Beyond Traditional Kernels: Classification in Two Dissimilarity-Based Representation Spaces

Elżbieta Pękalska and Robert P. W. Duin, *Member, IEEE*

***Abstract*—Proximity captures the degree of similarity between examples and is thereby fundamental in learning. Learning from pairwise proximity data usually relies on either kernel methods for specifically designed kernels or the nearest neighbor (NN) rule. Kernel methods are powerful, but often cannot handle arbitrary proximities without necessary corrections. The NN rule can work well in such cases, but suffers from local decisions. The aim of this paper is to provide an indispensable explanation and insights about two simple yet powerful alternatives when neither conventional kernel methods nor the NN rule can perform best. These strategies use two proximity-based representation spaces (RSs) in which accurate classifiers are trained on all training objects and demand comparisons to a small set of prototypes. They can handle all meaningful dissimilarity measures, including non-Euclidean and nonmetric ones. Practical examples illustrate that these RSs can be highly advantageous in supervised learning. Simple classifiers built there tend to outperform the NN rule. Moreover, computational complexity may be controlled. Consequently, these approaches offer an appealing alternative to learn from proximity data for which kernel methods cannot directly be applied, are too costly or impractical, while the NN rule leads to noisy results.**

***Index Terms*—Classifier design and evaluation, indefinite kernels, similarity measures, statistical learning.**

I. Introduction

PROXIMITY plays an essential role in human learning as it is believed to underpin the process of recognition. It may be seen as a natural link between observations of objects, i.e., perception, and an overall judgment of their shared commonalities, i.e., higher level knowledge [29]. In pattern recognition, a suitable object representation is needed in order to train a classifier. A good representation is crucial since it will lead to a good discrimination between similar and different examples [23]. It is thereby natural to use proximity as a basic quality to identify patterns and model group characteristics, and to derive representations via pairwise object comparisons.

Many researchers are aware of the fundamental role that proximity plays both in recognition and class description [5], [24], [27], [28], [41], [51], [71], [74]. The use of pairwise procedures in hierarchical clustering for taxonomy can be traced back at least to Sokal [70]. Classical scaling is one of the first projection methods for dissimilarity data. It originates from the works of Cayley [8], Menger [48], and Schoenberg [66]. Later, nonlinear multidimensional scaling techniques were developed, mostly for visualization purposes [4]. However, (dis)similarity data were not systematically studied in supervised learning.

Goldfarb [27] was one of the first researchers to observe the importance of dissimilarity for object representation.[1] It was however only until mid-1990s before the true potential of proximity-based learning was realized when Vapnik laid a foundation to kernel methods [75]. Kernel tools have rapidly emerged due to an elegant relation between the kernel trick and optimization/regularization in reproducing kernel Hilbert spaces (RKHSs) [78]. Although kernel methods have been successfully applied to various learning scenarios, such techniques were not directly applicable to general proximity data. So, the severe limitation of a kernel, its positive semidefinite (psd) property, has been challenged. First, the so-called "featureless approach" was introduced in [22] and followed by [51]. This was later renamed and generalized to the dissimilarity (proximity) representation [57], [61]. This is a numerical representation whose elements encode the degrees of similarity between examples and given or optimized prototypes. Such representations extend kernels to indefinite kernels [27], [35], [43], [52], dyadic kernels [33], [34], [39], or descriptions based on pairwise relations. Since proximity measures are studied in all learning frameworks, such proximity representations are universally applicable. They can be defined over sensor measurements, such as images, and also over features, strings, graphs, probability distributions, and other knowledge-based descriptions. Suitable structure-aware measures such as edit distances can be used to compare nonvectorial examples as they usually contain an inherent and identifiable structure.

In statistical learning, proximity is usually *imposed* beforehand either as the Euclidean distance between vectors or as a psd kernel. In applications, proximity measures are defined for arbitrary patterns such as strings, histograms, shapes, bags of words, probabilistic models, etc. Many proposed measures incorporate some prior information about the problem. Hence, similarity functions are often non-psd, while dissimilarity functions are either nonmetric or lack the Euclidean behavior, i.e., are not isometrically embeddable into a Euclidean space [32], [57]. These are naturally derived when objects, shapes, or sequences are aligned in a template matching process. As argued in [41], nonmetric measures are preferred in the presence of partially occluded objects, in which violation of triangle inequality is inherent to the problem of robust matching. Moreover, incorporation

Manuscript received May 16, 2007; revised October 16, 2007. Current version published October 20, 2008. This work was supported in part by the British Engineering and Physical Science Research Council (EPSRC) under Project EP/D066883/1 and in part by the Dutch Organization for Scientific Research (NWO). This paper was recommended by Associated Editor M. Last.

E. Pękalska is with the School of Computer Science, University of Manchester, Manchester M13 9PL, U.K. (e-mail: pekalska@cs.man.ac.uk).

R. P. W. Duin is with the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2628 CN Delft, The Netherlands (e-mail: r.duin@ieee.org).

Digital Object Identifier 10.1109/TSMCC.2008.2001687

[1] Later, Goldfarb developed a theory of inductive structural learning in which objects were evolving structural processes and the dissimilarity measure was dynamically learned [28]. This theory is beyond the scope of this paper.

of invariance leads to indefinite kernels, i.e., non-psd similarities [36]. Nonmetric examples include structural local alignments of proteins [62], modified-Hausdorff distances [20], measures based on deformable templates [42], tangent distance [69], normalized edit distances [6], or Kullback–Leibler divergence [25].

Although such general (dis)similarity measures are widely used for matching and object comparison [6], [20], [25], [41], [76], classification often relies on the nearest neighbor (NN) rule or its variants. The NN rule is commonly applied because it is simple and tends to perform well on large training sets. If the derivation of dissimilarities is costly, complexity needs to be reduced in the test stage, i.e., the number of objects to which dissimilarities have to be computed. The NN rule often worsens its performance for such reduced prototype sets. And this is where proximity-based linear and quadratic classifiers become profitable as they often lead to higher classification accuracy than this condensed NN rule. They perform well, irrespectively whether the measure is Euclidean, non-Euclidean, metric or nonmetric. This avoids the use of regularization when kernels are derived [35], [47] or metric correction techniques that may cause significant loss of information when the deviation from the psd, Euclidean, or metric behavior is large [43], [59]. This important fact is illustrated in our experiments.

In this paper, we provide a summary about general proximity-based learning methods, focussing on classification in two simple yet powerful proximity-based representation spaces (RSs): dissimilarity spaces and pseudo-Euclidean spaces. A fundamental explanation of these spaces is given, emphasizing their applicability to general measures. Important findings and insights are summarized based on experiments conducted on artificial and real data.

The paper is organized as follows. Section II starts with an overview of proximity-based learning paradigms. Section III provides the foundation of proximity-based RSs. Section IV focusses on techniques that determine a representation set. Sections V and VI describe experiments that illustrate the properties of the reviewed representations. The discussion and overall conclusions are in Section VII.

II. Overview of Proximity-Based Learning

Basic definitions and properties related to proximities and kernels can be found in the Appendix. We will now focus on two main and significantly different proximity-based learning scenarios. The first one relies on neighborhood relations defined via the (dis)similarity values. This leads to variants of the NN rule that are directly applied to the given proximity data. The other strategy trains classifiers in suitable RSs. Such vector spaces are determined either by linear or nonlinear projection methods, or remain implicit, but approachable via the kernel trick for kernel methods.

A. Direct NN Rule Without the Use of RSs

In statistical learning, one often starts from a predefined feature space in which objects are represented by feature vectors. In this context, proximity-based techniques include variants of (weighted) NN rule [16] or condensed NN rule, often based on the Euclidean distance [13], [15], [21], [37]. To account for variability in feature spaces, a local structure is taken into account in order to learn the metric or to weight neighbor contributions appropriately [18], [38], [45], [55]. Such approaches are designed to optimize either the parameters of the measure in local regions of the feature space or the number of NNs. We are not interested in these methods here, since we deal with the *given* proximity data, usually provided without any accompanying feature-based representation.

The direct NN rule is our baseline method. It means that the NN rule is applied to the given proximity matrix, without the use of an RS. In general, the 1NN and kNN rules are the simplest learning strategies for arbitrary proximity data. They are asymptotically optimal in the Bayes sense for Euclidean (or metric) distances [16]. They can generalize well for large training sets but at high storage and computational costs. Moreover, the accuracy of the kNN rule may significantly be affected by noisy or erroneously labeled examples. Various prototype optimization techniques exist to alleviate these drawbacks in feature vector spaces. Initial training examples are either merged [21], [40], [50] or reduced [13], [15], [37] to a small prototype set. Some of such techniques are adapted for the use of the direct NN rule.

B. Statistical Classifiers in Representation Vector Spaces

RSs are data-dependent inner product vector spaces that encode proximity information. Although finite metric data can be embedded with low distortion into normed and metric spaces such as l_1 or l_∞, their applicability for statistical learning is limited, mainly to a fast and approximate search of NNs or simple statistics on huge amounts of data; see chapters of Indyk in [30]. The reason is that statistical techniques often rely on an inner product, which is not originally defined in arbitrary normed or metric spaces. So, inner product spaces are of interest here, because they allow us to use traditional statistical classifiers such as linear ones. We now start with the definition of proximity representation.

1) Proximity Representation Versus Kernel: Assume a set of prototype objects $R = \{p_1, p_2, \ldots, p_n\}$ in any initial representation (e.g., strings, shapes, or bags of words) called a *representation set*. Let d be a nonnegative dissimilarity measure that ideally incorporates prior knowledge and the invariance of the application domain. In general, we only require that d is reflexive, i.e., $d(x, x) = 0$ for all x. An object x is represented as a vector of dissimilarities computed between x and the prototypes from R, i.e., $D(x, R) = [d(x, p_1)\, d(x, p_2)\ \cdots\ d(x, p_n)]^{\mathsf{T}}$. Given a training set $\mathcal{X} = \{x_1, \ldots, x_N\}$ of N examples, a dissimilarity representation is an $N \times n$ dissimilarity matrix $D(\mathcal{X}, R)$, in which $D(x_i, R)$ is now a row vector [57], [61]. If a similarity measure k is used instead, we will get a similarity representation $K(\mathcal{X}, R)$ defined by similarity vectors $K(x, R) = [k(x, p_1)\, k(x, p_2)\ \cdots\ k(x, p_n)]^{\mathsf{T}}$. Moreover, if $\mathcal{X} = R$ and k is psd, then K is a kernel matrix. Often $R \subseteq \mathcal{X}$, but R and $\mathcal{X}$ may also be disjoint sets. R is either given or optimized to guarantee a good tradeoff between recognition accuracy of the final classifier and computational complexity. One may therefore control the size of R.

As as result, kernel matrices form a specific class of proximity representations. A kernel is a (conditionally) psd function $K(x, y)$ of two variables x and y, interpreted as a generalized inner product (hence similarity) in an RKHS $\mathcal{H}$ induced by K [67], [68], [78]. Due to the reproducing property of K, kernel-based classifiers are indirectly built in $\mathcal{H}$ and often expressed as linear combinations of kernel values. Although the applicability of traditional kernels has been extended to general nonvectorial descriptions, e.g., [44] and [79], the class of permissible kernels is limited due to their requirement of being psd [or conditionally positive definite (cpd)]. A natural generalization leads to indefinite kernels [7], [35], [52] or dyadic kernels [39], which are also examples of proximity representations.

2) Representation Spaces: Let $D(\mathcal{X}, R)$ be an $N \times n$ dissimilarity representation with the elements $d(x_i, p_j)$. Let $K(\mathcal{X}, R)$ be the corresponding $N \times n$ similarity representation of the elements $k(x_i, p_j)$. By "corresponding," we mean the following. If the dissimilarity d is designed first, then k is defined as $k(x_i, p_j) = 1/2[d^2(x_i, 0) + d^2(0, p_j) - d^2(x_i, p_j)]$, where 0 represents a specific element that acts as a reference. If the similarity k is defined first, then d is computed such that $d^2(x_i, p_j) = k(x_i, x_i) + k(p_j, p_j) - 2k(x_i, p_j)$.

Proximity-based RSs are used either directly or indirectly. Later, we list main linear and nonlinear projection approaches; the list is nonexhaustive. *A linear projection* is based on linear operations of the given/transformed proximity representation. The main techniques are as follows.

