formulation (25). By imposing $||\boldsymbol{w}||_1$ to be constant, the modified version of (25) can be introduced as:

$$\begin{aligned} \text{Minimize} \quad & \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \xi_i - \mu\,\rho \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{n} (\alpha_i + \beta_i) = 1 \\ & y_i\, f(D(\boldsymbol{x}_i, R)) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i,\, \alpha_i,\, \beta_i,\, \rho \geq 0 \end{aligned} \tag{26}$$

In this approach, a sparse solution $\boldsymbol{w}$ is obtained, which means that important objects are selected (by nonzero weights) from the original representation set R ($R = T$), resulting in a reduced set R'. This solution is a similar adaptation of the SVC for feature representations defined with the LP machines (Smola et al., 1999, Schölkopf et al., 2000). It is of essential importance since for novel objects only dissimilarities to the objects from R' have to be computed.

5.3 Support vector classifier

The support vector classifier can be built on the dissimilarity kernel. Recall, that for a chosen representation set $R = \{p_1, p_2, \ldots, p_n\}$, a dissimilarity mapping $D(z, R) : \mathcal{F} \to \mathcal{R}^n$ is defined as $D(z, R) = [d(z, p_1)\ d(z, p_2)\ \ldots\ d(z, p_n)]^T$. Since the linear decision function in such a space is given by (22), the support vector kernel K consists of the elements:

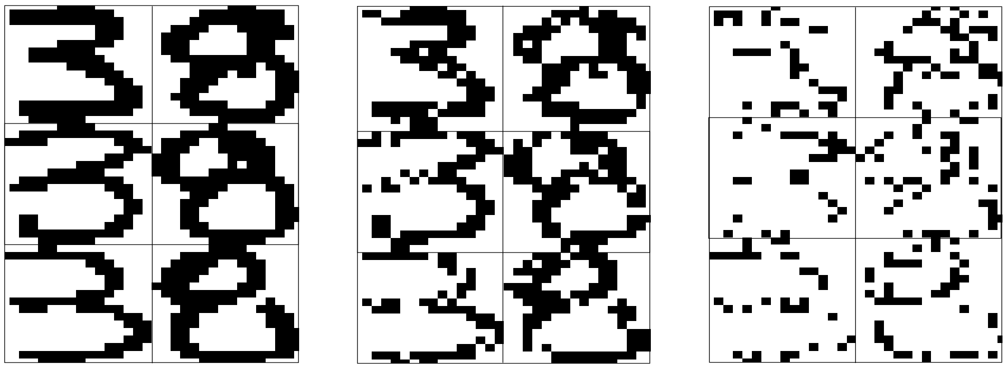
$$K_{ij} = \langle D(\boldsymbol{x}_i, R), D(\boldsymbol{x}_j, R)\rangle,$$

where $\langle \cdot, \cdot \rangle$ stands here for the Euclidean inner product. Therefore, in the formulation of the linear support vector classifier (see Section 4.3), the matrix K is given by $K = D\,D^T$, which is positive definite and can be used for the construction of the SVC. In such a case, however, a sparse solution, provided by the method, is obtained in the whole dissimilarity space $D(\cdot, R)$. It means that for evaluation of new objects, still the dissimilarities to all training objects should be computed, because our SVC is in the form of (22).

6. Experiments with the NIST digits

All experiments in this section are based on a 2-class problem, i.e. the recognition of digits 3 and 8 from a NIST database (Wilson and Garris, 1992). In total, the data consist of 2000, equally sampled, 128×128 binary images. Since no features are given to describe the images, they are represented by dissimilarity kernels.

The aim of this paper is to illustrate the potentials of dissimilarity kernels as well as studying the behavior of simple classifiers constructed by using such kernels. Therefore, we are not going to use perfect distance kernels (constructed for the specific problem such that the 1-NN gives a perfect result). Instead, our goal is to investigate what can be done in other cases. Of course, the ultimate goal is to build well-discriminating dissimilarities, however, it might be not possible as it is not always possible to find well-discriminating features for traditional classification tasks. The first series of experiments focuses on the comparison between two distance kernel approaches to classification: via the linear embedding or via dissimilarity spaces.



(a) No degradation; $P = 0$ (b) $P = 0.2$ (c) $P = 0.6$

Figure 6: Degradation of images of handwritten digits 3 and 8. Subplot a shows examples of 16×16 binary digits used in our experiments. Images of degraded digits are presented in subplots b and c. The level of degradation is governed by the probability P that an individual pixel is set to background.

Two different dissimilarity measures are considered here: Euclidean on blurred images and modified-Hausdorff (Dubuisson and Jain, 1994) on digit contours. They were chosen to illustrate the behavior of our kernel approaches with respect to the distance properties. While the first distance is a metric, the second one is not, as it violates the triangle inequality.

The modified-Hausdorff distance, applied on digit contours, is used here, since it is found useful for template matching purposes (Dubuisson and Jain, 1994). It measures the difference between two sets $A = \{a_1, \ldots, a_g\}$ and $B = \{b_1, \ldots, b_h\}$ and is defined as:

$$D_{MH}(A,B) = \max\{h_M(A,B), h_M(B,A)\} \quad \text{and} \quad h_M(A,B) = \frac{1}{g}\sum_{a\in A} \min_{b\in B} ||a-b||.$$

To calculate such a distance between two images, first the digits are detected and then the dissimilarity is computed with respect to the boxes surrounding them, which means that it is shift-invariant.

To find the second dissimilarity measure, images are first blurred with the Gaussian function with the standard deviation of 8 pixels (which is similar to the distance transform of an image). The motivation for such preprocessing is to avoid sharp edges of the digits. We, thus, make our technique robust to small tilts or variable thickness. Then, the Euclidean distance is computed between the blurred versions.

Experiments are performed 20 times and the results are averaged. In each single experiment, the data is randomly split into a training set and a testing set. The testing set consists always of 1000 samples (i.e. 500 per class). A number of different training sets are chosen, with the sizes varying from 10 to 250 objects per class. Our goal is to investigate three different directions: the behavior of the (generalized) nearest mean classifier, the behavior of the Fisher linear discriminant (Fukunaga, 1990) and the use of representation sets for classifiers constructed directly on the dissimilarity kernels.

Table 1: Dissimilarity coefficients for the binary images r and s.

Similarity measure	Metric	Euclidean	Similarity	Dissimilarity
Jaccard	Yes	Yes	$S_{rs} = \dfrac{a}{a+b+c}$	$D_{rs} = \sqrt{1 - S_{rs}}$
Simple matching	Yes	No	$S_{rs} = \dfrac{a+d}{a+b+c+d}$	$D_{rs} = 1 - S_{rs}$
Yule	No	No	$S_{rs} = \dfrac{ad-bc}{ad+bc}$	$D_{rs} = 1 - S_{rs}$

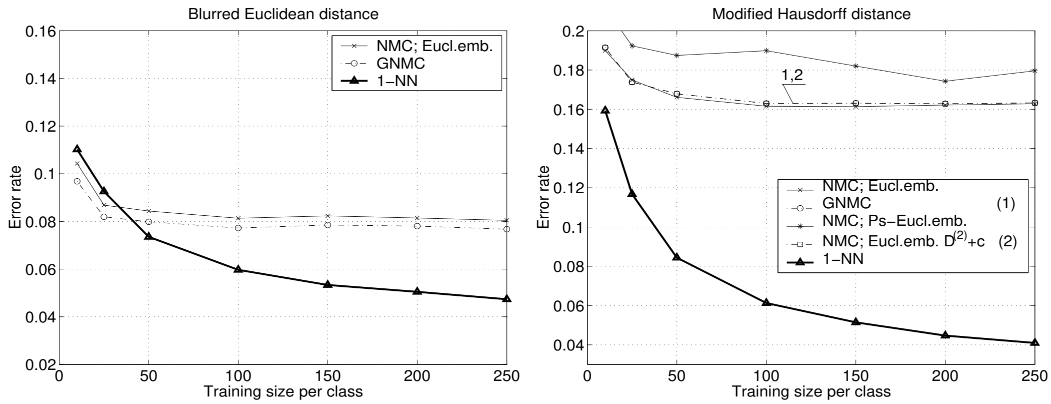
In the second type of experiments, our goal is to study the usefulness of dissimilarity kernels for data with missing values. We think that dissimilarity kernels are designed for tackling such types of problems. In order to study the performance of classifiers as a function of the number of missing values, we simulated such data by randomly corrupting the images of 3 and 8. The level of degradation is governed by a probability P that a particular image pixel is set to the background. We used four different degradation levels in our experiments, i.e. $P = \{0.0, 0.2, 0.4, 0.6\}$. Although this type of a procedure can be seen as introducing extra noise, it still simulates the missing values problem, since we assumed that the corrupted pixels had originally unknown values, and because the images are binary, the values can be just assigned to the background pixels. To simplify our experiment (to make the computation of distances less expensive), we used a resampled dataset, for which the original binary images were rescaled to a 16×16 raster; see Figure 6 for an illustration.

