# Possibilities of zero-error recognition by dissimilarity representations

Robert P.W. Duin and Elżbieta Pękalska

Pattern Recognition Group, Department of Applied Physics
Delft University of Technology, The Netherlands
{duin,ela}@ph.tn.tudelft.nl

**Abstract**.
Feature based approaches to pattern recognition suffer from the fact that feature representations of different classes of objects may overlap. This is the consequence of reducing the description of an object to a feature vector. As a result an error free recognition system is even asymptotically (for infinite training sizes) impossible. In this paper it is argued that this limitation does not hold for dissimilarity based representations. Suggestions are made how this may be exploited in practice.

## 1 Introduction

Template matching is a simple and early approach to pattern recognition. Objects are classified on the basis of distances to prototypes. Often, just a single prototype per class is selected. This approach, however, does not take into account the distribution of the object variations. Therefore, it puts high demands on the possibilities to normalize objects such that an accurate matching can be realized.

Feature-based classifiers are more advanced. In this case, objects are described by carefully selected properties (measurements) which are combined in feature vectors. Such features build a feature space. A classifier can be trained in this space, using a set of examples reflecting the variations in the classes to be distinguished. Usually, the features are a reduced description of the objects. Some information is lost and, as a consequence, essentially different objects may be represented by the same point in the feature space. If this happens for objects of different classes, these classes overlap. There is no way to distinguish such objects in the feature space and, thereby, any recognition scheme based on such a feature representation has a non-zero classification error.

A different, not yet very intensively studied approach is to use distances between objects, like in template matching, but now making use of the variations in distances between the objects in the training set. So, objects are not represented by features, but by distances or dissimilarities directly measured on the raw data. The loss of information by the reduction to features, that was just mentioned, may hereby be avoided.

It is the purpose of this paper to clarify the consequences for the class overlap of such a dissimilarity representation. We will argue that under some circumstances this overlap can be entirely avoided, resulting in a zero-error classification. Let us first introduce the dissimilarity representation further.

The starting point of this approach is a given set of dissimilarities between a training set of objects of known class memberships (their labels) and a selected set of prototypes. These prototypes may be selected by some expert, or may be the result of

an automatic selection procedure [7], [8]. Classification is usually done by the nearest neighbor rule (or in this context by the nearest prototype rule). Sometimes more advanced schemes are used, see for instance [14], [15], [16].

An intriguing point in relation with the nearest neighbor rule is that it is usually not trained. Just the distances to the prototypes are used for classifying new objects. The set of distances between the training set and the prototypes themselves is information that is sometimes used in the selection of prototypes, but is often completely neglected by using this classification rule. Recently, we have shown that these distances may be used by the so called dissimilarity-based classifiers [2], [3], [17]. They may not only demand less prototypes and, thereby, the computation of less distances, but, at the same time, perform significantly better than the nearest neighbor rule. In section 2, an overview of the various approaches to build classifiers from a given dissimilarity representation is presented.

As mentioned above, under some conditions, the class overlap related to the use of feature spaces can be avoided by the use of dissimilarities. The question to be discussed in section 3 is whether we can build classifiers that exploit this in practice. In other words, whether we can construct classifiers that have asymptotically (for increasing training set sizes) a zero classification error. In section 4, a few examples will be given and in the last section conclusions are summarized.

## 2 Classification approaches to dissimilarity descriptions

Here we will present a summary of the various approaches that can be used for building classifiers from a dissimilarity matrix of the training data. This section is based on, and makes use of some of, our previous papers, [1]-[5].

To construct a classifier on dissimilarities, the training set $T$ of size $n$ (having $n$ objects) and the representation set $R$ [5] of size $r$ will be used. $R$ is a set of prototypes covering all classes present. $R$ is chosen here to be a subset of $T$ ($R \subseteq T$), although, in general, $R$ and $T$ might be disjunct. In the learning process, a classifier is built on the $n \times r$ distance matrix $D(T, R)$, relating all training objects to all prototypes. The information on a set $S$ of $s$ new objects is provided in terms of their distances to $R$, i.e. as an $s \times r$ matrix $D(S, R)$.