1) *Implicit use of RSs via (indefinite) kernel matrices:*
   a) *psd kernel $K(\mathcal{X}, \mathcal{X})$ or cpd kernel $-D^{\star 2}(\mathcal{X}, \mathcal{X})$: K* can be forced to be psd by regularization [47], approximation, or transformation [11]. Note that support vector machine (SVM) is also reformulated for dyadic kernels $K(\mathcal{X}, R)$, $R \subseteq \mathcal{X}$ [39].
   b) *Any symmetric kernels $K(\mathcal{X}, \mathcal{X})$ or $-D^{\star 2}(\mathcal{X}, \mathcal{X})$:* Indefinite kernel methods rely on the reproducing property of the kernel matrix and work in the reproducing kernel Krein (pseudo-Euclidean) spaces [7], [35], [52].
2) *Explicit use of RSs via maps into inner product spaces:*
   a) *Proximity representations $K(\mathcal{X}, R)$ or $D(\mathcal{X}, R)$, $R \subseteq \mathcal{X}$, based on symmetric measures, are used to determine Euclidean or pseudo-Euclidean spaces.*
      i) Pseudo-Euclidean linear embedding. It simplifies to classical scaling, if k is psd or d has a Euclidean behavior [4], [27] (see Section III-B).
      ii) FastMap, a distance-preserving linear embedding into a Euclidean space [26]. d has to be Euclidean.
      iii) Locally linear embedding (LLE) [65]. d is the Euclidean metric or the corresponding k is psd.
      iv) Laplacian, Hessian, and other eigenmaps [1], [19]. Euclidean distances are further used in kernels.
      v) Linear map into a (Euclidean) proximity space [61] for arbitrary d and s (see Section III-C). Proximity space is related to dyadic kernels [39]. Note that relevance SVM [73] can be seen as a Bayesian approach in a similarity space.
      vi) Nonlinear multidimensional scaling or variants of Sammon mapping [4], [11]. Neural nets [14], regression or reformulated stress optimization [56] can be used to learn the map afterwards.
      vii) Embedding into a Euclidean space within a regularization framework [47].
      viii) Neural nets or nonlinear optimization techniques as general tools for nonlinear projections.
   b) *Arbitrary proximity representations $K(\mathcal{X}, R)$ or $D(\mathcal{X}, R)$ are used to determine Euclidean spaces.* These include linear maps found by singular value decomposition, proximity space, or nonlinear maps found by (auto-associative) neural nets or via optimization techniques.
3) *Explicit use of RSs via embeddings into normed or metric spaces. $D(\mathcal{X}, \mathcal{X})$ is metric.* These include Lipschitz embeddings into compact metric spaces [49], [77] or low-distortion embeddings into normed l_p-spaces, $p = 1, \infty$ (see [30] for details).

3) Classifiers in RSs: Inner product spaces are of high interest, because of the availability of various statistical techniques. It is important to emphasize that a proximity space is the simplest and most general concept for such an inner product RS. It is applicable to all proximity representations and all measures. Moreover, it makes use of the original (dis)similarities. Other simple approaches such as FastMap, LLE or Laplacian eigenmaps work only for Euclidean distances or kernels (such as Gaussian) defined over Euclidean distances. Hence, their applicability is limited.

Because we deal with finite data, kernel methods can be interpreted in both similarity spaces defined by the kernel matrix K and the RKHSs, which are Euclidean spaces induced by a finite K. So, there are close relations between these two spaces. For example, SVM is the largest margin linear classifier in the RKHS induced by the kernel matrix $K(\mathcal{X}, \mathcal{X})$, which is at the same time a linear function of the kernel values to the selected support vectors. Given a two-class problem, the analogy between the SVM and linear decision (LD) functions in proximity spaces is SVM: $f(x) = \sum_{p_i \in SV} w_i K(x, p_i) + w_0$, LD in a similarity space: $f(x) = \sum_{p_i \in R} w_i K(x, p_i) + w_0$, and LD in a dissimilarity space: $f(x) = \sum_{p_i \in R} w_i D(x, p_i) + w_0$. $R \subseteq \mathcal{X}$ is a representation set and $SV \subseteq \mathcal{X}$ denote support vectors determined by SVM. All these decision functions are linear combinations of proximity values. The basic difference is that in SVM, both the classifier and support vectors are optimized together to guarantee the largest margin in the RKHS. Linear classifiers in proximity spaces are usually defined in a two-stage process. R is determined first, and then, a linear classifier is estimated based on a chosen model or assumption, e.g., a linear discriminant. So, the representation set and weights w_i are optimized for SVM differently than for other linear classifiers in proximity spaces. SVM is simply a specific classifier in a similarity space based on an elegant mathematical interpretation in

the RKHS. If a psd kernel $K(\mathcal{X}, R)$ defines a similarity space, all classifiers built there can be interpreted in a suitable RKHS.

III. Two Important Dissimilarity-Based RSs

Assume $\mathcal{X}$ is a training set, R is an optimized representation set and U is an evaluation set. $D(\mathcal{X}, R)$ is an $N \times n$ training dissimilarity matrix and $D(U, R)$ is an $M \times n$ test dissimilarity matrix. Accordingly, $K(\mathcal{X}, R)$ is an $N \times n$ training similarity matrix, while $K(U, R)$ is an $M \times n$ test similarity matrix.

Note that the direct 1NN and kNN rules used in this context are the nearest prototype methods. Unless stated otherwise, the NN rule will act on the representation set R. When applied to $D(U, R)$, the test objects from U are assigned to classes that most frequently occur among the kNNs in R [as judged by k smallest dissimilarities $d(u, p_j)$, for each test object $u \in U$ and $p_i \in R$]. If the original vectorial representation is also available, prototype generation techniques can be employed to determine R [40], [46], [50]. Otherwise, R is selected out of the training set, e.g., by condensing methods. If classes are densely sampled, the NN rule is expected to perform well for metric distances and a very large representation set R.

A. Classification in Proximity-Based RSs

We now focuss on pseudo-Euclidean spaces and dissimilarity spaces, which provide the simplest, yet powerful ways of encoding (dis)similarity information. Although more complex procedures exist, as listed in Section II-B, they usually rely on vectorial data and Euclidean distances [19], [65], [80], and are not of general applicability. Classification in proximity-based spaces involves the following steps.

1) First, dissimilarity/similarity measure is defined for the raw data, given by an application expert, learned from the examples, or defined in a feature space.
2) Representation set R is usually chosen (or generated) from the initial set of learning objects $\mathcal{X}$. R should preferably be a fraction, such as 5%–10%, or a logarithm of the total number of objects $|\mathcal{X}|$. It may be reduced to a few objects for the execution speed (see Section IV for details).
3) Having determined R, an RS is constructed from either $D(R, R)$ or $K(R, R)$. The two main procedures are discussed in Sections III-B and C.
4) All training objects are mapped into such a constructed RS and a classifier is trained by these. Traditional statistical classifiers are used there or new ones are formulated. Linear and quadratic functions are especially of interest, because of their simplicity.
5) Finally, test objects are mapped into the RS based on their proximities to the examples from R. The resulting test vectors are classified.

Several dissimilarity measures applied in practice are non-Euclidean or nonmetric, often due to incorporation of invariance. Even if the Euclidean distance is used as the basic point-to-point distance (as, for example, in modified-Hausdorff distances [20]), the minimum distances between sets of invariant representations may conflict the triangle inequality, as illustrated in Fig. 1. It is shown in [36] that the use of kernels in relation to invariants may lead to indefinite kernels.

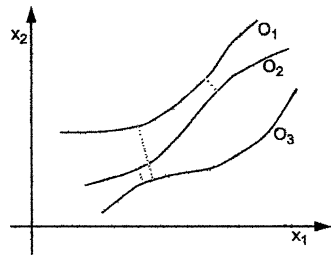


Fig. 1. Parameter vector space with the invariant trajectories for objects O_1, O_2, and O_3 representing equivalence classes. If the chosen dissimilarity measure is the minimal distance between these trajectories, triangle inequality can easily be violated, i.e., $d(O_1, O_2) + d(O_1, O_2) < d(O_1, O_3)$.

The two dissimilarity-based RSs discussed here are universal, because there is no restriction to either Euclidean or metric measures. They are thereby suitable to handle problems described earlier, in which non-Euclidean measures naturally arise. In Section III-B, we discuss the possibility of pseudo-Euclidean embedding, in which the non-Euclidean behavior is explicitly modeled. In Section III-C, we present the dissimilarity space for which the nature of the measure makes no difference. It can always be applied in the same way.

B. Pseudo-Euclidean Linear Embedding Approach

Let d be a symmetric dissimilarity measure and $R \subseteq \mathcal{X}$. The corresponding dissimilarity matrix $D := D(R, R)$ can be embedded in a pseudo-Euclidean space $\mathcal{E}$ by an isometric (distance-preserving) mapping [27], [57]. $\mathcal{E} = \mathbb{R}^{(p,q)} = \mathbb{R}^p \oplus \mathbb{R}^q$ is a real vector space equipped with a nondegenerate indefinite inner product $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ such that $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ is positive definite on $\mathbb{R}^p$ and negative definite on $\mathbb{R}^q$. $\mathcal{E}$ is therefore characterized by the so-called signature (p, q), indicating the dimensions of both subspaces. Note that $\mathbb{R}^{(p,0)}$ is a Euclidean space.

The inner product between two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{(p,q)}$ with respect to an orthonormal basis is defined as $\langle \mathbf{x}, \mathbf{y} \rangle_{\mathcal{E}} = \mathbf{x}^{\mathsf{T}} \mathcal{J}_{pq} \mathbf{y}$, where

$$\mathcal{J}_{pq} = \begin{bmatrix} I_{p \times p} & 0 \\ 0 & -I_{q \times q} \end{bmatrix}$$

and I denotes the identity matrix. Hence, $d^2_{\mathcal{E}}(\mathbf{x}, \mathbf{y}) = (\mathbf{x} - \mathbf{y})^{\mathsf{T}} \mathcal{J}_{pq} (\mathbf{x} - \mathbf{y})$, and it becomes square Euclidean for $\mathcal{J}_{pq} = I$. Since $\mathcal{E}$ is a linear space, operations based on inner products are appropriately extended from the Euclidean case. The interpretations are, however, different. More details can be found, e.g., in [3], [27], [57], and [61].

In a pseudo-Euclidean embedding, we look for a configuration X (here vectors are stored as rows) in some $\mathbb{R}^{(p,q)}$ that preserves the distances given by an $n \times n$ dissimilarity matrix $D^{\star 2} = (d^2_{ij})$. First, the indefinite Gram matrix G is computed as $G = -(1/2) J D^{\star 2} J$, where $J = I - (1/n)\mathbf{1}\mathbf{1}^{\mathsf{T}}$ is the centering matrix [27]. J projects[2] the data such that X has a zero mean vector. The factorization of G is found by its eigendecomposition as $G = Q \Lambda Q^{\mathsf{T}} = Q |\Lambda|^{1/2} [\begin{smallmatrix} \mathcal{J}_{pq} \\ 0 \end{smallmatrix}]$

[2] A more general projection sets a weighted mean of X to a zero vector. Then, $J_s = I - \mathbf{1}\,\mathbf{s}^{\mathsf{T}}$, where $\mathbf{s}$ is such that $\mathbf{s}^{\mathsf{T}}\mathbf{1} = 1$ and $G = -(1/2)\, J_s\, D^{\star 2}\, J_s^{\mathsf{T}}$. By choosing a suitable $\mathbf{s}$, any vector from X can be projected at the origin.

$|\Lambda|^{1/2}Q^{\mathsf{T}}$, where Λ is a diagonal matrix of first decreasing p positive eigenvalues, followed by increasing q negative eigenvalues, and zeros. Q is a matrix of the corresponding eigenvectors. Since G is a Gram matrix, then $G = X\mathcal{J}_{pq}X^{\mathsf{T}}$ by definition. As a result, given $r = p + q$ nonzero eigenvalues ($r < n$), an r-dimensional X is found as $X = Q_r\,|\Lambda_r|^{1/2}$, where $Q_r \in \mathbb{R}^{n\times r}$ is a matrix of r leading eigenvectors and $\Lambda_r \in \mathbb{R}^{r\times r}$ contains the corresponding eigenvalues. X is uncorrelated, because the estimated pseudo-Euclidean covariance matrix $C = [1/(n-1)]X^{\mathsf{T}}X\,\mathcal{J}_{pq} = [1/(n-1)]\,\Lambda_r$ is diagonal. Moreover, the eigenvalues λ_i encode variances in this space. This holds thanks to the centering effect of J in the definition of G and it is not valid for a general J_s (see footnote 2).

The eigenvalues of G play a key role as they scale the basis eigenvectors. Since only some eigenvalues are expected to be large in magnitude, the remaining ones, if close to zero, can be disregarded as uninformative. By their removal, the data are not only denoised, but the curse of dimensionality may be avoided. Since X is uncorrelated, the reduced representation X' is determined by the largest p' positive and the smallest q' negative eigenvalues as $X' = Q_m\,|\Lambda_m|^{1/2}$, $m = p' + q' < r$. As such, X' is derived from an approximate embedding and reflects the principal component analysis (PCA) result. More precisely, the embedding is equivalent to an indefinite kernel PCA, in which G is a reproducing kernel for the pseudo-Euclidean space $\mathbb{R}^{(p,q)}$. In general, the embedding can also start from a similarity matrix $K(R,R)$ (interpreted as a Gram matrix $G := K$), instead of D. Note that for the Euclidean distance d, the resulting embedding is known as classical scaling [4].

Let $D^{\star 2}_{\mathrm{new}}(U,R)$ be a matrix of square dissimilarities relating new objects from U to the set R. An m-dimensional X'_{new} is determined by orthogonal projections to $\mathcal{E}$. These are uniquely defined since $\mathbb{R}^{(p',q')}$ is a nondegenerate space (as dimensions corresponding to zero eigenvalues are neglected). Based on the indefinite cross-Gram matrix $G_{\mathrm{new}} = -1/2(D^{\star 2}_{\mathrm{new}} - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathsf{T}}D^{\star 2})J$, X'_{new} is derived as $X'_{\mathrm{new}} = G_{\mathrm{new}}X'|\Lambda_m|^{-1}\mathcal{J}_{p'q'}$ by the use of orthogonal projections [27], [57], [61].