		object s	
		1	0
object r	1	a	b
	0	c	d

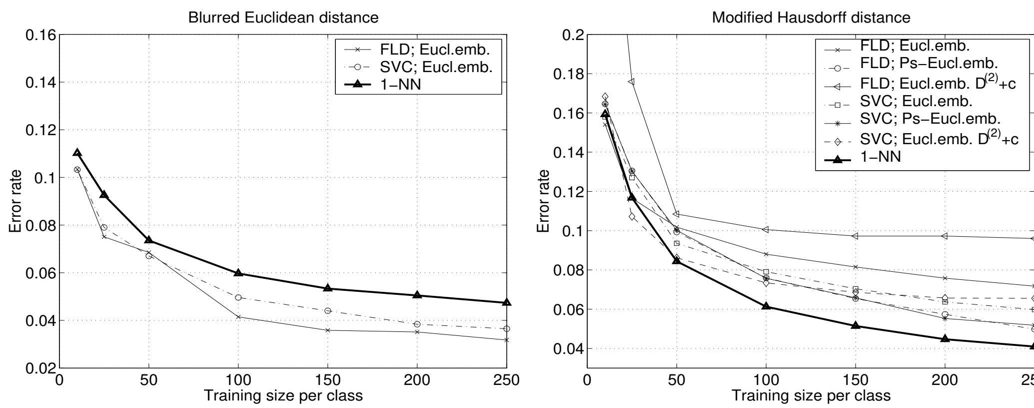
Figure 7: Similarity for binary images.

The usual way to compute dissimilarities on the binary data is to construct a similarity measure first, and then, transform it to a corresponding distance. Similarity measures for binary objects s and r are often based on variables a, b, c and d reflecting the number of elementary matches between objects (see Figure 7). For instance, the variable a reflects the number of cases where both objects scored 1. For our experiments, three measures were chosen, yielding different properties: Jaccard, simple matching and Yule similarities (Cox and Cox, 1995, Gower, 1986); see summary in Table 1. Jaccard measure is of interest, because it is an overlap ratio excluding all non-occurrences, and, thereby, disregarding the information on matches between background pixels. On the contrary, the simple matching measure, describing the proportion of matches to the total number of instances (pixels), might not be useful in our case. This comes from the fact that it counts matches between the background pixels, where some of them are considered as unknown. The Yule similarity is of different type, i.e. a cross-product ratio, measuring association between pixels as a predictability of one, given another.

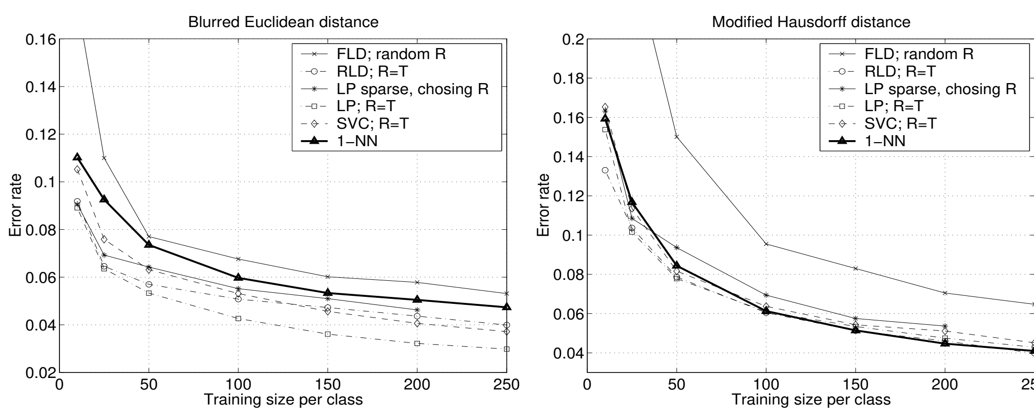
Our aim is to compare the behavior of the above presented methods on these dissimilarities with respect to different degradation. For each level of degradation, complete 2000×2000 distance matrices were computed. We assume that both, the training and the testing data, are degraded in a similar way. A training set of a fixed size of 250 samples per class was randomly chosen. All the classifiers are tested on the independent testing set containing 500 samples per class. The testing procedure is repeated again 20 times and the results are averaged.



(a) Nearest mean classifiers constructed in the embedded spaces



(b) The FLD and the SVC functions built in the embedded spaces



(c) Classifiers built on dissimilarity kernels, using the notion of the representation set

Figure 8: Comparison of different classification methods with blurred Euclidean (left column) and modified-Hausdorff (right column) dissimilarity kernels.

6.1 Results and discussion on kernel approaches to dissimilarities

For both dissimilarity kernels, modified-Hausdorff and blurred Euclidean, two experimental directions are considered. In the first direction, linear classifiers are built in the embedded space, while in the second one, linear classifiers were constructed in dissimilarity spaces, i.e. built directly on dissimilarity kernels. The results are presented with reference to the 1-NN rule, as the one, commonly applied to dissimilarity representations (for both distance kernels, the 1-NN rule is mostly the best among k-NN rules for larger k).

Concerning embedded spaces, three decision functions were used: the nearest mean classifier, the Fisher linear discriminant and the support vector classifier. They are applied in an Euclidean space, based on *only* positive eigenvalues, in a pseudo-Euclidean space and in an Euclidean space, obtained from the embedding of the 'corrected' dissimilarity kernel; see Formula (5). Note that the last classifier makes only sense for a pseudo-Euclidean space. For Euclidean dissimilarity measures, Euclidean and pseudo-Euclidean spaces coincide. Because, in such a case, both embeddings are the same, results in the pseudo-Euclidean case are not reported.

Figure 8a presents the performance of all nearest mean classifiers with reference to the 1-NN rule. For both distance measures, it can be observed that such classifiers are not complex enough for this problem to give a good performance. They perform much worse than the 1-NN rule, especially, in the case of the modified-Hausdorff kernel. This suggests that for this dissimilarity, the classes revealed in an embedded space are not of a Gaussian shape. Concerning the pseudo-Euclidean space, the GNMC reaches a higher accuracy than the NMC, however, the NMC built in an Euclidean space, based on only positive eigenvalues behaves nearly the same as the GNMC. This indicates that the 'negative' directions in a pseudo-Euclidean space are not of much significance. An interesting observation is also that the nearest mean classifiers do not nearly improve their accuracy with the increase of training size larger than 50 objects per class. This indicates that this number of objects represents well the classes in the underlying feature space.

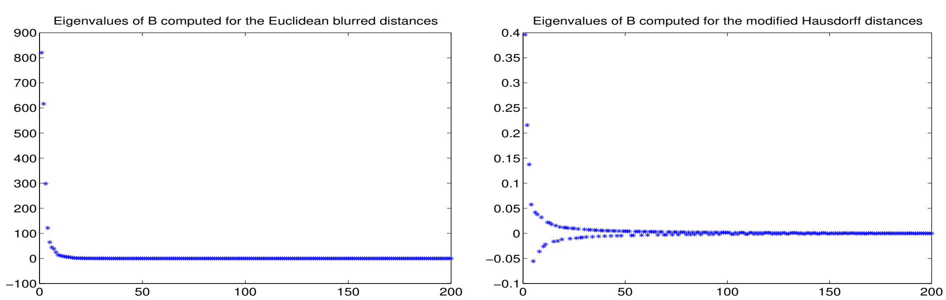


Figure 9: The eigenvalues of the matrix B of inner products for the blurred Euclidean (left) and modified-Hausdorff (right) dissimilarity kernels of the size 200×200.