The following three different approaches to classification based on distances are distinguished:

1. The nearest neighbor rule, finding the smallest distances in $D(S, R)$.
2. Dissimilarity representations, directly using the distances of $D(T, R)$ to build an $r$-dimensional space. In this space, the $n$ training objects, represented by their distances to the objects of R, are used for training a classifier.
3. Embedding, i.e. constructing a new space $\Re$ in which the Euclidean distances between the training objects correspond as well as possible to the given dissimilarities $D(R, R)$.

### 2.1 Nearest neighbor method

A straightforward approach to dissimilarity representations leads to the k-Nearest-Neighbor ($k$-NN) rule [9],[11]. Such classifiers make use of distance information in a rank-based way. The NN rule, in its simplest form, i.e. 1-NN rule, assigns a new object to the class of its nearest neighbor from the representation set $R$ by finding minima in the rows of $D(S, R)$. The $k$-NN decision rule is based on majority voting: an unknown

object becomes a member of the class the most frequently occurring among the *k* nearest neighbors. The 1-NN rule does not make any use of the available information in the dissimilarity matrix *D(T,R)*, it directly operates on *D(S,R)*. One may expect, therefore, that improvements are possible. On the other hand, the asymptotic error of this rule for $r \to \infty$ is $2\varepsilon^*(1-\varepsilon^*)$, in which $\varepsilon^*$ is the Bayes error. So, for non-overlapping classes ($\varepsilon^* = 0$), the error of the 1-NN rule is also zero. This may be, however, unfeasible, since it may demand the storage and handling of an infinite training set.

### 2.2 Linear/quadratic dissimilarity classifiers

This approach relies on interpreting distances as a representation of a dissimilarity space. In particular, $D(T, R)$ is treated as a description of a space where each dimension corresponds to the distance to a prototype. The prototypes constitute, thereby, an *r*-dimensional dissimilarity space. In general, $D(x, R)$ defines a vector consisting of *r* distances found between the object *x* and all the objects in the representation set *R*, i.e. if $R = \{p_1, ..., p_r\}$, then $D(x, R) = [D(x, p_1), ..., D(x, p_r)]$. Therefore, $D(\bullet, R)$ is seen as a mapping on an *r*-dimensional dissimilarity space. In this convention, neither *x* nor *R* refers to points in a feature space, instead they refer to the objects themselves. The advantage of such a representation is that any traditional classifier operating on feature spaces can be used. Moreover, it can be optimized by using training sets larger than the given representation set. This does not complicate the decision rule, but does increase its accuracy.

The choice of Bayesian classifiers [11] assuming normal distributions, is a natural consequence of the central limit theorem applied to dissimilarities. It is supported by the observation that most of the commonly-used dissimilarity measures, e.g. Euclidean distance or Hamming distance, are based on sums of differences between measurements. The central limit theorem states that the sum of random variables tends to be normally distributed in the limit, provided that none of the variances of the sum's components dominates. Therefore, summation-based distances (built from many components) tend to be approximately normally distributed, which suggests that Bayesian classifiers, and also Fisher's linear discriminant [11],[12], should perform well in dissimilarity spaces. A problem arises for representation set sizes that are almost equal to the size of the training set. In these cases it is needed to estimate the normal densities from *D(T,R)* by using robust estimators, e.g. based on regularization [12].

### 2.3 Linear embedding of dissimilarities

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

#### 2.3.1 Embedding of Euclidean distances

Let the representation set $R = \{p_1, p_2, \ldots, p_r\}$ refer to *r* objects. Given an Euclidean distance matrix $D \in \Re^{r \times r}$ between those objects, a distance preserving mapping on an Euclidean space can be found. Such a projection is known in the literature as classical scaling or metric multidimensional scaling [20]. In other words, the dimensionality $k \leq r$ and the configuration $D \in \Re^{r \times k}$ can be found such that the (squared) Euclidean distances are preserved. Note that having determined one configuration, another one can

be found by a rotation or a translation. To remove the last degree of freedom, without loss of generality, the mapping will be constructed such that the origin coincides with the centroid (i.e. the mean vector) of the configuration *X*.