Pseudo-Euclidean embedding into $\mathbb{R}^m$ relies on $D(R,R)$. All training examples $D(\mathcal{X},R)$ are projected to $\mathbb{R}^m$ and used for training. Classifiers based on inner products are appropriately extended from the Euclidean case. For example, an LD function $f(\mathbf{x}) = \mathbf{v}^{\mathsf{T}}\mathcal{J}_{pq}\mathbf{x} + v_0$ can be derived in $\mathcal{E}$ or constructed by addressing it as $f(\mathbf{x}) = \mathbf{w}^{\mathsf{T}}\mathbf{x} + v_0$, where $\mathbf{w} = \mathcal{J}_{pq}\mathbf{v}$ [27], [34], [57]. Additionally, since $\mathbb{R}^{(p',q')}$ is an inner product vector space with the induced (strong) Hilbert topology of the norm in the associated Euclidean space $\mathbb{R}^{(p'+q')}$ [64], one may choose to train classifiers in $\mathbb{R}^{(p'+q')}$.

C. Dissimilarity Space Approach

In this approach, a dissimilarity representation $D(\mathcal{X},R)$ is addressed as a data-dependent mapping $D(\cdot,R) : \mathcal{X} \to \mathbb{R}^n$ from an initial representation or the index set $\mathcal{X}$ to the so-called *dissimilarity space* [22], [34], [61], [79]. In this space, each dimension $D(\cdot,p_i)$ describes a dissimilarity to a prototype p_i. Such a vector space is equipped with the traditional inner product and Euclidean metric. Since dissimilarities are nonnegative, all data are mapped as vectors to a nonnegative orthotope. Any symmetric or asymmetric dissimilarity measure d can be used.

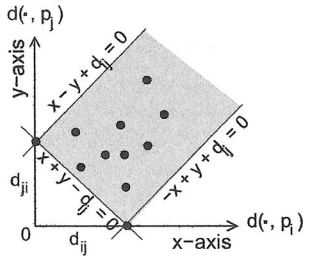


Fig. 2. Two-dimensional metric dissimilarity space. $x := d(x,p_i)$ and $y := d(x,p_j)$.

Let $R \subseteq \mathcal{X}$. If $D(R,R)$ is a nonsingular $n \times n$ matrix, then all vectors $D(p_i,R)$ are corners of an $(n-1)$-dimensional simplex (R-simplex) in the n-dimensional dissimilarity space $D(\cdot,R)$. If d is metric, then all $D(x,R)$ lie in an open n-dimensional hyperprism defined by a perpendicular move of its R-simplex base. If d is bounded, i.e., $D(x,p_i) \le c$ for $c > 0$, then all $D(x,R)$ lie in the intersection between the open hyperprism and the c-length hypercube placed with one corner at the origin in the positive orthotope of $D(\cdot,R)$. The vertices of the hyperprism's base lie in $(n-1)$-dimensional subspaces of $D(\cdot,R)$, which are the axes of $D(\cdot,R)$ if $|R| = 2$ (see Fig. 2). For a nonmetric measure, $D(x,R)$ may lie outside the hyperprism.

The property that the dissimilarity should be small for similar objects and large for distinct objects gives a possibility for discrimination. So, $D(\cdot,p_i)$ can be interpreted as a dissimilarity-based feature. If d is metric and $d(p_i,p_j)$ is small, then $d(x,p_i) \approx d(x,p_j)$ holds for all x due to the backward triangle inequality. Since p_i and p_j encode similar dissimilarity information, one of them is sufficient to be considered as a prototype. In the nonmetric case, R should be chosen such that the vectors $D(x,R)$ and $D(z,R)$ are correlated for two similar objects x and z, even if $d(x,p_i)$ and $d(z,p_i)$ differ for some $p_i \in R$. Classifiers defined by linear or quadratic combinations of $d(\cdot,p_i)$ $\forall_{p_i\in R}$ are useful for nonmetric or poorly discriminative measures (see Section III-D). For a two-class problem, an LD in $D(\cdot,R)$ is defined as $f(D(x,R)) = \sum_{i=1}^{n} w_i\,d(x,p_i) + w_0$.

D. Direct 1NN Rule Versus Simple Classifiers in Dissimilarity Spaces

We now provide some intuition on why the 1NN rule can be outperformed by classifiers built in dissimilarity spaces. Assume F classes $\omega_1,\ldots,\omega_F$, with their corresponding prototype sets $R_1,\ldots,R_F$, and the complete representation set $R = \{R_1,\ldots,R_F\}$, $|R| = n$. A measure d is perfect if all training examples share class memberships with their NNs (determined by the smallest dissimilarity, i.e., for each ω_c and each training example $x_i^c \in \omega_c$, one has $\{\min_{p^c\in\{R_c\setminus x_i^c\}} d(x_i^c,p^c) < \min_{p^{\neg c}\in\{R\setminus R_c\}} d(x_i^c,p^{\neg c})\}$. This means that the leave-one-out (LOO) 1NN error on $D(\mathcal{X},R)$ is zero. If new testing objects are perfectly classified, then the 1NN rule is a zero-error classifier.

Imperfect measures are common in practice, e.g., when derived by suboptimal procedures or defined for complex

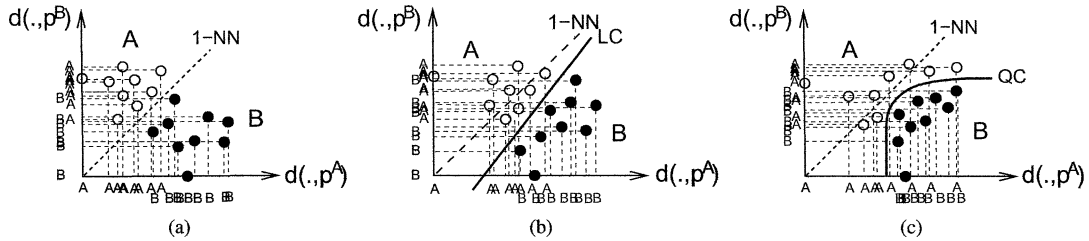


Fig. 3. Direct 1NN rule (here: nearest prototype rule) versus linear or quadratic classifiers in a 2-D dissimilarity space $D(\cdot, R)$, $R = [p^A\ p^B]$. Projections on the axes are indicated by dashed lines and classes A or B. (a) Measure is perfect, so is the 1NN rule (nearest prototype rule). (b) Measure is imperfect, but a linear classifier is (LC: linear classifier). (c) Individual prototypes are bad (see projections on the axes), but their quadratic combination is optimal. This does not hold for any linear classifier (QC: quadratic classifier).

nonvectorial data. Moreover, even perfect measures become imperfect when large prototype sets are reduced in order to lower the computational costs, especially for multimodal data. As a result, the dissimilarity measure will be badly descriptive for some classes. This means that the dissimilarities computed to nearest prototypes within such a class will be large and may be larger than the dissimilarities computed to the nearest prototypes of other classes. So, the 1NN and kNN rules will deteriorate, as a result. However, when the dissimilarity contributions are appropriately weighted to either emphasize or weaken discriminative abilities of the prototypes, their linear or quadratic combinations will lead to better decisions.

Consider a two-class problem in 2-D dissimilarity spaces, as depicted in Fig. 3. Classes A and B are represented by the prototypes p^A and p^B, respectively; $R = [p^A\ p^B]$. Ideally, if d is perfect, then $d(x, p_A) < d(x, p_B)$ holds for any $x \in A$ and $d(x, p^B) < d(x, p^A)$ holds for any $x \in B$ [see Fig. 3(a)]. So, all vectors $D(x, R)$ of the classes A or B that lie above, or respectively, below the decision plane $d(x, p^A) = d(x, p^B)$, will correctly be classified by the nearest prototype rule. In practice, d is often imperfect and prototypes are weakly discriminative, i.e., the objects and their nearest prototypes may not belong to the same class. As a result, classes will overlap as judged by the dissimilarity-based features $d(x, p^A)$ or $d(x, p^B)$ [see Fig. 3(b)]. To improve that, dissimilarities have to be appropriately weighted, which leads to an LD function, $w^A d(x, p^A) + w_0^A = w^B d(x, p^B) + w_0^B$ [see Fig. 3(b)]. However, when some prototypes are badly discriminative, a quadratic decision may be necessary [see Fig. 3(c)].

This intuition remains valid for any prototype set $R = \{R_A, R_B\}$, $|R| = n$, except that the 1NN rule becomes a piecewise linear function in a dissimilarity space $D(\cdot, R)$. If $\min_j\ d(x, p_j^A) \leq \min_l\ d(x, p_l^B)$, then x is assigned to A, and otherwise to B. Since such a rule may be badly influenced by noisy examples or weak prototypes, statistics over dissimilarities can help. A more robust rule, therefore, is

$$x \to A, \text{ if } \frac{1}{n_A} \sum_j d(x, p_j^A) \leq \frac{1}{n_B} \sum_l d(x, p_l^B) \qquad (1)$$

$$x \to B, \qquad \text{otherwise}$$

which is a simple LD function in $D(\cdot, R)$. In such a reasoning, we have implicitly assumed that dissimilarity values $d(\cdot, p_i)$ span similar range for all p_i. This often does not hold in practice, e.g., for imperfect measures, weak prototypes, or data with clusters. To account for different discriminative powers of prototypes, we need to weight the dissimilarity contributions appropriately. Negative weights may be useful, e.g., to diminish the influence of bad prototypes. Moreover, to emphasize either small or large dissimilarities, a power transformation can be included. So, for $q > 0$, we arrive at the following rule:

$$x \to A, \text{ if } \sum_j w_j^A\ d^q(x, p_j^A) + w_0^A \leq \sum_l w_l^B\ d^q(x, p_l^B) + w_0^B$$

$$x \to B, \quad \text{otherwise.} \qquad (2)$$

This decision boundary is a linear combination of the dissimilarities in their qth power, hence, a linear classifier in the space $D^{\star q}(\cdot, R)$ [see also Fig. 3(b)]. When some prototypes are badly discriminative, a quadratic function may be more successful than a linear one [see Fig. 3(c)].

Linear or quadratic classifiers can be defined by sparse mathematical programming methods or based on hypothesized models. One may assume that classes are approximately normally distributed in dissimilarity spaces. This is reasonable when distances are based on sums of differences and their variances are in similar order of magnitude. Such resulting dissimilarities will approximate normal distributions (or χ^2 distribution if there are some dominant variances) due to the central limit theorem. This observation suggests that NLCs may perform well in dissimilarity spaces.

E. River Example

Artificial 2-D river example is now used to illustrate how linear classifiers in dissimilarity spaces become nonlinear classifiers in the original feature spaces. This follows because the nonlinearity about the problem is incorporated into the representation, here, by Euclidean distances. The variability between the classes is captured properly when different examples are chosen as prototypes. A linear combination of the dissimilarities to such prototypes can model nonlinear boundaries in the feature space. The data consist of two classes, ideally separated by a sine-shaped discriminant, as shown in Fig. 4(a) and (b). First, a learning set L of 1000 examples per class is created. Then, a collection of training sets $\{\mathcal{X}\}$ of growing sizes is sampled from L. For each set $\mathcal{X}$, classifiers are trained in dissimilarity spaces based on the Euclidean distance. The performance is measured

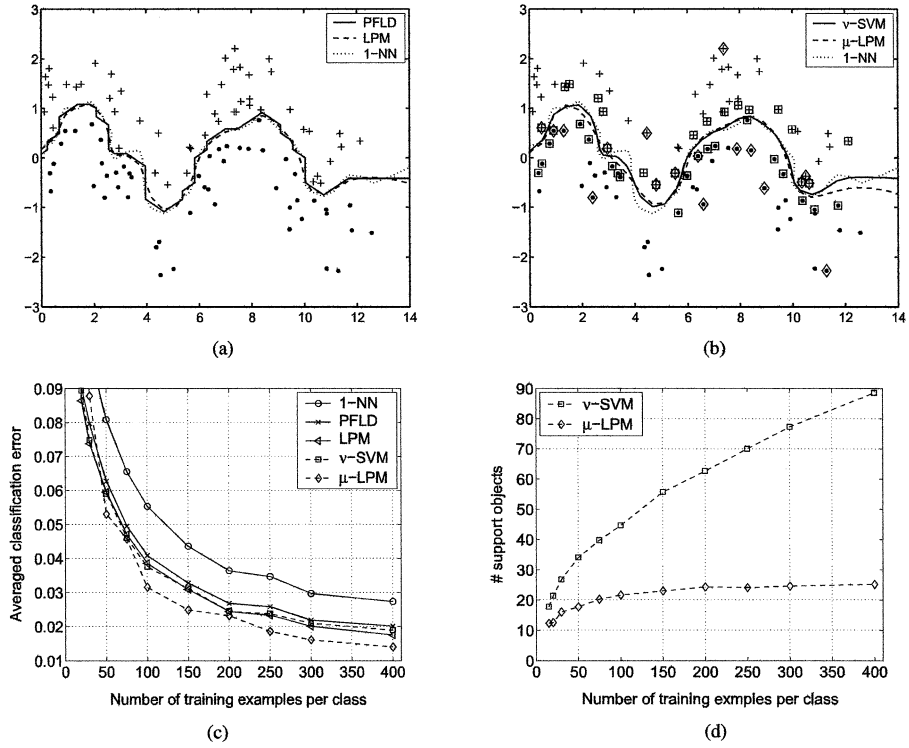


Fig. 4. River example. 1NN rule versus linear classifiers in $D(\cdot, \mathcal{X})$. (b) R is automatically detected. Prototypes are denoted by squares for ν-SVM and by diamonds for μ-LPM [1NN rule versus sparse linear classifiers in $D(\cdot, R)$]. (c) All standard errors of the mean results are smaller than 0.003 for $|\mathcal{X}| \geq 50$ per class (average classification errors). (d) μ-LPM requires only a fixed number of prototypes, in contrast to ν-SVM (# prototypes as a function of $|\mathcal{X}|$ per class).

on the test set $L \backslash \mathcal{X}$. This procedure is repeated 25 times and the results are averaged.