Next, we studied the behavior of the FLD and the SVC constructed in the embedded spaces. Here, the FLD and SVC are built on the reduced representations, as described in Section 3.4. Such a reduced representation can be achieved as the solution of the PCA projection. The dimensionality was assigned to a value of 10% or 25% of the training size for the blurred Euclidean and the modified-Hausdorff distances, respectively. Such choices were motivated by visual exploration of the eigenvalues of the inner product matrix B and selecting the number of values significantly large in magnitude (see Formulas (1)–(3)) for the embedding purpose and compare Figure 9. The results are presented in Figure 8b, for which the following conclusions can be drawn:

- For the blurred Euclidean dissimilarity kernel, both the SVC and the FLD, constructed in the embedded space, outperform the 1-NN rule. The good performance, especially of the FLD, indicates a Gaussian description of the classes (although overlapping) as revealed by original distances.
- For the modified-Hausdorff dissimilarity kernel, a linear classifier in the embedded space seems not to be complex enough, in a similar way as the NMC. The 1-NN method mostly outperforms all other decision rules considered. Note also that for the training size larger than 100 objects, the 1-NN rule here gives lower errors than the 1-NN rule on blurred Euclidean kernel.

Concerning the behavior of different variants of the FLD and the SVC for the modified-Hausdorff distances, the pseudo-Euclidean embedding allows for reaching a higher accuracy than either an Euclidean part described only by the positive eigenvalues of B (see Formulas 1–3) or an Euclidean embedding of the enlarged dissimilarity kernel (see Formula 5). This still shows that using the 'negative' directions makes sense. For the embedding into Euclidean spaces, the FLD performs worse than the SVC, but in a pseudo-Euclidean space, they behave similarly.

In these experiments, the results depended on the training size while the dimensionality was fixed. An additional experiment was also performed, where the training size was fixed to 100 objects per class, and the constructed dimensionality was varied. As before, the test size was set to 500 samples per class. The goal is to illustrate the performance of the FLD as a function of retrieved dimensionality. Analyzing Figure 10, the following observations can be made:

- For the blurred Euclidean dissimilarity kernel, the FLD in an embedded (Euclidean) space outperforms significantly the FLD built on the dissimilarity kernel D for dimensionalities smaller than 50. The best result is reached for the dimensionality in the range $20 - 30$. For larger dimensionalities, the error increases. Judging from Figure 9, left, we observe that there are at most 20 essential eigenvalues, since the remaining ones seem not to have a significant contribution for the FLD in the embedded space. The classifier behavior for those two error curves indicates that the classes are again linearly separable.
- For the modified-Hausdorff dissimilarity kernel, the FLD in an embedded pseudo-Euclidean space performs mostly better than the FLD in an embedded Euclidean space. Here, however, the FLD on distance kernel D reaches the highest accuracy. This confirms that the classes can be separated best in a nonlinear way, since a linear classifier built on the dissimilarity kernel can be interpreted as a nonlinear decision function in an underlying feature space.

In the second direction of our experiments, we focussed on building linear classifiers in the dissimilarity space (i.e. directly on the kernel D) and on the importance of the representation set R. Until now, R was identical to the training set T. In such a case,

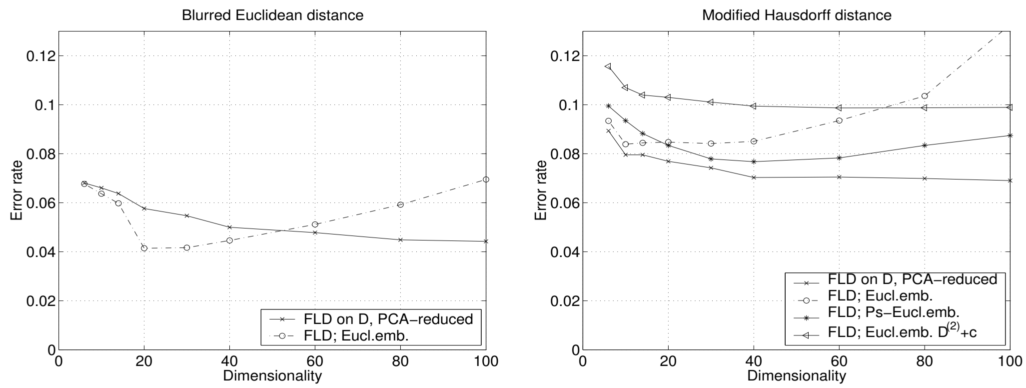


Figure 10: The performance of the FLD for blurred Euclidean (left) and modified-Hausdorff (right) dissimilarity kernels as a function of retrieved dimensionality for the fixed training size of 100 objects per class.

distances to all representation (training) objects have to be computed for an evaluation of new objects. Therefore, the reduction of the size of R is essential from the computational point of view.

In Sections 5.1 and 5.2, two different approaches were proposed. The first possibility is to reduce R by selecting objects according to a pre-specified criterion. Here, for simplicity, the random selection of objects is used, since we found that this often gives reasonable results (Duin et al., 1999, Pękalska and Duin, 2001). In the experiments, the FLD is built on the kernel $D(T,R)$, where R is always randomly reduced to 25% of the training size. We also studied the RLD with the fixed regularization parameter of 0.01, constructed on the complete dissimilarity kernel $D(R,R)$.

Another possibility is to enforce a sparse solution in a dissimilarity space in a way similar to solving the feature selection problem. This can be achieved, e.g. via linear programming schema (Bradley et al., 1998) and the classification problem can be formulated in terms of the sparse LP, as given by (26). For comparison, also an LP non-sparse solution was found, as described by (24). As a reference to all methods used, the 1-NN rule is also given.

Experimental results are presented in the Figure 8c. In case of the blurred Euclidean distances, the reduced representation sets lead to lower accuracy in comparison to the classifiers based on identical T and R. The loss in accuracy is on average 1.4%, however it is gained by using only 25% of the training samples to build R for the FLD, and by $\approx 15\% - 20\%$ of the training samples in case of the LP formulation. The representation set R is chosen in a different manner for these classifiers. In the case of the FLD, R is arbitrarily chosen beforehand. For the LP, it is, on the contrary, provided as a sparse solution of the corresponding optimization task. The classification error for this method is not provided in the case of 250 training objects in Figure 8c since the minimization problem was infeasible and the solution could not be found.

Most linear classifiers reach higher accuracy than the 1-NN rule (only the FLD with a randomly chosen R shows somewhat worse performance), thereby, it confirms that building linear classifiers on dissimilarity kernels may be beneficial.

For the modified-Hausdorff distances, most linear classifiers perform worse than in case of the blurred distances. The worst results are reported for the FLD based on the reduced, randomly selected R. The 1-NN rule is the best for training sizes larger than 100 objects per class. For smaller training sizes, it is outperformed, especially by the RLD and the non-sparse LP.

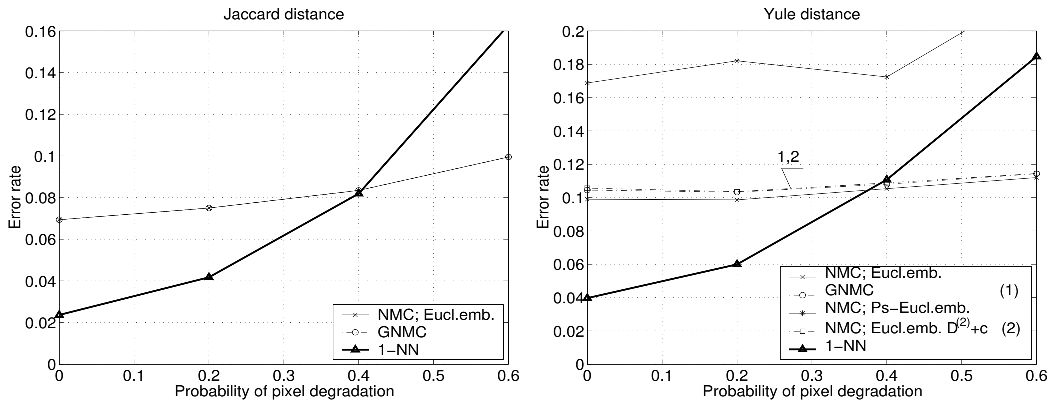
In summary, from this series of experiments, we can conclude that the blurred Euclidean dissimilarity kernel describes more compact and bounded classes than the modified-Hausdorff kernel. Also, most of the linear classifiers, both in the underlying space or on the dissimilarity kernel, perform worse in the latter case. This is probably caused by the fact, that the modified-Hausdorff distance, as applied here, does not offer rotation invariance, needed for this problem. Blurred Euclidean distance is, on the other hand, partially robust against tilting due to the initial blurring step. The 1-NN method for larger training sizes is of the best classifiers for the modified-Hausdorff case. Only the RLD and the non-sparse LP perform about the same, and for smaller training sizes, outperform the 1-NN rule.