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

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

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

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

where *J* is the centering matrix $J = I - \frac{1}{r}\mathbf{1}\mathbf{1}^T \in \Re^{r\times r}$ and *I* is the identity matrix. *J* projects the data such that the final configuration has zero mean. *B* is positive definite since it is a Gram matrix [21]. Then, the factorization of *B* by its eigendecomposition can be found as:

$$XX^T = B = Q\Lambda Q^T \tag{3}$$

where Λ is a diagonal matrix of the first non-negative eigenvalues, ranked in descending order, followed by the zero values, and Q is an orthogonal matrix of the corresponding eigenvectors [20]. For $k < r$ non-zero eigenvalues, a *k*-dimensional representation *X* can be then found as:

$$X = Q_k\Lambda_k^{\frac{1}{2}},\ Q_k \in \Re^{r\times k},\ \Lambda_k^{\frac{1}{2}} \in \Re^{k\times k} \tag{4}$$

where $Q_k$ is a matrix of the first *k* leading eigenvectors and $\Lambda_k^{\frac{1}{2}}$ contains the square roots of the corresponding eigenvalues. Note that X, determined in this procedure, is unique up to rotation (the centroid is now fixed), since for any orthogonal matrix *T*, $XX^T = (XT)(XT)^T$. Note also that *X* is an uncorrelated representation, i.e. given w.r.t. the principal axes.

### 2.3.2 Linear embedding of non-metric dissimilarity data

Non-metric distances may arise when shapes or objects in images are compared e.g. by template matching [13],[14]. For projection purposes, the symmetry condition is necessary, but for any symmetric distance matrix, an Euclidean space is not 'large enough' for a distance-preserving linear mapping onto the specified dimensionality. It is, however, always possible [18] for a pseudo-Euclidean space.

A pseudo-Euclidean space $\Re^{(p,q)}$ of the signature $(p,q)$ [18] is a real linear vector space of dimension $p+q$, composed of two Euclidean subspaces, $\Re^p$ and $\Re^q$, such that $\Re^{(p,q)} = \Re^p \oplus \Re^q$ and the inner product $\langle .,. \rangle$ is positive definite on $\Re^p$ and negative definite on $\Re^q$. The inner product w.r.t. the orthonormal basis is defined as

$$\langle x, y \rangle = \sum_{i=1}^{p} x_i y_i - \sum_{j=p+1}^{p+q} x_i y_i = x^T M y \tag{5}$$

with

$$M = \begin{bmatrix} I_{p\times p} & 0 \\ 0 & -I_{q\times q} \end{bmatrix} \tag{6}$$

where $I$ is the identity matrix. Using the notion of inner product,

$$d^2(x,y) = \|x-y\|^2 = \langle x-y, x-y\rangle = (x-y)^T M(x-y) \tag{7}$$

can be positive, negative or zero. Note that an Euclidean space $\Re^p$, is a pseudo-Euclidean space $\Re^{(p,0)}$.

The matrix $B = -\frac{1}{2}JD^{(2)}J$ is positive definite if and only if the distance matrix $D \in \Re^{r\times r}$ is Euclidean [20]. Therefore, for a non-Euclidean $D$, $B$ is not positive definite, i.e. $B$ has *negative* eigenvalues. As a result, $X$ cannot be constructed from $B$, since it relies on the square roots of eigenvalues. However, it is possible to use a pseudo-Euclidean space.

To embed the data, the same reasoning as for an Euclidean space is applied here. The essential difference refers to the notion of an inner product and a distance. Now, $B = -\frac{1}{2}JD^{(2)}J$, is still the matrix of inner products, but it is expressed as

$$B = XMX^T \tag{8}$$

where $M$ is a matrix of the inner product operation in a pseudo-Euclidean space. Following [19], we can write (compare to equation (3)):

$$XMX^T = B = Q\Lambda Q^T = Q|\Lambda|^{\frac{1}{2}} \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} |\Lambda|^{\frac{1}{2}} Q^T \tag{9}$$

where M is given by () and $p + q = k$. $\Lambda$ is now based on $p$ positive and $q$ negative eigenvalues, which are presented in the following order: first, positive eigenvalues with decreasing values, then negative ones with decreasing magnitude and finally, zero values. Therefore, $X$ can be now represented in a pseudo-Euclidean space $\Re^k$=$\Re^{(p,q)}$ (see [18]), as follows:

$$X = Q_k|\Lambda_k|^{\frac{1}{2}} \tag{10}$$

### 2.3.3 Projection of new points

Having found a configuration $X$ in a pseudo-Euclidean space that preserves all pair wise distances *D(R,R)*, new objects can be can added to this space via the linear projection. Given the distance matrix $D_s^{(2)} \in \Re^{s\times r}$, expressing dissimilarities between $s$ novel objects and all objects of the representation set $R$, a configuration $X_S$ is to be determined in a pseudo-Euclidean space $\Re^k$=$\Re^{(p,q)}$. First, the matrix $B_s \in \Re^{s\times r}$ of inner products relating all new objects to all objects from $R$ should be found, which becomes:

$$B_s = -\frac{1}{2}(D_s^{(2)}J - UD^{(2)}J) \tag{11}$$

where $J$ is the centering matrix and $U = \frac{1}{s}\mathbf{1}\mathbf{1}^T \in \Re^{s\times r}$. Since $B_s$ can be expressed as:

$$X_s MX^T = B_s \tag{12}$$

with $M = I \in \Re^{k\times k}$ if $\Re^k$ is Euclidean, or M defined by (6) if $\Re^k$ is pseudo-Euclidean,

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

$$X_s = B_s X |\Lambda|^{-1} M \text{ or } X_s = B_s Q_k |\Lambda_k|^{-\frac{1}{2}} M \tag{13}$$

**2.3.4 Reduction of dimensionality**

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

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

**2.3.5 Classifiers in the reduced embedded space**

For a pseudo-Euclidean configuration, a linear classifier $f(x) = \langle v,x \rangle + v_0 = v^T M x + v_0$ can be constructed by addressing it as in the Euclidean case, i.e $f(x) = \langle w,x \rangle_{Eucl} + v_0 = w^T x + v_0$, where $w = Mv$ (see [18],[4]).

**2.4 Example of dissimilarity based classifiers.**

The following example illustrates the use and benefits of dissimilarity-based classifiers over the direct use of the 1-NN rule. We used two datasets, one real, based on digit recognition, obtained from Jain and Zongker [15], and one based on artificially generated polygons.

For the 10-class digit recognition problem we used a 2000x2000 dissimilarity matrix computed by Zongker [15] using deformable templates. The polygon dataset is similar to the one described in section 4. It has two classes and is given by a 2000x2000 matrix of Hausdorff distances [13] computed for the polygon corners. In both cases, training sets $T$ of 1500 objects were used and a test set $S$ of 500 objects. For growing representation sets $R \subset T$, the following classifiers are built on $D(R,R)$ and tested on $D(S,R)$:

1. The regularized Bayes linear discriminant assuming normal densities (RLDC) on the dissimilarity representation $D(R,R)$, see section 2.2.

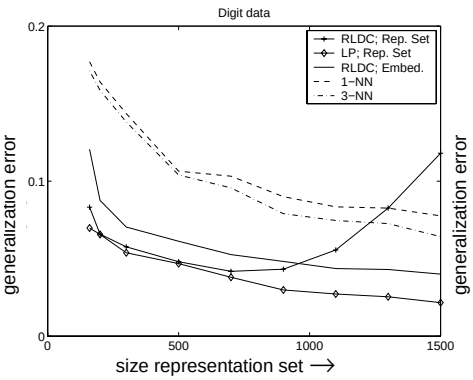



*Fig. 1 Dissimilarity based classifiers compared with the nearest neighbor rule for a two-class digit classification problem.*

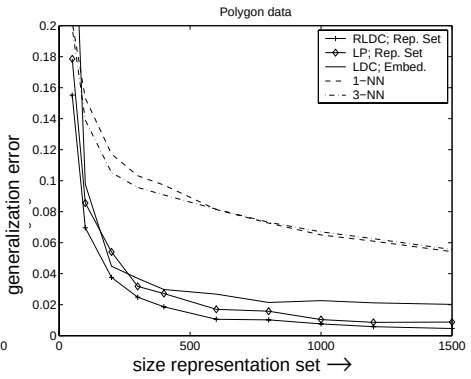



*Fig. 2 Dissimilarity based classifiers compared with the nearest neighbor rule for a two-class polygon classification problem.*

2. A linear programming optimizer (LP) for *D(R,R)*. Since this matrix is square, there are as many points as dimensions, so there exists a perfect, error-free solution for *R*, [10].
3. The regularized Bayes linear discriminant assuming normal densities (RLDC) on an embedded space of fixed dimensionality, see section 2.3. For the digit data this dimensionality was set to 100, for the polygon data we used 45.