Two classifiers are trained in the complete dissimilarity space $D(\cdot, \mathcal{X})$. These are a pseudo-Fisher linear discriminant (PFLD) based on a pseudoinverse of the singular covariance matrix and linear programming machine (LPM), a hyperplane found by solving a standard nonsparse linear programming problem. The other two classifiers, ν-SVM and μ-LPM [34], automatically determine the prototype set R—objects that support the decision boundary (see Section V for details). ν-SVM relies on the cpd kernel $K = -D$, where D is a Euclidean distance matrix (see Section II-B). All other classifiers are trained on D. For simplicity, we set $\nu = 0.05$ and $\mu = 0.05$. If the optimal values of ν and μ are sought, e.g., by tenfold cross-validation, then the ν-SVM may improve. The results presented in Fig. 4(c) show that the 1NN rule (the best kNN rule here) is systematically outperformed by linear classifiers in the Euclidean dissimilarity space.

IV. Finding the Representation Set

A set R in a dissimilarity space $D(\mathcal{X}, R)$ plays a similar role as a condensed set for the 1NN rule. Once selected, however, the condensed set defines the 1NN rule independently of the remaining training objects, in contrast to classifiers built in RSs. R should be chosen such that it enables both high recognition accuracy and low computational effort. Since similar objects yield similar contributions, only some of them need to be included in R. Moreover, for a multimodal problem, it may be advantageous to select objects related to such modes, e.g., detected via clustering techniques. Such unsupervised methods do not, however, consider the quality of the resulting set R in terms of class separability. This can be done by employing a separability criterion in a process of either feature or instance selection or by optimizing classification performance. In total, R will consist of n objects, selected from the training set $\mathcal{X}$. If an algorithm is applied in a class-wise manner, then n_i objects are chosen for each class ω_i such that $\sum_i n_i = n$.

A. Pseudo-Euclidean Embedding Approach

We now consider a dissimilarity-preserving projection of a symmetric dissimilarity matrix $D(\mathcal{X}, \mathcal{X})$ into a pseudo-Euclidean space $\mathcal{E} = \mathbb{R}^{(p,q)}$. Since many "extracted features" tend to be uninformative due to low variances, the dimension is determined by the number $m = p' + q'$ of dominant eigenvalues. Given an embedded representation X' in $\mathbb{R}^{(p',q')}$, we want to find the set R such that the projection based by R (with the remaining $\mathcal{X} \backslash R$ objects projected afterwards) gives a configuration X'_R that is beneficial for learning. A faithful representation of X' is not necessarily of interest here, since our goal is good classification in the embedded space. So, R should ideally preserve separability between the classes.

Let us first focus on the representation aspects. Our reasoning starts from X' whose mean coincides with the origin in the pseudo-Euclidean space. The origin is first shifted to $\mathbf{x}'_{(1)}$, which is the projection of the prototype $p_{(1)}$, found as the closest one to the origin. For simplicity, let X' now refer to such a shifted configuration. Let $R_1 = \{p_{(1)}\}$ be the first chosen prototype. Other

objects are then successively added until $n \geq m+1$ prototypes are found. Let $R_j = \{p_{(1)}, \ldots, p_{(j)}\}$ be the representation set after the jth step. In each step, an object is selected that minimizes some criterion [57]. This does not guarantee the optimal solution, but the best immediate solution. Two such criteria are described next.

In the average projection error (APE) criterion, $\mathcal{E}_j$ denotes a $(j-1)$-dimensional subspace, defined by R_j, of the complete space $\mathcal{E} = \mathbb{R}^{(p,q)}$. We will iterate over all candidate objects z to find the optimal $p_{(j+1)}$ for R_{j+1}. For each z, a subspace $\mathcal{E}_{j+1}$ is defined by $\{R_j, z\}$, hence, determined from a dissimilarity matrix $D_{jz} := D([R_j, z], [R_j, z])$. Based on the properties of $\langle \cdot, \cdot \rangle_{\mathcal{E}}$ and given that $p_{(1)}$ is projected as $\mathbf{x}'_1$ at the origin, the square approximation error of projecting $\mathbf{x}'_i \in \mathbb{R}^m$ into $\mathcal{E}_{j+1}$ is $e_{\mathrm{apr}}(\mathbf{x}'_i) = ||\mathbf{x}'_i - \mathbf{x}_i^{'\mathcal{E}_{j+1}}||^2_{\mathcal{E}} = d^2(p_{(i)}, p_{(1)}) - (\mathbf{g}_{\cdot i}^{\mathrm{new}})^{\mathsf{T}} G^{-1} \mathbf{g}_{\cdot i}^{\mathrm{new}}$, where G is the Gram matrix and $\mathbf{g}_{\cdot i}^{\mathrm{new}}$ is the ith column of the cross-Gram matrix G_{new}. Both G and G_{new} refer to the representations in $\mathcal{E}_{j+1}$. Hence, $G = -(1/2) J_{(1)} D^{\star 2}_{jz} J^{\mathsf{T}}_{(1)}$, where $J_{(1)} = I - \mathbf{1}\mathbf{s}^{\mathsf{T}}$ projects $p_{(1)}$ at the origin; $\mathbf{s}$ is a zero vector except for $s_{(1)} = 1$. $G_{\mathrm{new}} = -(1/2)(D^{\star 2}_{jz,\mathrm{new}} - \mathbf{1}\mathbf{s}^{\mathsf{T}} D^{\star 2}_{jz}) J^{\mathsf{T}}_{(1)}$. In the $(j+1)$th step, we look for a z such that the APE error $\sum_{\mathbf{x}'_i \in X'} e_{\mathrm{apr}}(\mathbf{x}'_i)$ onto $\mathcal{E}_{j+1}$ is the smallest and the z for which this holds is the prototype $p_{(j+1)}$.

In the criterion based on the largest approximation error (LAE), candidate objects z are also iteratively evaluated to find the best one to be chosen as $p_{(j+1)}$. An object z is selected for $p_{(j+1)}$ such that it yields the LAE when projected onto a space $\mathcal{E}^j$ defined by R_j. Note that for $R_1 = \{p_{(1)}\}$, $e_{\mathrm{apr}}(\mathbf{x}'_i) = d^2(p_{(i)}, p_{(1)})$.

Concerning supervised techniques, two variants of forward feature selection are considered here. The first approach FSel-M relies on repetitive embeddings. Starting from $R_1 = \{p_{(1)}\}$, prototypes are sequentially added. In the $(j+1)$th step, a candidate object z (here from a set of randomly preselected objects to speed up the computation) is chosen as $p_{(j+1)}$ as the one for which the embedded configuration X_{j+1} of $D(\mathcal{X}, [R_j, z])$ maximizes the smallest Mahalanobis distance between the classes. This Mahalanobis distance in a pseudo-Euclidean space simplifies to the one computed in the Euclidean case [27], [57]. As earlier, the embedding is performed each time when a candidate object is evaluated. Since the pseudo-Euclidean embedding relies on square dissimilarities $D^{\star 2}$, the second FSel-M2 approach works in the square dissimilarity space $D^{\star 2}(\cdot, \mathcal{X})$. The prototypes are selected in a greedy fashion to maximize the smallest Mahalanobis distance between the classes in the dissimilarity space. The search is therefore faster than for the FSel-M criterion.

B. Dissimilarity Space Approach

Given original vectorial data, prototypes can be generated in a feature space. This can be realized by clustering data into groups and merging their members into prototypes. The most common approach is the k-means/expectation-maximization (EM) clustering algorithm [21], which models clusters by Gaussian distributions. The cluster means define the prototypes. Linear subspaces can also be used to describe clusters such that their weighted means are chosen as prototypes. So, $D(\mathcal{X}, R)$ relies on prototypes created from the initial training feature vectors [46].

If no original vectorial representation is available, prototype selection techniques are applied. Unsupervised approaches include k-centers (KCent) and mode seeking (ModeSeek). The KCent algorithm, if applied class-wise, looks for a set $R_{\omega_i} = \{p_1^{\omega_i}, \ldots, p_{n_i}^{\omega_i}\}$, evenly distributed with respect to $D(\mathcal{X}_{\omega_i}, \mathcal{X}_{\omega_i})$, where $\mathcal{X}_{\omega_i}$ denotes the training objects of ω_i. So, $p_i^{\omega_i}$ are chosen to minimize $E = \max_i \min_z d(p_i^{\omega_i}, p_z^{\omega_i})$. The search is performed in a forward strategy, starting from a random initialization [60]. The final prototypes minimize E over $M = 50$ trials. In the ModeSeek algorithm, R_{ω_i} is a set of estimated modes of the distribution of ω_i with respect to $D(\mathcal{X}_{\omega_i}, \mathcal{X}_{\omega_i})$ [9]. $|R_{\omega_i}|$ depends on the neighborhood size s_n. The larger s_n, the smaller R_{ω_i}. So, s_n is chosen to generate the largest set R, not larger than the demanded one.

Concerning supervised techniques, editing and condensing (EdCon) schemes can be used [15], [46]. Editing removes noisy examples before condensing can guarantee good NN performance on the reduced set. R can also be determined by a sparse LPM in a dissimilarity space $D(\cdot, \mathcal{X})$. A solution is obtained, e.g., by μ-LPM, which looks for a separating hyperplane $f(D(x, \mathcal{X})) = \mathbf{w}^{\mathsf{T}} D(x, \mathcal{X}) + w_0$. Sparseness is imposed by minimizing $||\mathbf{w}||_1 = \sum_{j=1}^{N} |w_j|$ [34]. The prototype set R is found automatically and consists of objects corresponding to nonzero weights. Since an (indefinite) SVM can also be trained on $K = -D$, R can be chosen as the set of support vectors. However, note that SVM cannot always be found for arbitrary K (non-Euclidean D) [35].

Finally, feature selection (FSel) methods try to find an optimal set of n dissimilarity-based features in the space $D(\cdot, \mathcal{X})$ according to some class separability function. A greedy forward selection may be employed using several criteria such as the Mahalanobis distance or the LOO 1NN error. The latter approach is also modified here to make use of the given dissimilarity data. A set of prototypes is selected according to the 1NN LOO error; however, the features are now interpreted in a dissimilarity space, while the 1NN error is derived on the distances $D(\mathcal{X}, \mathcal{X})$ directly, and *not* on distances computed in a dissimilarity space. The method is thereby fast as it is entirely based on comparisons and sorting of the already given dissimilarities. Ties are solved by selecting the set R for which the sum of dissimilarities is minimum.

V. Experiments With Vectorial Data

We will now illustrate the potential of linear and quadratic classifiers trained in dissimilarity spaces arising from vectorial representations. Eight vectorial datasets from http://www.ics.uci.edu/~mlearn/MLRepository.html are used with categorical, continuous, and mixed features [58]. City block distances (l_1) are employed for data with categorical or mixed types after a linear scaling to the same domain (interval). Euclidean distances (l_2) are used otherwise (see Table I for details).

Datasets are split into training sets $\mathcal{X}$ and test sets U in the ratio of 75% : 25%. The sets are first appropriately scaled, and then, the distance representations are derived. Four different

TABLE I
VECTORIAL DATA

Data	#Obj/Dim	Class sizes	Variable	Scaling	Dist.
Australian	690/14	383/307	Mixed	D	l_1
Biomed	194/5	127/67	Mixed	D	l_1
Breast	683/9	444/239	Cat.	D	l_1
Diabetes	768/8	500/268	Mixed	D	l_1
Heart	297/3	160/137	Mixed	D	l_1
Ecoli	272/5	143/77/52	Cont.	S	l_2
Glass	214/9	70/76/17/51	Cont.	S	l_2
Ionosphere	351/32	225/126	Cont.	S	l_2
Liver	345/6	145/200	Cont.Int.	S	l_2
Sonar	208/60	97/111	Cont.	S	l_2

"D" stands for domain scaling of features, while "S" stands for data standardization.

methods are used to optimize R: *EMgen*, *FSel*, *EdCon*, and *LPauc*. *EMgen* creates a set of prototypes, cluster means, in the original feature space. Clusters, modeled by Gaussians with diagonal covariance matrices, are found by the EM algorithm. (Note that noise is added to categorical data to prevent degenerated solutions.) Other methods select prototypes from the training set $\mathcal{X}$ by working on $D(\mathcal{X}, \mathcal{X})$. *FSel* finds a set of prototypes by a forward feature selection in a dissimilarity space $D(\cdot, \mathcal{X})$ with the criterion based on the Mahalanobis distance. *EdCon* is the traditional edited and condensed set optimized for the NN performance [15]. *LPauc* gives a prototype set optimized for the performance of a sparse LPM, auc-LPM, defined by maximizing the area under the receiver operating characteristic (ROC) curve (see [72] for details). $|R|$ is set *a priori* for *EMgen* and *FSel* such that $n_i = \lceil \sqrt{|\omega_i|} \rceil$ prototypes are optimized per class. If the dimensionality is very high (e.g., Sonar data), $|R| = \sum_i n_i + \log(\#\dim)$. Other methods determine $|R|$ automatically.