6.2 Results and discussion on kernel approaches to dissimilarities for missing values problem

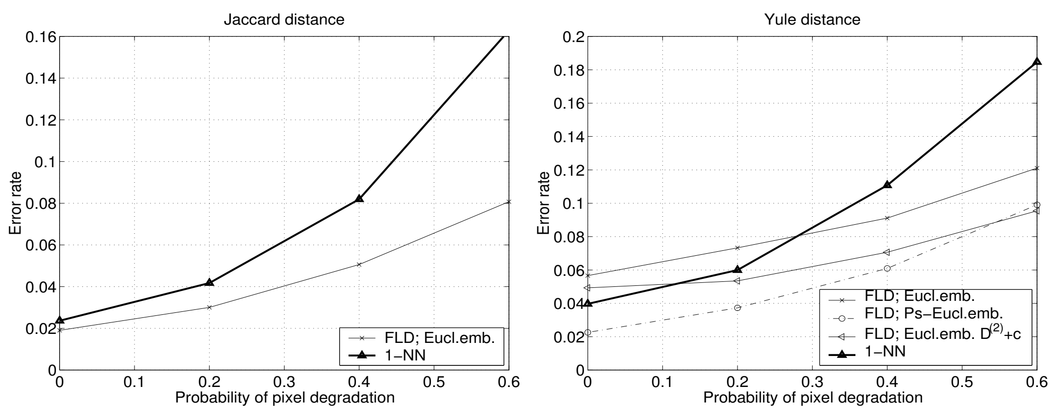
In the second track of our experiments, we studied the applicability of dissimilarity kernels to data with missing values. In order to keep a similar line of reasoning, as in the previous part, we simulated the missing values problem by setting some pixel values to the background. Both the training and testing sets have now fixed sizes and the varying quantity is the level of image degradation which was described in Section 6.

Figure 11 presents the generalization error rate as a function of increasing data degradation for Jaccard and Yule measures and three discussed groups of kernel methods: the variants of the NMC, the FLD and methods based on the representation set R. Since the results of the Yule measure and the simple matching dissimilarities are similar, we have chosen for a presentation the Yule distance kernel as the distance which is both non-Euclidean and non-metric. As an indication, the most distinct results are presented in Figure 12. The following conclusions can be made in overall:

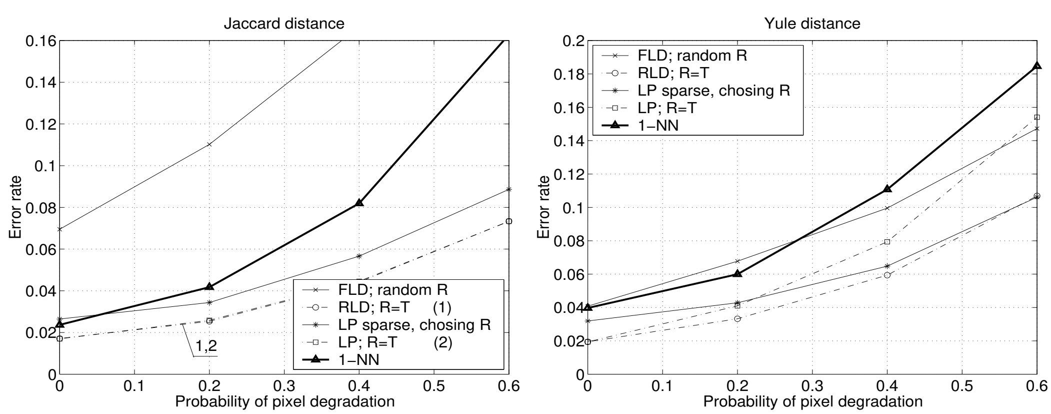
- The nearest mean classifiers are surprisingly robust against the increase of data deterioration, see Figure 11a.
- The FLD functions are less robust to data degradation, but they achieve higher accuracy than any of the NMC.
- Classifiers constructed directly on the dissimilarity kernel D, as observed in Figure 11c and 12a, generally outperform the 1-NN rule. They are also more robust against the degradation of images. Comparing all results, the 1-NN rule deteriorates the most.
- Surprisingly, the performance of classifiers does not deteriorate much for increasing image degradation. In the best cases, the error between $8\% - 10\%$ is achieved for the degradation of $P = 0.6$ (see Figure 11 and Figure 6 for reference). It suggests that simple dissimilarity kernels based on binary images are highly robust against missing information.
- Besides the NMC variants, the accuracy of classifiers on simple binary dissimilarity kernels, applied to non-degenerated data ($P = 0$), are often $1.5 - 2$ times higher than of the same classifiers on more complicated dissimilarity measures such as the blurred



(a) Nearest mean classifiers constructed in the embedded spaces

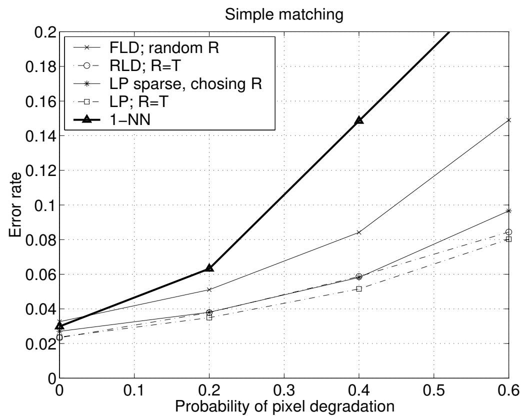


(b) The FLD built in the embedded spaces

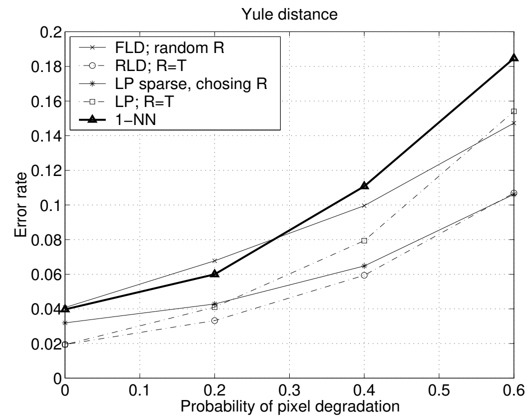


(c) Classifiers built on dissimilarity kernels, using the notion of the representation set

Figure 11: Comparison of different classification methods with the Jaccard (left column) and Yule (right column) dissimilarity kernels. The training size was fixed to 100 objects per class.



(a) Simple matching



(b) Yule distance

Figure 12: Comparison of methods using the notion of the representation set on the simple matching and Yule dissimilarities.

Euclidean and modified-Hausdorff (compare Figures 11–12 for the training size of 100 objects with Figure 8). One of the reasons is that the binary distances are applied on the rescaled images, where the digits are more 'aligned' than in case of the original 128×128 binary images.

- Surprisingly, for the Yule dissimilarity kernel, the performance of the sparse LP is better than for the non-sparse LP, when the level of degradation is higher (see Figure 11c, right).
- For the Jaccard dissimilarity kernel, both the RLD and the LP classifiers applied to the degraded images at the level $P = 0.4$ perform still comparably to the best results on the blurred Euclidean or modified Hausdorff dissimilarities.
- On average, Jaccard dissimilarity kernel allows for a better separability of classes than the Yule distance (being observed in Figure 11).
- In case of the Jaccard distance measure, two methods give identical solutions: the RLD and the LP formulation (both with $R = T$). These methods also achieve the best overall classification results compared to all other classifiers and distance measures we have investigated (error 1.7% for not degraded Jaccard distance; compare Figure 11 for $P = 0$ with Figure 8 for the training size of 100 objects).