The results are shown in figure 1and figure 2. For comparison, the test results of the 1-NN and 3-NN classifiers are shown as well. (The 3-NN seems to be best over for the k-NN rule over a range of k=1, ..., 15). These pictures make clear that the dissimilarity based classifiers may perform very well. The increasing error for RLDC in the digit classification problem is due to bad (i.e. constant) regularization as we did not optimize that in this experiment.

**3 The asymptotic separability of classes**

If for the dissimilarities holds that they can only be zero if and only if the corresponding objects are identical, then class overlap is avoided if objects are allowed to belong to one class only. As stated in section 2.1, this implies that the 1-NN rule will constitute a zero-error classifier. It may demand, however a very large training set. As the dissimilarity-based classifiers (section 2.2) and the embedding classifiers (section 2.3) appear to be much more efficient in the requirement of storing a training set (the representation sets needed to be stored by them are much smaller than the training sets needed by the 1-NN rule), the question arises whether these classifiers may also have an asymptotic zero error.

The following discussion is based on a set of assumptions. They are often fulfilled, or can easily be fulfilled:

1. The real, physical object classes are separable, i.e. there is no physical object that is a member of more than one class. For optical character recognition this implies for instance that there is still a difference between the characters ‘0’ (zero) and ‘O’ (the letter ‘O’)

2. The raw measurements of the objects are made such that this separability is maintained. One way to inspect this is to have the objects labeled by human analysts from the measurements (e.g. a video screen that displays the object image to be used for further processing). It is fulfilled if the characters can still be labeled correctly after scanning and display.
3. For a distance measure *D(x,y)* between objects *x* and *y* represented by their raw measurements (e.g. scanned images) holds that *D(x,x)* = 0 and *D(x,y)* > δ > 0 if *x* and *y* belong to different classes. This assumption states that there is some 'gap' between the classes of size δ: Objects of different classes have a distance of at least δ.
4. The raw measurement of any object includes *just* continuous noise (e.g. changing lighting conditions, small rotations or sensor deviations). This noise is such that for any two measurements *x* and *y* of the same physical object holds that *D(x,y)* < δ.
5. The digitalization of the measurements and thereby the computer representation of the objects is such that the minimum class gap δ is preserved.

Since the digitized world is finite, there is a finite probability that in the δ-environment of an object *x* there is another object *y* of the same class:

$$\mathrm{Prob}(y|(D(x,y)<\delta, x\in\omega, y\in\omega))>\varepsilon>0 \tag{14}$$

Because of assumption number 3 there is no object in that neighborhood of another class:

$$\mathrm{Prob}(y|(D(x,y)<\delta, x\in\omega, y\notin\omega)) = 0 \tag{15}$$

From this, it can be concluded that with probability one an infinitely growing training set will contain an object $y \in \omega$ within a δ-environment of a given test object $x \in \omega$. Therefore, *D(x,y)* is smaller than the distance to all objects $z \notin \omega$. Consequently, any object *x* will be asymptotically correctly classified by the nearest neighbor rule.

As long as we are able to correctly label the objects from the digitized measurements and we have a distance measure between objects for which holds that *D(x,y)* = 0 for *x* = *y*, the nearest neighbor rule has this asymptotic property. However, this is not a feasible approach, since it implies the storage of an almost infinitely growing set of objects.

In the previous section, we showed that dissimilarity based classifiers may generalize better and may need smaller representation sets then the nearest neighbor rule. This will depend on how much 'space' there is between the classes, i.e. the smallest distances between objects of different classes. If the size of this gap is sufficient then a classifier defined by a subset of the data might fit into it. It depends on t the definition of the distance measure whether this can be achieved by a linear classifier.

The two classifiers discussed in the sections 2.2 and 2.3 depend on a representation set and a training set. The necessary size of the representation set is determined by the complexity of the problem, i.e. the nonlinearity. For smaller gaps between the classes, larger training set sizes may be needed to position the classifiers more accurately. In the next section we will experimentally investigate this further.