Four classifiers are trained by using all training objects in dissimilarity spaces $D(\cdot, R)$. These are normal density-based linear classifier (NLC), normal density-based quadratic classifier (NQC), regularized by a parameter $\lambda = 10^{-4}$ [60], logistic linear classifier (LOGC) [21], and 1NN based on Euclidean distances in the dissimilarity space $D(\cdot, R)$. In addition, the direct NN rule, 1NNd, is applied to $D(U, R)$ and the direct 1NNd and kNNd rules (with k optimized by an LOO error on $\mathcal{X}$) are applied to $D(U, X)$. As indicated, three classifiers are also trained in the initial feature spaces. These are NQC, NaiveBC (naive Bayes), and ν-SVM with a Gaussian kernel (ν is an LOO estimation of the 1NN error and σ is optimized by tenfold cross-validation). Prior probabilities are estimated by class frequencies. The procedure is repeated 30 times and the results are averaged.

Important observations can be made by analyzing Table II. First, both NLC and NQC trained in dissimilarity spaces $D(\cdot, R)$ outperform the direct 1NNd rule when based on the same R, irrespectively of how R is determined. This also holds for the *EdCon* condensed sets optimized for the performance of the 1NN. Second, NLC (and often NQC) trained in $D(\cdot, R)$ outperforms the 1NNd rule and performs similarly or better than the kNNd rule, both applied to $D(\cdot, \mathcal{X})$, i.e., based on the complete training sets $\mathcal{X}$. Moreover, NLC almost always outperforms LOGC (and is often significantly better) when both trained in $D(\cdot, R)$. This speaks in favor of the assumption of normally distributed classes in such dissimilarity spaces. Next, the *EMgen*-prototypes lead to somewhat better results than the *Fsel*-prototypes, in general. Our way of setting $|R|$ for these two methods works well when the number of features is not too large; it may however be insufficient for high-dimensional data such as Sonar. Note that the automatically detected cardinality of the *EdCon*- and *LPauc*-sets may be up to a few times our fixed cardinality for the *FSel*- and *EMGen*-sets. The *LPauc*-sets are well suited for highly overlapping classes, as for the diabetes, heart, glass, or liver data.

In relation to ν-SVM with an optimized Gaussian kernel, dissimilarity-based NLC is very attractive. It is simple and outperforms ν-SVM for moderately overlapping classes of the Australian or heart data, or for highly overlapping classes of the diabetes, glass, and liver data. Moreover, it usually requires less representation objects than support vectors needed by ν-SVM (compare $|R|$ to #SVs in the table).

In summary, NLC and NQC built in dissimilarity spaces $D(\cdot, R)$ are often more advantageous than the direct 1NN rule based on R. They also perform similarly or better than the best kNN rule on $D(\cdot, \mathcal{X})$, i.e., based on the training set $\mathcal{X}$. They can be recommended for problems with categorical/mixed variables, or moderate-high class overlap. In these cases, ν-SVM and other classifiers in feature spaces will tend to lose.

VI. EXPERIMENTS WITH NONVECTORIAL DATA

Five dissimilarity datasets are used in our study based on generated polygons, scanned digits, geophysical spectra, road sign images, and proteins. *PolyDist* data consist of two classes, 2000 examples each, of randomly generated convex quadrilaterals and irregular heptagons. The polygons are first normalized, and then, the modified Hausdorff distances [20] between their vertices are computed. *Zongker* digit data describe ten digit classes, each of 200 examples. Shapes of digits (from binary images) are compared by an asymmetric similarity s_{ij} based on deformable template matching [42]. Nonmetric dissimilarities are derived as $d_{ij} = \sqrt{s_{ii} + s_{jj} - s_{ij} - s_{ji}}$. *GeoShape* data consist of two multimodal geological classes, 500 examples each, described by high-dimensional wavelength spectra. The spectra are first normalized to a unit area, and then, the l_1 distances between their Gaussian smoothed ($\sigma = 2$ bins) first-order derivatives are computed [53]. *RoadSign* data consist of gray-level images of circular road signs. Three hundred road sign images (highly multimodal) and 300 outlier images acquired under general illumination are used [54]. Dissimilarities are derived as $d_{ij} = \sqrt{1 - s_{ij}}$, where s_{ij} is a normalized cross-correlation. Finally, *ProDom* is a subset of 2604 protein domain sequences

TABLE II
AVERAGE CLASSIFICATION ERRORS (IN PERCENT) FOR DISSIMILARITY-BASED AND FEATURE-BASED CLASSIFIERS

R optimized by	$\|R\|$	Direct	Dissimilarity space				Orig. feature space	
		1NNd / kNNd	LOGC	NLC	NQC	1NN		
2-class Australian data: 14 features								
EMgen	23	16.2(0.6)	13.9(0.5)	13.5(0.5)	14.2(0.4)	20.6(0.5)	NQC	20.0(0.5)
FSel	23	23.4(1.1)	14.6(0.3)	13.9(0.4)	15.2(0.4)	20.6(0.3)	NaiveBC	14.8(0.5)
EdCon	96	17.0(0.4)	17.4(0.4)	14.0(0.4)	19.5(0.4)	21.0(0.4)	SVM	15.6(0.4)
LPauc	18	22.9(0.2)	14.6(0.1)	13.9(0.1)	13.9(0.1)	20.2(0.1)	#SVs	178
All ($R=\mathcal{X}$)	519	20.7(0.4) / 14.1(0.4)	—	—	—	—	—	—
2-class Breast data: 9 features								
EMgen	23	5.9(0.3)	4.5(0.3)	3.0(0.2)	3.6(0.3)	3.6(0.2)	NQC	4.7(0.2)
FSel	23	5.6(0.4)	4.7(0.3)	3.3(0.2)	4.2(0.3)	3.9(0.3)	NaiveBC	2.8(0.2)
EdCon	28	4.6(0.3)	5.0(0.5)	3.1(0.2)	5.0(0.3)	3.8(0.3)	SVM	5.0(0.4)
LPauc	9	14.7(0.3)	3.6(0.0)	3.2(0.0)	3.2(0.0)	4.4(0.1)	#SVs	193
All ($R=\mathcal{X}$)	513	4.2(0.3) / 3.5(0.3)	—	—	—	—	—	—
2-class Diabetes data: 8 features								
EMgen	24	27.9(0.5)	23.6(0.5)	23.4(0.4)	24.8(0.5)	32.6(0.4)	NQC	25.6(0.4)
FSel	24	32.5(0.7)	24.3(0.5)	24.1(0.4)	24.2(0.5)	31.1(0.5)	NaiveBC	24.5(0.5)
EdCon	147	30.4(0.6)	28.3(0.5)	25.9(0.5)	25.6(0.5)	31.4(0.4)	SVM	28.3(0.6)
LPauc	36	39.0(0.2)	25.4(0.1)	24.5(0.1)	24.7(0.1)	31.9(0.1)	#SVs	280
All ($R=\mathcal{X}$)	576	31.0(0.5) / 26.0(0.4)	—	—	—	—	—	—
2-class Heart data: 13 features								
EMgen	15	19.0(0.9)	18.1(0.7)	17.0(0.6)	20.2(0.6)	22.9(0.8)	NQC	19.1(0.6)
FSel	15	25.3(1.4)	18.7(0.7)	17.4(0.6)	19.0(0.6)	22.5(1.0)	NaiveBC	17.3(0.6)
EdCon	51	21.8(0.9)	21.4(0.9)	18.6(0.6)	23.0(0.7)	22.5(0.8)	SVM	18.8(0.5)
LPauc	16	27.9(0.2)	20.5(0.2)	16.2(0.2)	16.2(0.2)	22.3(0.2)	#SVs	102
All ($R=\mathcal{X}$)	223	20.9(0.6) / 18.3(0.6)	—	—	—	—	—	—
4-class Glass data: 9 features								
EMgen	21	36.4(1.1)	42.0(3.3)	31.4(1.2)	27.8(0.9)	26.3(0.8)	NQC	39.4(0.9)
FSel	21	41.6(1.6)	34.6(1.9)	29.9(1.2)	25.2(0.7)	28.3(0.8)	NaiveBC	33.9(1.2)
EdCon	46	33.3(1.0)	53.7(3.0)	28.4(1.1)	25.6(0.8)	27.5(0.8)	SVM	46.6(1.3)
LPauc	72	35.2(0.2)	27.6(0.2)	28.2(0.2)	28.8(0.2)	26.7(0.2)	#SVs	9
All ($R=\mathcal{X}$)	162	28.0(1.0) / 29.6(1.0)	—	—	—	—	—	—
2-class Ionosphere data: 32 features								
EMgen	16	17.5(0.7)	6.8(0.3)	6.6(0.4)	5.7(0.4)	5.4(0.4)	NQC	14.3(0.6)
FSel	16	15.1(0.7)	6.6(0.3)	6.4(0.3)	5.3(0.3)	5.8(0.3)	NaiveBC	9.1(0.5)
EdCon	26	16.5(0.5)	8.2(1.2)	6.9(0.4)	5.5(0.3)	9.2(0.6)	SVM	5.7(0.4)
LPauc	25	19.7(0.2)	6.3(0.1)	6.7(0.1)	6.8(0.1)	6.1(0.1)	#SVs	64
All ($R=\mathcal{X}$)	264	13.0(0.5) / 13.0(0.5)	—	—	—	—	—	—
2-class Liver data: 6 features								
EMgen	16	42.8(1.1)	29.0(0.8)	29.7(0.8)	41.5(1.0)	42.9(0.7)	NQC	40.3(1.2)
FSel	16	44.1(1.1)	29.5(0.9)	29.1(0.9)	38.1(0.9)	41.9(0.8)	NaiveBC	37.1(0.9)
EdCon	95	42.3(0.7)	35.4(1.4)	29.4(0.7)	36.8(0.9)	43.9(0.9)	SVM	30.9(0.9)
LPauc	57	45.7(0.2)	39.7(0.2)	28.4(0.1)	28.4(0.1)	42.2(0.1)	#SVs	148
All ($R=\mathcal{X}$)	259	38.1(0.8) / 36.7(0.6)	—	—	—	—	—	—
2-class Sonar data: 60 features								
EMgen	21	19.8(1.3)	20.0(1.1)	19.6(1.1)	16.9(1.1)	20.4(1.2)	NQC	31.8(1.0)
FSel	21	27.2(1.2)	22.2(1.1)	20.8(1.3)	20.8(1.0)	21.0(1.3)	NaiveBC	26.4(1.2)
EdCon	39	23.8(1.1)	22.6(1.1)	19.3(1.2)	15.8(0.9)	20.7(0.9)	SVM	15.9(1.1)
LPauc	43	26.5(0.3)	18.4(0.2)	20.5(0.2)	20.6(0.2)	18.6(0.2)	#SVs	84
All ($R=\mathcal{X}$)	157	16.1(0.8) / 16.6(1.0)	—	—	—	—	—	—

Standard errors are in brackets.

from the ProDom set [10]. We use the same four-class problem (878/404/271/1051 examples) as in [62]. Dissimilarities are derived as $d_{ij} = \sqrt{s_{ii} + s_{jj} - 2s_{ij}}$, where s_{ij} are pairwise structural alignments computed by Roth [62].

Basic properties of these measures, linearly scaled to small ranges, are characterized in Table III [60]. Let λ denote eigenvalues arising from a pseudo-Euclidean embedding. The magnitudes of negative eigenvalues indicate the amount of deviation from the Euclidean behavior, as captured by two indexes: $r^{nE}_{mm} = |\lambda_{\min}|/\lambda_{\max}$, the ratio of largest (in magnitude) negative and positive eigenvalues, and $r^{nE}_{\text{rel}} = \sum_{i=p+1}^{p+q} |\lambda_i| / \sum_{i=1}^{N} |\lambda_i|$, the relative contribution of negative eigenvalues. The nonmetric behavior can be quantified by the percentage of disobeyed triangle inequalities r^{nM}_{tr}.

Classification experiments conducted in dissimilarity-based RSs aim to illustrate the behavior of prototype selection techniques. We choose to train NQC. It tends to work well with dissimilarity data derived from nonvectorial collections, especially for homogeneous dissimilarities based on sums of differences. NQC is more flexible than NLC, although not

TABLE III
PROPERTIES OF DISSIMILARITY DATASETS USED IN EXPERIMENTS

Data	#Obj	#Class	α	Dissimilarity	Property	r_{mm}^{nE} [%]	r_{rel}^{nE} [%]	r_{tr}^{nM} [%]
PolyDist	4000	2	0.05	Modified Hausdorff	Non-metric	11.0	31.4	0.01
RoadSign	600	2	0.33	Correlation	Euclidean	0.0	0.0	0.00
GeoShape	1000	2	0.2	Shape l_1	Metric, non-Euclidean	2.6	7.2	0.00
ProDom	2604	4	0.35	Structural	Non-metric	1.3	0.9	10^{-5}
Zongker	1000	10	0.25	Template-match	Non-metric	38.9	35.0	0.41

α is the fraction of training objects per class. r_{mm}^{nE} and r_{rel}^{nE} indicate deviation from the Euclidean behavior and r_{tr}^{nM} is the percentage of disobeyed triangle inequalities.

always better. It is also costly if R is large and there are many classes. Usually, regularization is necessary for high-dimensional RSs (large R), because class covariance matrices become singular. Here, a tradeoff NQC, the TNQC rule, is trained, in which class covariance matrices C_i are regularized as $C_i^\kappa = (1-\kappa)\, C_i + \kappa\, p(\omega_i)\, \text{diag}\,(C_i)$, where $\kappa \in [0,1]$ and $p(\omega_i)$ denotes prior probabilities. To speed up computations, the regularization parameter κ is set to a value from $[0.001, 0.3]$ found in a fivefold cross-validation. This is done separately in dissimilarity spaces and in embedded spaces for the largest R considered.