It is interesting, that a simple distance measure (like Jaccard), operating on binary images of digits, outperforms the modified-Hausdorff dissimilarity, computed on the contours. A possible explanation is that in the latter case, the modified-Hausdorff was not implemented to be robust against tilting or stretching, while in the first case, by rescaling images to a lower raster, the digits became aligned. The binary dissimilarity measures are also considerably robust against the data degradation. For example, both the RLD and the LP classifiers, based on images with a high level of degradation, achieve the performance comparable to the best classifiers built on intact blurred Euclidean or modified-Hausdorff dissimilarities.

In summary, we have to conclude that the presented dissimilarity measures (especially the Jaccard one) are in general robust against degradation. Among all classifiers considered, the 1-NN rule shows the highest sensitivity to data degradation, which is to be expected due to its sensitivity to noisy examples. Most classifiers outperform the 1-NN rule, even the variants of the NMC for $p = 0.6$ and the FLD based on a random R for Yule and simple matching measures. This again proves our point, that often more can be achieved by constructing more complex classifiers making use of dissimilarity kernels than by the 1-NN rule. The most robust are the variants of the NMC, especially the GNMC and the NMC in an embedded Euclidean space determined by the positive eigenvalues, however, the overall performance is not the highest. The best results are achieved by the RLD and the LP.

7. Experiments with the Kimia dataset

In this experiment, we want to show how the k-NN result, based on noisy prototypes, can be outperformed by other types of more advanced classifiers constructed via embeddings or in the dissimilarity spaces. For this purpose, we use the Kimia dataset (Sebastian et al., 2001) with binary shapes. Sebastian et al. (2001) proposed to compare shapes by computing an edit-distance between their medial axes. In this type of a distance, all the efforts are put into its construction. They report that such a dissimilarity measure allows for a very good performance of the k-NN. Here, we used the same dataset but with a 'less-perfect' dissimilarity measure, i.e. the modified-Hausdorff distance. As studied here, it is not robust against rotation.

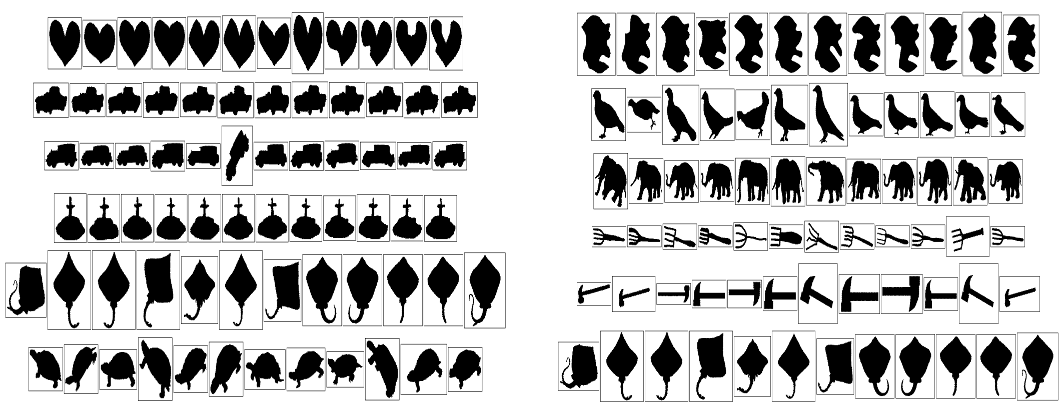


Figure 13: Group A, left, and group B, right, of the binary images, used in our experiments.

The Kimia dataset consists of 18 categories, where each category contains 12 binary images of shapes. For our experiment, to make it both illustrative and feasible, we created two groups of images, each of 6 categories, as presented in Figure 13. Notice that the images are of different sizes, which supports our idea of not starting from feature space representations. Since the classes have a small number of objects, the classification is done by using the leave-one-out procedure. It means that the dissimilarity measure is built each time on the representation set consisting of 71 objects, and one object, each time different, is used for testing. The procedure is repeated 72 times and the results are averaged.

Table 2: The leave-one-out results for two groups of binary images. The average number of support vectors or the size of the representation set R is given in parenthesis; when not given, the complete dissimilarity kernel is used. Note that a misclassification of one object gives an error of 1.39%. This means that the error of 6.94% or 12.5% stands for 5 or 9 wrongly assigned objects, respectively.

Classifier	Group A	Group B
NMC; Eucl. emb.	18.06	12.50
GNMC	16.67	15.28
NMC; ps-Eucl. emb.	20.83	16.67
NMC; Eucl. emb; $D^{(2)} + c$	27.78	12.50
FLD; Eucl. emb.	18.06	18.06
FLD; ps-Eucl. emb.	13.89	8.33
SVC; Eucl. emb.	12.50 (46)	16.67 (57)
SVC; ps-Eucl. emb.	11.11 (54)	9.72 (45)
FLD on D; R by D_{max} crit	6.94 (18)	8.33 (18)
FLD on D; random R	12.50 (32)	6.94 (18)
RLD; $R = T$	8.33	6.94
LP; sparse, choosing R	6.94 (55)	5.56 (44)
LP; $R = T$	11.11	12.50
SVC on D; $R = T$	8.33 (42)	5.56 (32)
1-NN	12.50	6.94
3-NN	13.89	11.11
5-NN	16.67	11.11
7-NN	16.67	16.67

The results of our classifiers are presented in Table 2. In case of the support vector classifiers, a number of different values for the parameters C or μ (see Formulas 16 and 26) have been considered and the best results are reported.

Group A is an example of a problem, where the k-NN error is relatively high, i.e. the smallest 1-NN error equals 12.5%, which means that 9 objects are wrongly classified. More advanced classifiers can, however, outperform the k-NN method. The best result of 6.94% is given here by the FLD constructed on the dissimilarity kernel with the representation set R of 18 objects, reduced by randomly initialized D_{max} criterion operating on a dissimilarity kernel. The criterion looks for such k (here $k = 18$) objects that the distances between them are maximized. The same performance of 6.94% is achieved by the LP classifier with the optimized R of 55 objects. Good results are also provided by the RLD and the SVC built in the dissimilarity space.

Group B is an example, where the performance of the 1-NN rule is significantly better than in the previous case, but it deteriorates much when more neighbors are taken into account. In such a case, when the 1-NN generalizes much better than the k-NN for larger k, other type of classifiers may have difficulties to outperform this method. Still, some improvement can be gained, which is reached here by the SVC and the sparse formulation of the LP classifier in the dissimilarity space. Notice, however, that many classifiers (besides the NMCs) perform better than 3-NN, 5-NN, or 7-NN, which proves our point, that they profit from a larger number of objects and are able to reduce the noise in local (of a few neighbors) neighborhoods.

For both groups, the linear classifiers built in the embedded spaces perform much worse than the linear classifiers on the dissimilarity kernel. This, however, does not exclude yet embedded spaces as being not useful in this case; it shows only that a linear classifier might be not the best solution (remember that a linear classifier on the dissimilarity kernel is a nonlinear classifier in the underlying space), while e.g. polynomial one could be.

Studying the behavior of classifiers, a few important conclusions can be made:

- The nearest mean classifiers generalize relatively poorly, which suggest that the classes are rather elongated. This can be also concluded, observing the images in Figure 13, where some classes include rotated variants of the shapes. Since our dissimilarity is not robust against rotation, those shapes differ significantly within one class.
- The FLD and the SVC perform better in a pseudo-Euclidean space than in an Euclidean space, which suggests that employing this space is profitable.
- The LP classifier with a sparse formulation generalizes well. However, since it is trained always one class against all other classes, in the end, the optimized representation set becomes large. The FLD, with the reduced representation set, gives also good results, and it is based on a much smaller number of objects, which decreases the computational load for evaluation of novel objects.

In summary, for imperfect dissimilarity measures, the k-NN method can be outperformed by more sophisticated classifiers, taking into account a number of representative objects, thus becoming more global in their decisions.