In each experiment, the datasets are divided into training sets $\mathcal{X}$ and test sets U. Then, a representation set $R \subset \mathcal{X}$ is chosen according to the criteria described in Section IV. Both embedded pseudo-Euclidean spaces and dissimilarity spaces are used. First, an embedded space $\mathcal{E}$ is determined by $D(R,R)$ and all objects $D(\mathcal{X},R)$ are orthogonally projected there as X_T. TNQC is trained on X_T in $\mathcal{E}$ and tested on the projected evaluation objects X_U. Here, the dimension m of $\mathcal{E}$ is automatically detected by the number of dominant eigenvalues (found as a point where the "magnitude eigenvalue curve" flattens). This means that m may grow with the training size. Concerning the dissimilarity space, TNQC is trained on $D(\mathcal{X},R)$ and tested on $D(U,R)$. All experiments are repeated 30 times and the results are averaged.

Figs. 5 and 6 show generalization errors of TNQC as a function of $|R|$ for various prototype selection methods. Standard deviations are omitted to maintain clarity. They vary between $3\%\,\bar{e}$ for small $\bar{e}$ and $7\%\,\bar{e}$ for large $\bar{e}$, where $\bar{e}$ is the average error. For example, if $\bar{e} = 0.15$, its standard error is ≈ 0.01, while if $\bar{e} = 0.05$, it is equal to ≈ 0.002. To enhance interpretability of the results, supervised methods are plotted by continuous lines, unsupervised techniques are plotted by dash-dotted lines and random methods are plotted by dashed lines. *EdCon*-1*NN* stands for the 1NN result for the edited and condensed set. Additionally, also μ-LPM and ν-SVM are applied to $D(\mathcal{X},\mathcal{X})$ to automatically detect R (indefinite ν-SVM [35] is used for $K = -D$). Hence, *EdCon*, *LPM*, and *SVM* denote the errors of TNQC for representation sets chosen by these methods, while *LPM-LPM* and *SVM-SVM* refer to their original results. For example, *LPM-LPM* means the LPM result in a dissimilarity space based on R also selected by LPM. Concerning the NN methods, 1*NNd-final* and k*NNd-final* stand for the direct NN rules applied to $D(\cdot,\mathcal{X})$. The corresponding errors set our reference and are plotted as horizontal lines. k*NNd* is the direct kNN rule applied to $D(U,R)$, while k*NN* is based on Euclidean distances in the dissimilarity space $D(\cdot,R)$. k is optimized by an LOO procedure over $\mathcal{X}$, while R is randomly chosen.

Our main conclusion is that a suitably regularized TNQC built in RSs defined by $D(\cdot,R)$ leads to a significant improvement over the direct kNN applied to $D(\cdot,\mathcal{X})$. This is already achieved for R consisting of $n \leq \sum_{i=1}^{F} \lceil \sqrt{n_i} \rceil$ prototypes, where n_i are class sizes. Although RSs are defined by R, all training examples from $\mathcal{X}$ are used to build the classifier. This means that a reduced set of dissimilarities to the objects from R has to be computed in the test stage. If the derivation of dissimilarities is costly, such a dissimilarity-based quadratic classifier will be computationally more efficient than the kNN rule applied to $D(U,\mathcal{X})$, especially for a small number of classes F. To see this, compare the complexity $\mathcal{O}(\text{cost}(d)\, F\, n)$ of TNQC in an RS based on $n = \sum_{i=1}^{F} \lceil \sqrt{n_i} \rceil$ prototypes versus the complexity $\mathcal{O}\,(\text{cost}(d)\,(N + F \log F))$ of the direct kNN based on $N = \sum_{i=1}^{F} n_i$ training objects; $\text{cost}(d)$ denotes the average cost of deriving a single dissimilarity value.

The results of TNQC are much better than those of NLC (not shown here due to space limits). The reason is that TNQC is more flexible and may be necessary for imperfect measures and weakly discriminating prototypes, as argued in Section III-D. To avoid the curse of dimensionality, a strong regularization, such as $\kappa \geq 0.05$, is however needed, especially for multiclass problems. Note that TNQC applied here leads to better results than a weakly regularized NQC used before [60].

Concerning prototype selection techniques, forward feature (instance) selection based on the Mahalanobis distance leads to the most efficient results (small generalization error and small R) in both RSs. In case of multimode data, such as *GeoShape*, the mode-seeking algorithm is preferable for a small number of prototypes.

VII. FINAL DISCUSSION

A. Proximity Representations and RSs

Proximity representations offer a universal way to represent information about relations between pairs of objects. They extend kernels to indefinite kernels, dyadic kernels, and other flexible representations. Learning from such proximity data usually relies either on kernel methods or on the NN rule. When traditional kernel methods cannot directly be applied, are too costly or impractical, while the NN rule leads to noisy results,

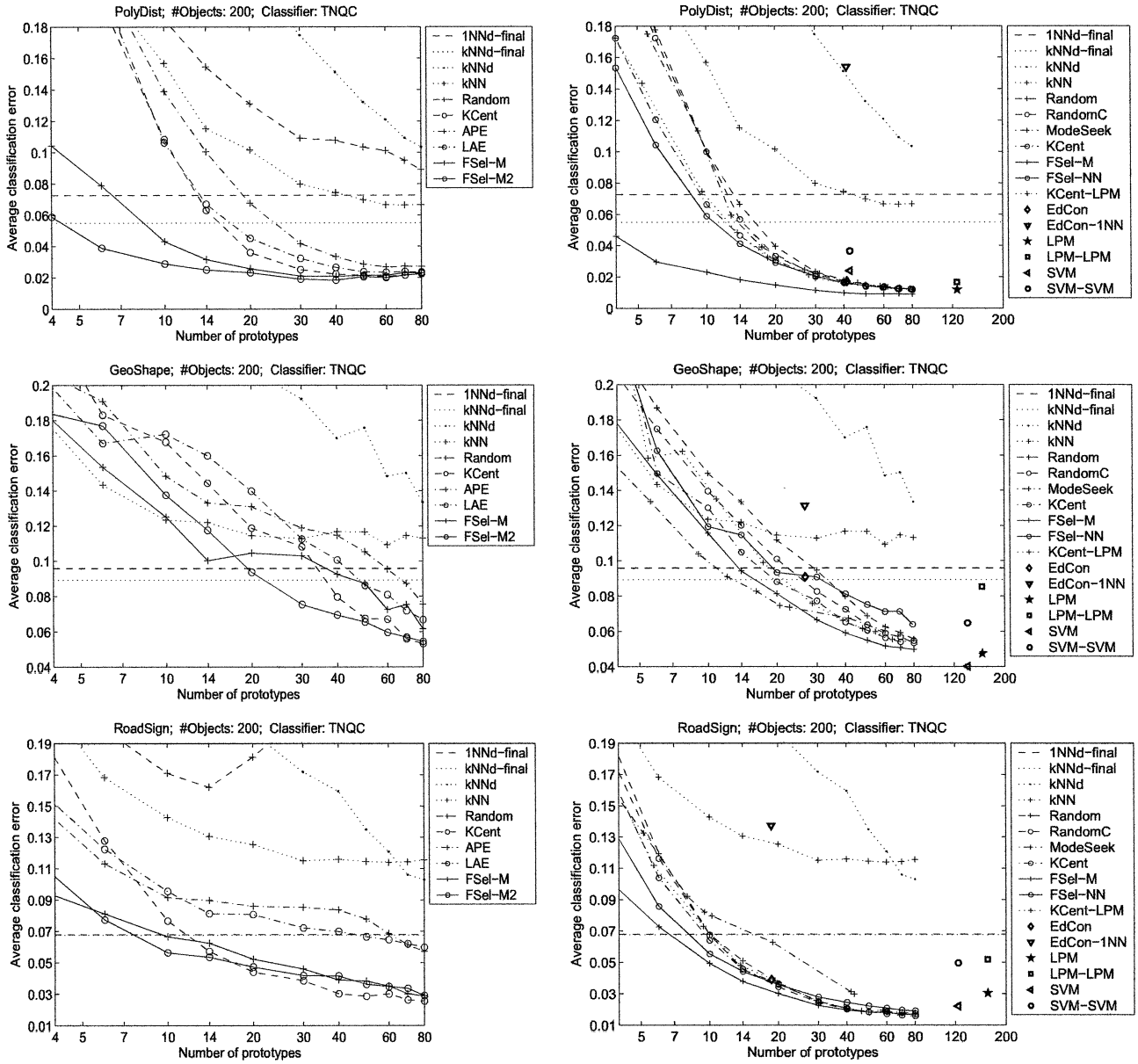


Fig. 5. Two-class dissimilarity data. Average classification errors for dissimilarity-based classifiers as a function of the number of prototypes $|R|$ in pseudo-Euclidean spaces (left) and dissimilarity spaces (right). The x-axis has a logarithmic scale.

an appealing alternative is to train statistical classifiers in two general dissimilarity-based RSs: pseudo-Euclidean and dissimilarity spaces.

These two RSs are essentially different. A pseudo-Euclidean embedded space is realized via a feature map to an indefinite inner product space such that the extracted features preserve the original dissimilarities. Individual features are derived from all dissimilarities. In an isometric mapping, the dissimilarities are perfectly preserved, i.e., $d(p_k, p_l) = d_{PE}(\mathbf{p}_k^{PE}, \mathbf{p}_l^{PE})$ holds for the prototype set, where $\mathbf{p}_k^{PE}$ and $\mathbf{p}_l^{PE}$ are projected vectors. There is no correction of the distances. Denoising, which may come later, removes insignificant dimensions to avoid the curse of dimensionality as well. Original dissimilarities are somewhat changed by this. The non-Euclidean character of the measure remains, however, if it was significantly present in the initial dissimilarities.

Dissimilarity space is realized via a direct map to a Euclidean space. Features are dissimilarity vectors to individual prototypes. Euclidean distances d_E in a dissimilarity space will differ from the original dissimilarities, i.e., $d(p_k, p_l) \neq d_E(D(p_k, R), D(p_l, R)) = (\sum_{p_i \in R}[d(p_k, p_i) - d(p_l, p_i)]^2)^{1/2}$. Objects in the dissimilarity space can have a zero distance d_E only if $d(p_k, p_i) = d(p_l, p_i)\ \forall_{p_i \in R}$. Different objects in a pseudo-Euclidean space can, however, have a zero distance, even if $d(p_k, p_i) \neq d(p_l, p_i)$. So, the pseudo-Euclidean space respects the originally measured dissimilarities, while the dissimilarity space reflects them in the *context* of all objects in the representation set and is

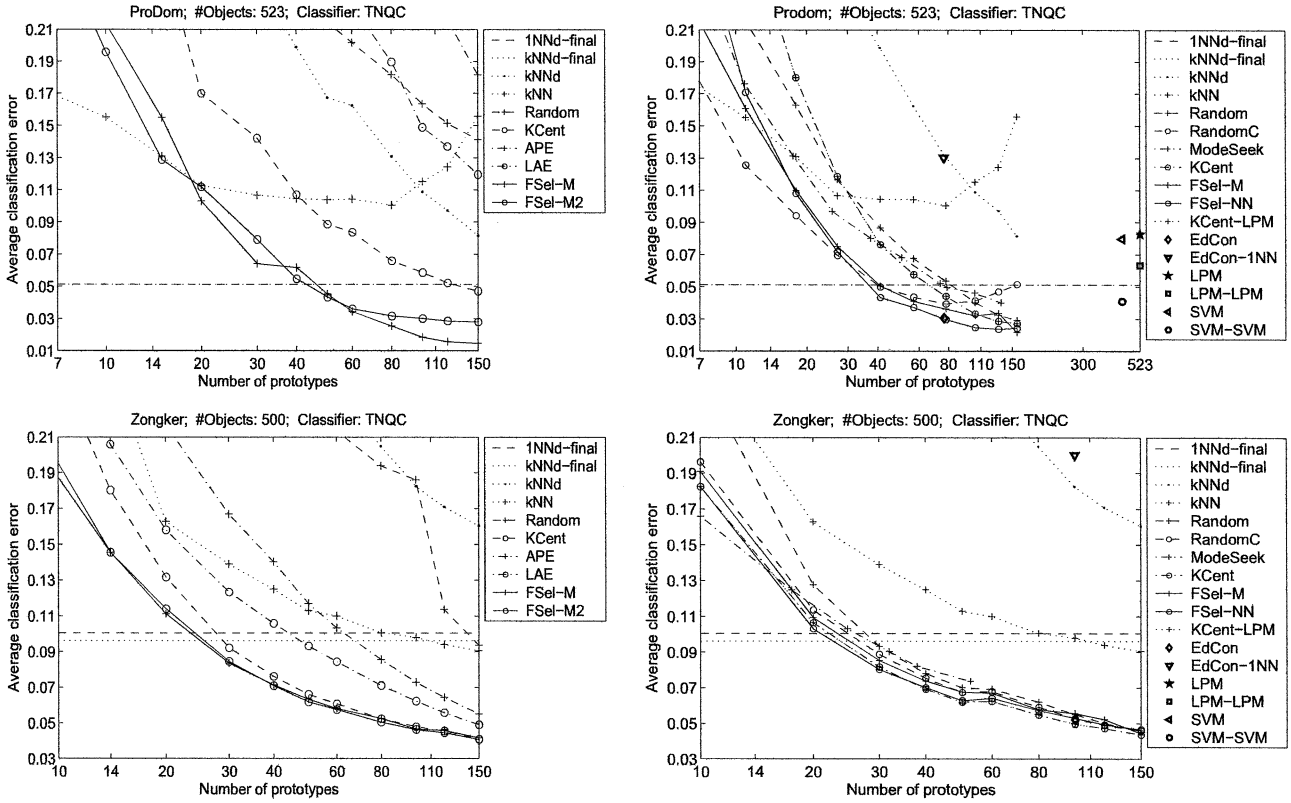


Fig. 6. Multiclass dissimilarity data. Average classification errors for dissimilarity-based classifiers as a function of the number of prototypes $|R|$ in pseudo-Euclidean spaces (left) and dissimilarity spaces (right). ν-SVM and μ-LPM failed for the *Zongker* data. The x-axis has a logarithmic scale.

thereby globally sensitive. What is profitable depends on the application.

B. Classification in RSs

Linear and quadratic classifiers built in the two dissimilarity-based RSs perform better than the direct NN rule for training sets of small and moderate sizes. When a training set is large and all training objects are prototypes, the NN rule will ultimately be the best (for metric distances), as it is Bayes-consistent [16].

Dissimilarity representations are defined in two ways: computed in initial feature spaces or derived from nonvectorial data. LD functions built in dissimilarity spaces defined over feature spaces offer a way to construct nonlinear classifiers there. More importantly, they play a role for data with categorical or mixed variables, or for moderately or highly overlapping classes. We showed that SVM with an optimized Gaussian kernel, a very good classifier for nonoverlapping classes, does not perform well in such cases. A NLC trained in dissimilarity spaces $D(\cdot, R)$ tends to be more advantageous. Prototype sets R can be optimized by procedures such as forward selection, condensing, or sparse linear programming. Although no technique is overall best, prototypes *generated* in the original feature spaces, e.g., by the EM algorithm, make good representation sets of controllable sizes [46]. In addition, linear programming works as an instance selection and can provide efficient prototype sets for highly overlapping classes.

Concerning nonvectorial data of moderate training sizes, NQCs are trained in RSs defined over reduced sets R. Experiments show that they outperform the 1NN rule and behave similarly or better than the kNN rule, both based on the complete training set $\mathcal{X}$. This holds independently whether the measure is metric or not. In general, flexible prototype optimization procedures are of interest. Hence, one may control the tradeoff between recognition accuracy and computational complexity. The forward selection based on the Mahalanobis distance criterion is such a procedure and is often the best. It allows one to select a small prototype set that is advantageous for classifiers in both dissimilarity spaces and pseudo-Euclidean spaces. If there is a strong cluster tendency, however, a procedure like mode seeking may be preferred.

In pseudo-Euclidean embedded spaces, the prototypes define a vector space that is distorted with respect to the one determined by all training examples. The prototypes should be chosen such that this reduced vector space preserves separability between the classes as compared to the complete space.[3] Since our selection procedures do not fully account for data characteristics in the embedded spaces, the classification results may be somewhat worse than the ones in dissimilarity spaces. More study is needed in this direction.

[3]This is analogous to asking how to choose vectors in a high-dimensional vector space that approximate a covariance matrix such that the separability between classes is preserved in a low-dimensional PCA projection.

C. Future Perspectives

The choice of representation set is an important issue. Although many good techniques are available, the best one depends on the problem. For nonvectorial data, the best prototype selection methods rely on some separability criterion. For vectorial data, the best methods generate prototypes from the training examples. How to optimize *a small set of very good prototypes* is still an open question.

A proper formulation of indefinite kernel methods is also an important task. It will help in determining explicit relations between similarity spaces defined by (indefinite) kernels and (Krein) Hilbert spaces induced by theses. This will lead to a unified framework of generalized kernel methods.

In this paper, no direct corrections have been made to force metric constraints or impose the Euclidean behavior. There are, however, studies in which non-Euclidean dissimilarities are corrected to become Euclidean, especially in the context of pseudo-Euclidean embedding [11], [32], [61], [63]. Future investigations are needed to understand when these corrections are necessary, when indifferent [63], or when spurious [59]. We plan to identify the causes of non-Euclidean and nonmetric behavior, such as incorporation of invariance or suboptimal optimization. This is the first step to understand the circumstances under which proximity data should be corrected. Only then, efficient corrections or transformations can be designed.

Finally, proximity representations offer new possibilities for marrying statistical learning techniques with structural and information-theoretic data descriptions. Since proximity measures are defined in all learning contexts, proximity representations offer a natural bridge between the structure and statistics. This is a hybrid approach. The structural or information-theoretic nature of objects is first incorporated into a proximity measure. The resulting proximity representation defines an inner product RS in which statistical methods are used. If sufficient prior knowledge is incorporated into the representation, simple learning methods will give good results.

D. Summary

In this paper, we studied classification methods in two dissimilarity-based RSs: pseudo-Euclidean embedded spaces and dissimilarity spaces. They can handle general proximity measures, including non-psd, non-Euclidean, or nonmetric ones. Such measures are important to study as they result from incorporation of invariance [36] or robustness [41], essential aspects in pattern recognition.

Simple linear and quadratic classifiers trained in these RSs extend the traditional kernel methods and fill the gap when such methods and the direct NN rule fall short. They are thereby especially useful for non-psd, non-Euclidean or nonmetric proximity data, highly overlapping classes or when the representation set has to be controlled, e.g., because of the computational cost. SVM is recommended for problems with small/moderate class overlap described by psd kernels, while the NN rule is recommended for large training/representation sets. In other cases, linear and quadratic classifiers in RSs provide clear advantages over the direct NN rule: better performance and lower (adjustable) computational complexity in classifying new objects. Such advantages have to be paid by more extensive training procedures: optimizing the representation set and training of a classifier in the derived RS. Classification with non-Euclidean and nonmetric dissimilarity representations leads to good results, but the optimal handling of these needs further research.

APPENDIX

Basic definitions and characteristics concerning proximity measures are provided next; see [12], [17], [31], [32], and [57] for more details.

Kernel: A function $K : X \times X \to \mathbb{R}$ of continuous linear operators on a compact set X is a kernel if $K(x, y) = K(y, x)$ $\forall_{x,y \in X}$. A linear operator on functions associated to K is defined by the integral $[L_K f](x) = \int_X K(x, y) f(y)\, dy$.

psd kernels/cpd kernels/conditionally negative definite (cnd) kernels: An $n \times n$ symmetric matrix K is cpd iff $\mathbf{z}^\mathsf{T} K \mathbf{z} \geq 0$ $\forall_{\mathbf{z} \in \mathbb{R}^n}$ such that $\mathbf{z}^\mathsf{T} \mathbf{1} = 0$. If this is satisfied for all $\mathbf{z} \in \mathbb{R}^n$, then K is psd. Symmetric K is cnd iff $\mathbf{z}^\mathsf{T} K \mathbf{x} \leq 0$ $\forall_{\mathbf{z} \in \mathbb{R}^n}$ such that $\mathbf{z}^\mathsf{T} \mathbf{1} = 0$. A function $K(x_i, x_j)$ over $X \times X$ is cpd/cnd/psd, if the previous conditions hold for any n-element subset of X.

Metric: A metric measure $d : X \times X \to \mathbb{R}_+ \cup \{0\}$ obeys the following axioms $\forall_{x,y,z \in X}$: *reflexivity:* $d(x, x) = 0$, *symmetry:* $d(x, y) = d(y, x)$, *definiteness:* $(d(x, y) = 0) \Rightarrow (x = y)$, and *triangle inequality:* $d(x, y) + d(y, z) \geq d(x, z)$. (X, d) denotes a metric space.

ℓ_p-distance: Family of ℓ_p-distances $(\mathbb{R}^m, d_p)$ with $d_p(\mathbf{x}, \mathbf{y}) = (\sum_{i=1}^m |x_i - y_i|^p)^{1/p}$ for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^m$ and $p > 0$. It is metric if $p \geq 1$. d_2 is the Euclidean distance, while d_1 is the city block distance. If d is isometrically embeddable into $(\mathbb{R}^m, d_1)$, then $d^{1/p}$ is embeddable into $(\mathbb{R}^m, d_p)$ for $1 \leq p \leq \infty$ [17].

Euclidean behavior [32]: A measure d on a set X has a Euclidean behavior if there exists a Euclidean space $(\mathbb{R}^m, d_2)$ such that (X, d) is isometrically embeddable into $(\mathbb{R}^m, d_2)$. If d has a Euclidean behavior, then so has $d^{1/r}$, $r \in (0, 1]$ [66].

Test for Euclidean behavior [32], [57]: Let $D^{\star 2} = (d_{ij}^2)$. A symmetric distance matrix D with a zero diagonal is Euclidean iff $S = -(1/2)(I - \mathbf{1}\mathbf{s}^\mathsf{T}) D^{\star 2} (I - \mathbf{s}\mathbf{1}^\mathsf{T})$ is psd for $\mathbf{s}^\mathsf{T}\mathbf{1} = 1$. Equivalently, D is Euclidean iff $D^{\star 2}$ is cnd. Note that $-D^{\star 2}$ is cpd, hence, an SVM kernel [12].

Similarity versus dissimilarity: Given a similarity measure $s : X \times X \to \mathbb{R}$, the corresponding d is defined as $d^2(x, y) = s(x, x) + s(y, y) - 2s(x, y)$. If $s(x, y)$ is psd, then s can be interpreted as a generalized inner product in a Hilbert space and $S = (s_{ij})$ is a psd kernel matrix. Moreover, d has a Euclidean behavior [32]. If $s(x, y) \in [0, 1]$, then d can be defined, e.g., as $d(x, y) = -\log(s(x, y))$ or $d(x, y) = (1 - s(x, y))^p$, $p = \{1, 1/2\}$.

Relations between psd and cnd kernels [2], [12]: Let K and D be real kernels and let $\sigma > 0$.

1) If K is psd, then $\tilde{K} = (e^{\sigma K_{ij}})$ is psd.
2) $D^{\star 2}$ is cnd iff $\tilde{K} = (e^{-\sigma\, dij^2})$ is psd.
3) $D^{\star 2}$ is cnd iff $\tilde{K} = (1/(\sigma + {d_{ij}}^2))$ is psd.

4) If $D^{\star 2}$ is cnd and $d_{ii}^2 \geq 0$ for all i, then $\tilde{K}_1 = (d_{ij}^{2r})$, $r \in (0,1)$, and $\tilde{K}_2 = (\log(1+d_{ij}^2))$ are cnd.

ACKNOWLEDGMENT

The authors would like to thank Prof. Jain, Dr. Zongker, Dr. Roth, and Dr. Paclík for providing dissimilarity data. The authors also thank anonymous reviewers for their good suggestions.

REFERENCES

[1] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2002.

[2] C. Berg, J. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups*. New York: Springer-Verlag, 1984.

[3] J. Bognár, *Indefinite Inner Product Spaces*. New York: Springer-Verlag, 1974.

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

[5] H. Bunke, S. Günter, and X. Jiang, "Towards bridging the gap between statistical and structural pattern recognition: Two new concepts in graph matching," in *Proc. Adv. Pattern Recognit.*, 2001, pp. 1–11.

[6] H. Bunke and A. Sanfeliu, Eds., *Syntactic and Structural Pattern Recognition Theory and Applications*. Singapore: World Scientific, 1990.

[7] S. Canu, C. Ong, and X. Mary, "Splines with non positive kernels," in *Proc. Int. ISAAC Congr*, 2005, pp. 1–10.

[8] A. Cayley, "On the theorem in the geometry of position," *Cambridge Math. J.*, vol. 2, pp. 267–271, 1841.

[9] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 790–799, Aug. 1995.

[10] F. Corpet, F. Servant, J. Gouzy, and D. Kahn, "ProDom and ProDom-CG: Tools for protein domain analysis and whole genome comparisons," *Nucleid Acids Res.*, vol. 28, pp. 267–269, 2000.

[11] P. Courrieu, "Straight monotonic embedding of data sets in Euclidean spaces," *Neural Netw.*, vol. 15, pp. 1185–1196, 2002.

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, UK: Cambridge Univ. Press, 2000.

[13] B. Dasarthy, Ed., *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. Los Alamitos, CA: IEEE Computer Society Press, 1991.

[14] D. de Ridder and R. Duin, "Sammon's mapping using neural networks: A comparison," *Pattern Recognit. Lett.*, vol. 18, no. 11–13, pp. 1307–1316, 1997.

[15] P. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[16] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

[17] M. M. Deza and M. Laurent, *Geometry of Cuts and Metrics*. New York: Springer-Verlag, 1997.