8. Conclusions

Objects are usually described by features. In this paper, an alternative approach is discussed, where objects are represented by their mutual proximities. This allows us to extend the notion of a kernel as a general proximity relation. In this paper, dissimilarity kernels are discussed, as started also by Schölkopf (2000). Such a framework is advantageous in situations, where the feature representation is not straightforward, for example when recognizing objects by their contours, for highly-dimensional data such as hyperspectral images or when dealing with missing information. Two different ways for building classifiers using dissimilarity kernels are discussed. In the first approach, dissimilarities are isometrically embedded in a (pseudo-)Euclidean space and the classification is performed there. In the second approach, classifiers are built directly on distance kernels.

We conducted a number of experiments for both approaches. This is illustrated by a 2-class digit recognition problem, for which various dissimilarity measures were investigated, and two 6-class shape classification problems based on the modified-Hausdorff distance. The experiments include also randomly distorted digit images, simulating the problem of missing values.

If a dissimilarity kernel separates the classes well, the k-NN method is supposed to achieve a good performance. However, in this paper, we focus on imperfect dissimilarity measures. Our main point is to show that in such cases, other, more advanced classifiers will usually outperform the k-NN rule.

An example when the performance of the 1-NN rule may be improved by other techniques is the blurred Euclidean distance computed on the NIST digits. Nearly all presented methods offer a higher accuracy. The best classifiers are a non-sparse LP formulation on a dissimilarity kernel and the FLD in the embedded space. Two methods do not give satisfactory results. These are the nearest mean classifiers as they do not fit the structure of the data well and the FLD classifier based on randomly chosen R. We can conclude that there is still space to improve the performance of the 1-NN rule by more global approaches, when the distance measure is not sophisticated enough for the given problem.

Another example is the modified-Hausdorff distance. The measure is built specifically for template matching purposes and it separates both digit classes well. It is almost impossible to improve the performance of the 1-NN method any further by other, more advanced approaches. Only for smaller training sizes, the accuracy of the non-sparse LP and the RLD is higher. Also, the absolute performance of the 1-NN rule is better for the modified-Hausdorff kernel than for the blurred Euclidean kernel for larger training sets. It is, however, interesting that, the non-sparse LP classifier built on a blurred Euclidean kernel outperforms the 1-NN method on the modified-Hausdorff . This shows that a more advanced classifier, constructed on a simpler dissimilarity representation, may be better than the simple 1-NN rule applied on a more sophisticated measure.

This can best be illustrated by analyzing the digit recognition problem on degradated data. The performance of the 1-NN rule seriously deteriorates with the increase of data degradation. At the same time, dissimilarities, based on less information, become simpler. In such a case, more global techniques work better than the 1-NN method. The simpler the dissimilarity, the larger the potential for improvement of the global classifiers. Even nearest mean classifiers in the embedded space or the FLD based on randomly chosen R provide better results for high data degradation.

Our experiments with the Kimia dataset confirms that in cases where the 1-NN rule gives a high error, other, more complex techniques, built on dissimilarity kernels, achieve better results.

Concerning pseudo-Euclidean spaces, we confirm the conclusion of Goldfarb (1985) that reduction of dimensionality is essential to diminish the influence of noise. The use of a pseudo-Euclidean space is often more advantageous than addressing the problem in an Euclidean space, which can be observed for both the digit and the Kimia datasets.

Building classifiers in an embedded space might be preferable to constructing them on a dissimilarity kernel, when a training set, consisting of representative objects, is sufficiently large and the retrieved dimensionality is small. This assures then a good generalization for novel objects, which can be observed in case of the blurred Euclidean distances, where the classes are well linearly separable for at least 100 training objects.

Linear classifiers operating on complete dissimilarity kernels often achieve a higher accuracy than methods with the reduced representation sets. The latter offer, however, sparse solutions, advantageous from the computational point of view. They can also achieve very good results, like in the case of the Kimia dataset. Linear classifiers, like the FLD or the SVC, generalize well in dissimilarity spaces, and often provide a better performance than the 1-NN rule.

In summary, our main conclusion is that for dissimilarities, which do not separate the classes well enough, the 1-NN, or, in general, the k-NN method can be outperformed by more advanced classifiers built either in an embedded space or on distance kernels.

Acknowledgments

This work was partly supported by the Dutch Organization for Scientific Research (NWO). Thanks go to David Tax for valuable discussions and his art of 'cutting' things. The authors are also grateful for both encouragement and criticism of the anonymous reviewers.

Appendix A. On pseudo-Euclidean spaces

This appendix is meant to provide additional information and make the picture more complete with respect to the embedding problem and pseudo-Euclidean spaces. Most of the derivations presented here are based on the following references (Borg and Groenen, 1997, Greub, 1975, Goldfarb, 1984, 1985, Gower, 1986, 1982).

A.1 A pseudo-Euclidean space

A pseudo-Euclidean space $\mathcal{E}$ is a real linear vector space equipped with a non-degenerate, indefinite, symmetric bilinear function $\langle\cdot,\cdot\rangle$, called inner product (Greub, 1975). Such a space can be interpreted as composed from two Euclidean subspaces, i.e. $\mathcal{E}_+$ of the dimensionality p and $\mathcal{E}_-$ of the dimensionality q such that $\mathcal{E} = \mathcal{E}_+ \oplus \mathcal{E}_-$ $(\mathcal{E}_+ \cap \mathcal{E}_-) = \{\mathbf{0}\}$ and the inner product is positive definite on $\mathcal{E}_+$ and negative definite on $\mathcal{E}_-$. $\mathcal{E}$ is therefore characterized by the signature (p,q) (Goldfarb, 1984). A basis $\{\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{p+q}\}$ is called orthonormal in a pseudo-Euclidean space if

$$\langle \boldsymbol{e}_i, \boldsymbol{e}_i\rangle = \begin{cases} +1, & i = 1,\ldots,p \\ -1, & i = p+1,\ldots,q \end{cases}$$

and $\langle \boldsymbol{e}_i, \boldsymbol{e}_j\rangle = 0$ for $i \neq j$. It is known that such a basis can always be constructed (Greub, 1975). Therefore, if the orthonormal basis is chosen, the inner product between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ is given by:

$$\langle \boldsymbol{x}, \boldsymbol{y}\rangle = \sum_{i=1}^{p} x_i y_i - \sum_{j=p+1}^{p+q} x_j y_j = \boldsymbol{x}^T M\, \boldsymbol{y}, \quad M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q}\end{bmatrix} \tag{27}$$

Note that the 'norm' of a non-zero vector $\boldsymbol{x}$ in a pseudo-Euclidean space is defined as:

$$||\boldsymbol{x}||^2 = \langle \boldsymbol{x}, \boldsymbol{x}\rangle = \boldsymbol{x}^T M\, \boldsymbol{x},$$

which can be positive, negative or zero. In the latter case, a non-zero vector $\boldsymbol{x}$ is called a light vector (Greub, 1975). The squared distance can be then defined in a similar way as in an Euclidean space using the notion of inner product:

$$d^2(\boldsymbol{x}, \boldsymbol{y}) = ||\boldsymbol{x}-\boldsymbol{y}||^2 = \langle \boldsymbol{x}-\boldsymbol{y}, \boldsymbol{x}-\boldsymbol{y}\rangle = (\boldsymbol{x}-\boldsymbol{y})^T M\, (\boldsymbol{x}-\boldsymbol{y}) \tag{28}$$

which can be, in general, positive, negative or zero. Note that, even when $\boldsymbol{x}$ and $\boldsymbol{y}$ are different, their distance might still equal to zero.

Alternatively, a pseudo-Euclidean space can be expressed as: $\mathcal{E} = \mathcal{R}^{(p,q)} = \mathcal{R}^p \times i\,\mathcal{R}^q$, where $i = \sqrt{-1}$, which justifies again Formulas (27) and (28) and allows one to express the square distance as:

$$d^2(\boldsymbol{x},\boldsymbol{y}) = d^2_{\mathcal{R}^p}(\boldsymbol{x},\boldsymbol{y}) - d^2_{\mathcal{R}^q}(\boldsymbol{x},\boldsymbol{y}),$$

where the distances on the right side are Euclidean. Note also, that an Euclidean space $\mathcal{R}^p = \mathcal{R}^{(p,0)}$, is a special case of the pseudo-Euclidean space.

A.2 Relation between distances and inner products

Assume n vectors given in an Euclidean space: $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$. Based on the definition of the Euclidean distance and the properties of the inner product (Greub, 1975), one may write:

$$\begin{aligned} d^2(\boldsymbol{x}_i,\boldsymbol{x}_j) &= \langle \boldsymbol{x}_i - \boldsymbol{x}_j, \boldsymbol{x}_i - \boldsymbol{x}_j \rangle \\ &= \langle \boldsymbol{x}_i, \boldsymbol{x}_i \rangle + \langle \boldsymbol{x}_j, \boldsymbol{x}_j \rangle - 2\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \\ &= d^2(\boldsymbol{x}_i, \mathbf{0}) + d^2(\boldsymbol{x}_j, \mathbf{0}) - 2\,\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle, \end{aligned} \tag{29}$$

where $\mathbf{0}$ is the origin in this space. Based on the above dependence, the inner product $\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle$ can be then expressed as:

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = -\frac{1}{2}\left[d^2(\boldsymbol{x}_i,\boldsymbol{x}_j) - d^2(\boldsymbol{x}_i,\mathbf{0}) - d^2(\boldsymbol{x}_j,\mathbf{0})\right]. \tag{30}$$

Making use of well known properties of the inner products and Formula (30), the distance $d^2(\boldsymbol{x}_i, \overline{\boldsymbol{x}})$ can be expressed only in terms of distances as follows (Goldfarb, 1985):

$$\begin{aligned} d^2(\boldsymbol{x}_i,\overline{\boldsymbol{x}}) &= ||\boldsymbol{x}_i - \overline{\boldsymbol{x}}||^2 = \langle \boldsymbol{x}_i - \overline{\boldsymbol{x}},\, \boldsymbol{x}_i - \overline{\boldsymbol{x}} \rangle \\ &= \langle \boldsymbol{x}_i,\, \boldsymbol{x}_i \rangle + \langle \overline{\boldsymbol{x}},\, \overline{\boldsymbol{x}} \rangle - 2\,\langle \boldsymbol{x}_i,\, \overline{\boldsymbol{x}} \rangle \\ &= d^2(\boldsymbol{x}_i,\, \mathbf{0}) + \tfrac{1}{n^2}\, \textstyle\sum_{p=1}^n \sum_{s=1}^n \langle \boldsymbol{x}_p,\, \boldsymbol{x}_s \rangle - \tfrac{2}{n}\, \sum_{s=1}^n \langle \boldsymbol{x}_i,\, \boldsymbol{x}_s \rangle \\ &= d^2(\boldsymbol{x}_i,\, \mathbf{0}) + \tfrac{1}{2n^2}\, \textstyle\sum_{p,s=1}^n \left[d^2(\boldsymbol{x}_p, \mathbf{0}) + d^2(\boldsymbol{x}_s, \mathbf{0}) - d^2(\boldsymbol{x}_p,\, \boldsymbol{x}_s)\right] \\ &\quad -\tfrac{1}{n} \textstyle\sum_{s=1}^n \left[d^2(\boldsymbol{x}_i,\, \mathbf{0}) + d^2(\boldsymbol{x}_s,\, \mathbf{0}) - d^2(\boldsymbol{x}_i,\, \boldsymbol{x}_s)\right] \\ &= \tfrac{1}{n}\, \textstyle\sum_{s=1}^n d^2(\boldsymbol{x}_i,\, \boldsymbol{x}_s) - \tfrac{1}{2n^2}\, \sum_{p,s=1}^n d^2(\boldsymbol{x}_p,\, \boldsymbol{x}_s) \\ &= d^2_{i\,\cdot} - \tfrac{1}{2} d^2_{\cdot\cdot}, \end{aligned} \tag{31}$$

where $d^2_{i\,\cdot}$ stands for the mean computed on each row of the kernel $D^{(2)}$ and $d^2_{\cdot\cdot}$ is the overall mean value.

Without loss of generality, let us further assume that the mean vector coincides with the origin, i.e. $\overline{\boldsymbol{x}} = \mathbf{0}$. It implies that $d^2(\boldsymbol{x}_i, \mathbf{0}) = d^2(\boldsymbol{x}_i, \overline{\boldsymbol{x}})$, which can, therefore, be used in Formula (30) and by plugging (31), one obtains:

$$\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle = -\frac{1}{2}\left[d^2(\boldsymbol{x}_i,\boldsymbol{x}_j) - \frac{1}{n}\sum_{s=1}^n d^2(\boldsymbol{x}_i,\boldsymbol{x}_s) - \frac{1}{n}\sum_{s=1}^n d^2(\boldsymbol{x}_s,\boldsymbol{x}_j) + \frac{1}{n^2}\sum_{p,s=1}^n d^2(\boldsymbol{x}_p,\boldsymbol{x}_s)\right] \tag{32}$$

Let $X \in \mathcal{R}^{n\times k}$ be a representation of all vectors and let B be the matrix of inner products, i.e. $B = X\,X^T$, such that $b_{ij} = \langle \boldsymbol{x}_i, \boldsymbol{x}_j\rangle$. Then, Formula (32) becomes:

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

Let $D^{(2)}$ be an $n \times n$ square Euclidean distance matrix. Substituting $d^2_{i\,.} = \frac{1}{n}\,D^{(2)}(\boldsymbol{x}_i,\cdot)\,\mathbf{1}^T$, $d^2_{.\,j} = \frac{1}{n}\,D^{(2)}(\cdot,\boldsymbol{x}_j)^T\,\mathbf{1}^T$ and $d^2_{..} = \frac{1}{n^2}\,\mathbf{1}\,D^{(2)}\,\mathbf{1}^T$, and after some mathematical operations, B is found to be:

$$B = -\frac{1}{2} J\,D^{(2)}\,J, \quad \text{where} \quad J = I - \frac{1}{n}\,\mathbf{1}^T\mathbf{1} \tag{33}$$

Using Formula (29), alternatively, $D^{(2)}$ can be expressed as:

$$D^{(2)} = \boldsymbol{b}\,\mathbf{1}^T + \mathbf{1}\,\boldsymbol{b}^T - 2\,B, \tag{34}$$

where $\boldsymbol{b}$ is a vector of the diagonal elements of the matrix B.

Note that precisely the same reasoning follows for a pseudo-Euclidean space $\mathcal{R}^{(p,q)}$, since the relation between square distances and inner products in both spaces is the same. Therefore, Formula (32) holds for a pseudo-Euclidean space with $\langle\cdot,\cdot\rangle$ being the inner product defined by (27). As a consequence, B, the matrix of inner products, should now be presented with respect to the pseudo-Euclidean space, i.e.:

$$B = X \begin{bmatrix} M & \\ & 0 \end{bmatrix} X^T.$$

This leads to the fact that Formulas (33) and (34) remain true for a pseudo-Euclidean space, as well.