[18] C. Domeniconi, J. Peng, and D. Gunopulos, "Locally adaptive metric nearest-neighbor classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1281–1285, Sep. 2002.

[19] D. Donoho and C. Grimes, "Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data," in *Proc. PNAS*, 2003, vol. 100, pp. 5591–5596.

[20] M. Dubuisson and A. Jain, "Modified Hausdorff distance for object matching," in *Proc. Int. Conf. Pattern Recognit.*, 1994, vol. 1, pp. 566–568.

[21] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[22] R. Duin, D. de Ridder, and D. Tax, "Experiments with object based discriminant functions; a featureless approach to pattern recognition," *Pattern Recognit. Lett.*, vol. 18, no. 11–13, pp. 1159–1166, 1997.

[23] R. Duin and E. Pękalska, "The science of pattern recognition. Achievements and perspectives," in *Challenges for Computational Intelligence*, W. Duch and J. Mańdziuk, Eds. New York: Springer-Verlag, 2007, pp. 221–259.

[24] S. Edelman, S. Cutzu, and S. Duvdevani-Bar, "Representation is representation of similarities," *Behavioral Brain Sci.*, vol. 21, pp. 449–498, 1998.

[25] F. Esposito, D. Malerba, V. Tamma, H. Bock, and F. Lisi, "Similarity and dissimilarity," in *Analysis of Symbolic Data*. New York: Springer-Verlag, 2000.

[26] C. Faloutsos and K.-I. Lin, "FastMap: A fast algorithm for indexing, data-mining and visualization of tradditional and multimedia datasets," in *Proc. ACM SIGMOD*, 1995, pp. 163–174.

[27] L. Goldfarb, "A new approach to pattern recognition," in *Progress in Pattern Recognition,* vol. 2, L. Kanal and A. Rosenfeld, Eds. Amsterdam, The Netherlands: Elsevier, 1985, pp. 241–402.

[28] L. Goldfarb, D. Gay, O. Golubitsky, and D. Korkin, "What is a structural representation? A proposal for a representational formalism," Univ. New Brunswick, Fredericton, Canada, Tech. Rep., 2006.

[29] R. Goldstone and J. Son, "Similarity," in *Cambridge Handbook of Thinking and Reasoning*, K. Holyoak and R. Morrison, Eds. Cambridge, UK: Cambridge Univ. Press, 2005, pp. 13–36.

[30] J. Goodman and J. O'Rourke, in *Handbook of Discrete and Computational Geometry*, J. Goodman and J. O'Rourke, Eds. Boca Raton, FL: CRC Press, 2004.

[31] J. Gower, "Euclidean distance geometry," *Math. Sci.*, vol. 7, pp. 1–14, 1982.

[32] J. Gower, "Metric and Euclidean properties of dissimilarity coefficients," *J. Classification*, vol. 3, pp. 5–48, 1986.

[33] T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer, "Classification on pairwise proximity data," in *Proc. Adv. Neural Inf. Syst. Process*, 1999, pp. 438–444.

[34] 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 *Proc. Int. Conf. Artif. Neural Netw.*, 1999, pp. 304–309.

[35] B. Haasdonk, "Feature space interpretation of SVMs with indefinite kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 482–492, Apr. 2005.

[36] H. Haasdonk and B. Burkhardt, "Invariant kernels for pattern analysis and machine learning," *Mach. Learn.*, vol. 68, pp. 35–61, 2007.

[37] P. Hart, "The condensed nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 3, pp. 515–516, May 1968.

[38] T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification and regression," in *Advances in Neural Information Processing Systems,* vol. 8. Cambridge, MA: MIT Press, 1996, pp. 409–415.

[39] S. Hochreiter and K. Obermayer, "Support vector machines for dyadic data," *Neural Comput.*, vol. 18, no. 6, pp. 1472–1510, 2006.

[40] Y. Huang, C. Chiang, J. Shieh, and W. Grimson, "Prototype optimization for nearest-neighbor classification," *Pattern Recognit.*, vol. 35, no. 6, pp. 1237–1245, 2002.

[41] D. Jacobs, D. Weinshall, and Y. Gdalyahu, "Classification with non-metric distances: Image retrieval and class representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 583–600, Jun. 2000.

[42] A. Jain and D. Zongker, "Representation and recognition of handwritten digits using deformable templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 12, pp. 1386–1391, Dec. 1997.

[43] J. Laub and K.-R. Müller, "Feature discovery in non-metric pairwise data," *J. Mach. Learn. Res.*, vol. 5, pp. 801–818, 2004.

[44] C. Leslie, E. Eskin, A. Cohen, J. Weston, and W. Noble, "Mismatch string kernels for discriminative protein classification," *Bioinformatics*, vol. 20, no. 4, pp. 467–476, 2004.

[45] D. Lowe, "Similarity metric learning for a variable-kernel classifier," *Neural Comput.*, vol. 7, no. 1, pp. 72–85, 1995.

[46] M. Lozano, J. Sotoca, J. Sanchez, F. Pla, E. Pękalska, and R. Duin, "Experimental study on prototype optimization algorithms for prototype-based classification," *Pattern Recognit.*, vol. 39, pp. 1827–1838, 2006.

[47] F. Lu, S. Keles, Y. Lin, S. Wright, and G. Wahba, "Kernel regularization and dimension reduction," presented at the ASA Statistical Comput. Graph. Sections, Joint Statistical Meet, Seattle, WA, 2006.

[48] K. Menger, "New foundation of Euclidean geometry," *Amer. J. Math.*, vol. 53, pp. 721–745, 1931.

[49] H. Minh and T. Hofmann, "Learning over compact metric spaces," in *Proc. Conf. Learn. Theory*, 2004, pp. 239–254.

[50] R. Mollineda, F. Ferri, and E. Vidal, "An efficient prototype merging strategy for the condensed 1-NN rule through class-conditional hierarchical clustering," *Pattern Recognit.*, vol. 35, no. 12, pp. 2771–2782, 2002.

[51] V. Mottl, S. Dvoenko, O. Seredin, C. Kulikowski, and I. Muchnik, "Featureless regularized recognition of protein fold classes in a Hilbert space of pairwise alignment scores as inner products of amino acid sequences," *Pattern Recognit. Image Anal.*, vol. 11, no. 3, pp. 597–615, 2001.

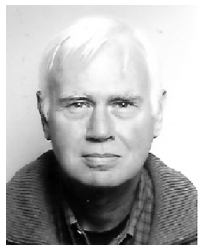
[52] C. Ong, X. Mary, S. Canu, and A. J. Smola, "Learning with non-positive kernels," in *Proc. Int. Conf. Mach. Learn*, 2004, pp. 639–646.

[53] P. Paclík and R. Duin, "Dissimilarity-based classification of spectra: Computational issues," *Real Time Imag.*, vol. 9, no. 4, pp. 237–244, 2003.

[54] P. Paclík, J. Novovičová, P. Somol, and P. Pudil, "Road sign classification using Laplace kernel classifier," *Pattern Recognit. Lett.*, vol. 21, no. 13–14, pp. 1165–1173, 2000.
[55] R. Paredes and E. Vidal, "A class-dependent weighted dissimilarity measure for nearest neighbor classification problems," *Pattern Recognit. Lett.*, vol. 21, no. 12, pp. 1027–1036, 2000.
[56] E. Pękalska, D. de Ridder, R. Duin, and M. Kraaijveld, "A new method of generalizing Sammon mapping with application to algorithm speed-up," in *Proc. Adv. School Comput. Imag.*, 1999, pp. 221–228.
[57] E. Pękalska and R. Duin, *The Dissimilarity Representation for Pattern Recognition. Foundations and Applications*. Singapore: World Scientific, 2005.
[58] E. Pękalska and R. Duin, "Dissimilarity-based classification for vectorial representations," in *Proc. Int. Conf. Pattern Recognit*, 2006, vol. 3, pp. 137–140.
[59] E. Pękalska, R. Duin, S. Günter, and H. Bunke, "On not making dissimilarities Euclidean," in *Proc. Joint IAPR Int. Workshops SSPR SPR*, 2004, pp. 1145–1154.
[60] E. Pękalska, R. Duin, and P. Paclík, "Prototype selection for dissimilarity-based classifiers," *Pattern Recognit.*, vol. 39, no. 2, pp. 189–208, 2006.
[61] E. Pękalska, P. Paclík, and R. Duin, "A generalized kernel approach to dissimilarity based classification," *J. Mach. Learn. Res.*, vol. 2, no. 2, pp. 175–211, 2002.
[62] V. Roth, J. Laub, J. Buhmann, and K.-R. Müller, "Going metric: Denoising pairwise data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 841–856.
[63] V. Roth, J. Laub, M. Kawanabe, and J. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 12, pp. 1540–1551, Dec. 2003.
[64] J. Rovnyak, "Methods of Krein space operator theory," *Oper. Theory: Adv. Appl.*, vol. 134, pp. 31–66, 2002.
[65] L. Saul and S. Roweis, "Think globally, fit locally: Unsupervised learning of low-dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, pp. 119–155, 2003.
[66] I. Schoenberg, "On certain metric spaces arising from Euclidean spaces by a change of metric and their imbedding in Hilbert space," *Ann. Math.*, vol. 38, pp. 787–797, 1937.
[67] B. Schölkopf, "The kernel trick for distances," in *Proc. Adv. Neural Inf. Process. Syst*, 2000, pp. 301–307.
[68] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, UK: Cambridge Univ. Press, 2004.
[69] P. Simard, Y. A. Le, Cun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition—Tangent distance and tangent propagation," *Int. J. Imag. Syst. Tech.*, vol. 11, no. 3, pp. 181–194, 2001.
[70] R. Sokal and P. Sneath, *Principles of Numerical Taxonomy*. San Francisco, CA: Freeman, 1963.
[71] A. Strehl and J. Ghosh, "Relationship-based clustering and visualization for high-dimensional data mining," *INFORMS J. Comput.*, vol. 15, no. 2, pp. 208–230, 2003.
[72] D. Tax and C. Veenman, "Tuning the hyperparameter of an AUC-optimized classifier," in *Proc. Belgian-Dutch Conf. Artif. Intell*, 2005, pp. 224–231.
[73] M. Tipping, "The relevance vector machine," presented at the Adv. Neural Inf. Process. Syst., San Mateo, CA, 2000.
[74] A. Tversky, "Features of similarity," *Psychol. Rev.*, vol. 84, no. 4, pp. 327–352, 1977.
[75] V. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.
[76] R. Veltkamp and M. Hagedoorn, "State-of-the-art in shape matching," in *Principles of Visual Information Retrieval*, M. Lew, Ed. New York: Springer-Verlag, 2001, pp. 87–119.
[77] U. von Luxburg and O. Bousquet, "Distance-based classification with Lipschitz functions," *J. Mach. Learn. Res.*, vol. 5, pp. 669–695, 2004.
[78] G. Wahba, "Support vector machines, reproducing kernel Hilbert spaces and the randomized GACV," in *Advances in Kernel Methods, Support Vector Learning*, B. Schölkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1999, pp. 69–88.
[79] C. Watkins, "Dynamic alignment kernels," in *Advances in Large Margin Classiers*, A. Smola, P. Bartlett, D. Scholköpf, and B. Schuurmans, Eds. Cambridge, MA: MIT Press, 2000, pp. 39–50.
[80] H. Zha and Z. Zhang, "Isometric embedding and continuum ISOMAP," in *Proc. Int. Conf. Mach. Learn.*, Washington, DC, 2003, pp. 864–871.

Elżbieta Pękalska received the M.Sc. degree in computer science from the University of Wrocław, Wrocław, Poland, in 1996, and the Ph.D. degree (*cum laude*) in computer science from the Delft University of Technology, Delft, The Netherlands, in 2005.

During 1998–2004, she was with Delft University of Technology, The Netherlands, where she worked on both fundamental and applied projects in pattern recognition. She is currently an Engineering and Physical Sciences Research Council Fellow at the University of Manchester, Manchester, U.K. She is engaged in learning processes and learning strategies, as well as the integration of bottom-up and top-down approaches, which not only includes intelligent learning from data and sensors, but also human learning on their development paths. She is the author or coauthor of more than 40 publications, including a book, journal articles, and international conference papers. Her current research interests focus on the issues of representation, generalization, combining paradigms, the use of kernels and proximity in the learning from examples, understanding brain research, neuroscience, and psychology.



Robert P. W. Duin (M'04) received the Ph.D. degree in applied physics from Delft University of Technology, Delft, The Netherlands, in 1978.

He studied applied physics at the Delft University of Technology, Delft, where he is currently an Associate Professor in the Faculty of Electrical Engineering, Mathematics and Computer Science. During 1980–1990, he studied and developed hardware architectures and software configurations for interactive image analysis. Then he was involved with pattern recognition via neural networks. He is currently an Advisory Editor of the *Pattern Recognition Letters*. His current research interests include design, evaluation, and application of algorithms that learn from examples, which includes neural network classifiers, support vector machines, classifier combining strategies, and one-class classifiers, especially complexity issues and the learning behavior of trainable systems receives much interest. Recently, he started to investigate alternative object representations for classification and thereby became interested in dissimilarity-based pattern recognition, trainable similarities, and the handling of non-Euclidean data.

Dr. Duin is a Fellow of the International Association for Pattern Recognition (IAPR). He was an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. He was the recipient of the Pierre Devijver Award for his contributions to statistical pattern recognition in August 2006.