A.3 Adding new points to an embedded space

Let $X \in \mathcal{R}^{n\times k}$ be a configuration found in a (pseudo-)Euclidean space (for a pseudo-Euclidean space $k = p+q$) and $D^{(2)}_n \in \mathcal{R}^{s\times n}$ be the square distance matrix between s new objects $v_1, v_2, \ldots, v_s$ and n objects of the representation set R. Let $\boldsymbol{y}_i$ be the vector representation of a novel object projected onto space $\mathcal{R}^k$. It follows from Formula (30) that the inner product between a new vector and the original points is given by:

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

Making use of Formula (31) and the fact that the centroid coincides with the origin, the inner product becomes:

$$\begin{aligned} \langle \boldsymbol{y}_i, \boldsymbol{x}_j\rangle \;&=\; -\tfrac{1}{2}\left[d^2_n(\boldsymbol{y}_i,\boldsymbol{x}_j) - \tfrac{1}{n}\textstyle\sum_{s=1}^n d^2_n(\boldsymbol{y}_i,\boldsymbol{x}_s) - \tfrac{1}{n}\sum_{s=1}^n d^2(\boldsymbol{x}_s,\boldsymbol{x}_j) + \tfrac{1}{n^2}\sum_{p,s=1}^n d^2(\boldsymbol{x}_p,\boldsymbol{x}_s)\right] \\ &=\; -\tfrac{1}{2}\,(d^2_n(\boldsymbol{y}_i,\boldsymbol{x}_j) - (d^2_n)_{i\,.} - d^2_{.\,j} + d^2_{..}\,), \end{aligned} \tag{35}$$

where $(d^2_n)_{i\,.}$ stands for the mean computed on each row of the dissimilarity matrix $D^{(2)}_n$. Let $B_n \in \mathcal{R}^{s\times n}$ be the matrix of inner products between s new vectors and n original ones. Using elementary matrix operations, (35) can be rewritten as:

$$B_n = -\frac{1}{2}(D^{(2)}_n\,J - U\,D^{(2)}\,J),$$

where $J = I - \frac{1}{n}\mathbf{1}^T\mathbf{1}$ and $U = \frac{1}{n}\mathbf{1}^T\mathbf{1} \in \mathcal{R}^{s\times n}$. The configuration X_n is found as a solution to $X_n\, M\, X^T = B_n$, yielding

$$X_n = B_n\, X|\Lambda|^{-1}M.$$

A.4 Generalized variability

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

$$V_d(X) = \frac{1}{n}\sum_{j=1}^{n} ||\boldsymbol{x}_j||^2 - ||\overline{\boldsymbol{x}}||^2 \tag{36}$$

Note that $||.||$ is a norm defined in a space $\mathcal{E}$. Since X reflects the same geometry as imposed by the distance matrix $D^{(2)}(R,R)$, based on the representation set $R = \{p_1, p_2, \ldots, p_n\}$, it is possible to express $V_d(X)$ only in terms of those distances as follows:

$$V_d(R) = \frac{1}{2n^2}\sum_{j=1}^{n}\sum_{k=1}^{n} d^2(p_j, p_k) \tag{37}$$

We will show now the equivalence of (37) and (36) by using equivalent transformations. Making use of Formula (34) and the facts that $\mathbf{1}^T\mathbf{1} = n$, and $\mathbf{1}^T\boldsymbol{b} = \mathrm{tr}(B) = \sum_{j=1}^n ||\boldsymbol{x}_j||^2$, one gets:

$$\begin{aligned}
V_d(R) &= \tfrac{1}{2n^2}\textstyle\sum_{j=1}^n \sum_{k=1}^n d^2(p_j,p_k) \\
&= \tfrac{1}{2n^2}\,\mathbf{1}^T D^{(2)}\,\mathbf{1} \\
&= \tfrac{1}{2n^2}\,[\mathbf{1}^T\boldsymbol{b}\,\mathbf{1}^T\mathbf{1} + \mathbf{1}^T\mathbf{1}\,\boldsymbol{b}^T\mathbf{1} - 2\,\mathbf{1}^T B\,\mathbf{1}] \\
&= \tfrac{1}{2n}\,\mathbf{1}^T\boldsymbol{b} + \tfrac{1}{2n}\boldsymbol{b}^T\mathbf{1} - \tfrac{1}{n^2}\,\mathbf{1}^T B\,\mathbf{1} \\
&= \tfrac{1}{n}\,\mathbf{1}^T\boldsymbol{b} - \tfrac{1}{n^2}\mathbf{1}^T\,B\,\mathbf{1} \\
&= \tfrac{1}{n}\textstyle\sum_{j=1}^n ||\boldsymbol{x}_j||^2 - ||\overline{\boldsymbol{x}}||^2,
\end{aligned}$$

since $\mathbf{1}^T\,B\,\mathbf{1} = \mathbf{1}^T\,X\,M\,X^T\,\mathbf{1} = \overline{\boldsymbol{x}}\,M\,\overline{\boldsymbol{x}} = ||\overline{\boldsymbol{x}}||^2$, for M being the matrix of inner products in the space $\mathcal{E}$ ($M = I$ in case of an Euclidean space). Note that by following the reasoning given in (31) for an arbitrary point $\boldsymbol{z}_x$, the equivalence between (12) and the second equation of (13) for the proximity function can be proven.

References

A.G. Arkadev and E.M. Braverman. *Computers and Pattern Recognition.* Thompson, Washington, D.C., 1966.

K.P. Bennett and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–24, 1992.

I. Borg and P. Groenen. *Modern Multidimensional Scaling.* Springer-Verlag, New York, 1997.

P.S. Bradley, O.L. Mangasarian, and W.N. Street. Feature selection via mathematical programming. *INFORMS Journal on Computing*, 10:209–217, 1998.

C.J.C. Burges. Geometry and invariance in kernel based methods. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*. MIT Press, 1998.

T.F. Cox and M.A.A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1995.

M.P. Dubuisson and A.K. Jain. Modified hausdorff distance for object matching. In *12th International Conference on Pattern Recognition*, volume 1, pages 566–568, 1994.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, 2001.

R.P.W. Duin. Compactness and complexity of pattern recognition problems. In *International Symposium on Pattern Recognition 'In Memoriam Pierre Devijver'*, pages 124–128, Royal Military Academy, Brussels, 1999.

R.P.W. Duin. Classifiers in almost empty spaces. In *15th International Conference on Pattern Recognition*, volume 2, pages 1–7, Barcelona (Spain), 2000.

R.P.W. Duin and E. Pękalska. Complexity of dissimilarity based pattern classes. In *SCIA*, 2001.

R.P.W. Duin, E. Pękalska, and D. de Ridder. Relational discriminant analysis. *Pattern Recognition Letters*, 20(11-13):1175–1181, 1999.

K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Acad. Press, 1990.

L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17:575–582, 1984.

L. Goldfarb. A new approach to pattern recognition. In L.N. Kanal and A. Rosenfeld, editors, *Progress in Pattern Recognition*, volume 2, pages 241–402. Elsevier Science Publishers B.V., 1985.

J.C. Gower. Euclidean distance geometry. *Mathematical Scientist*, 7:1–14, 1982.

J.C. Gower. Metric and euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3:5–48, 1986.

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, 1999a.

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, 1999b.

W. Greub. *Linear Algebra*. Springer-Verlag, 1975.

D.W. Jacobs, D. Weinshall, and Y. Gdalyahu. Classification with non-metric distances: Image retrieval and class representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):583–600, 2000.

A.K. Jain and D. Zongker. Representation and recognition of handwritten digits using deformable templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1386–1391, 1997.

E. Pękalska and R.P.W. Duin. Automatic pattern recognition by similarity representations. *Electronic Letters*, 37(3):159–160, 2001.

B. Schölkopf. *Support vector learning*. PhD thesis, Verlag, Munich, 1997.

B. Schölkopf. The kernel trick for distances. In *Neural Information Processing Systems*, Vancouver, British Columbia, Canada, 2000.

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*, pages 327–352. MIT Press, Cambridge, MA, 1999.

B. Schölkopf, A.J. Smola, R. Williamson, and P.L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

T.B. Sebastian, P.N. Klein, and B.B. Kimia. Recognition of shapes by editing shock graphs. In *International Conference on Computer Vision*, page to appear, 2001.

A.J. Smola, T.T. Friess, and B. Schölkopf. Semiparametric support vector and linear programming machines. In M.J. Kearns, S.A. Solla, and D.A. Cohn, editors, *Advances in Neural Information Processings Systems 11*, pages 585–591, Cambridge,MA, 1999. MIT Press.

A. Tversky. Features of similarity. *Psychological Review*, 84(4):327–352, 1977.

V. Vapnik. *The Nature of Statisticsal Learning*. Springer, N.Y., 1995.

G. Wahba. Support vector machines, reproducing kernel hilbert spaces and the randomized GACV. In B. Schölkopf, C.J.C. Burges, and A.J. Smola, editors, *Advances in Kernel Methods, Support Vector Learning*, pages 69–88. MIT Press, Cambridge, MA, 1999.

C.L. Wilson and M.D. Garris. Handprinted character database 3. Technical report, National Institute of Standards and Technology, February 1992.

G. Young and A.S. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3:19–22, 1938.