*Fig. 3 Objects misclassified by the nearest neighbor rule (top), their nearest neighbor '3' and their nearest neighbor '8'.*

**4 Some experiments**

The experiments in this section are based on two datasets, one artificial and one real. The real world dataset consists of the digits '3' and '8' of the NIST character database [22]. We computed the Hamming difference between 32x32 sampled versions of the digits. The first question that arises is whether the dataset fulfills the assumptions formulated in section 3. We checked all the nearest neighbor relations. In figure 3, the objects are shown that are misclassified by the nearest neighbor rule together with their nearest neighbors in both classes. For some objects, it may be concluded that they are badly segmented as they contain isolated dots. As a consequence, they do not fulfill assumption 4 in section 3. Object representations based on segmentation errors are not expected to have close neighbors. In a practical situation, they may be removed from the training set. New objects, having such defects, are thereby expected to be misclassified.

In figure 4, the distances to the nearest neighbors in the dataset are shown for a fraction of the data. In a very few cases the nearest neighbor belongs to a different class. This causes a classification error. The total error for this set, using the leave-one-out error estimate, is 0.0185. The figure clearly suggests that there is a gap between the classes. In the following experiments, we try to construct some classifiers in this area. We use a fixed training set of 2x500 objects. The remaining 2x500 objects are used for testing.

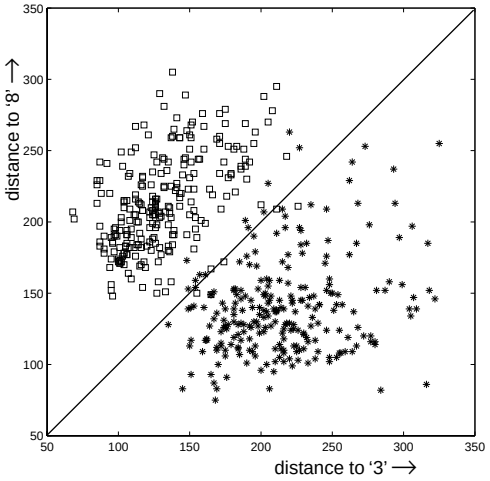
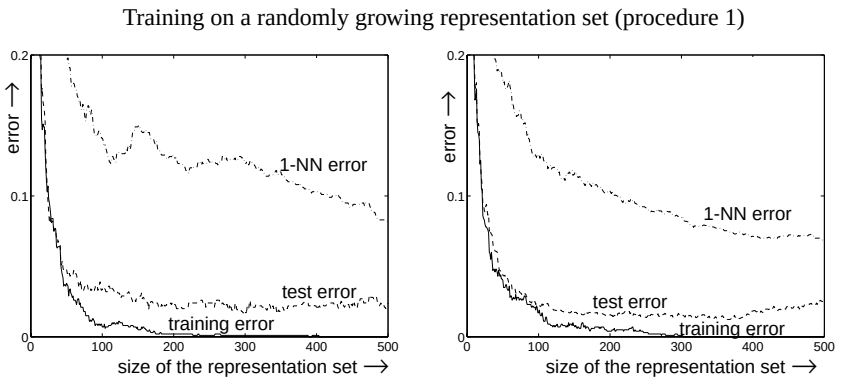
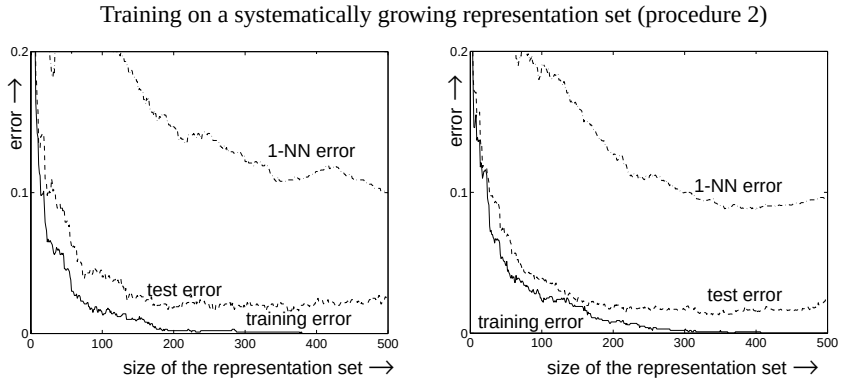



*Fig. 4 Scatter plot of distances to the nearest neighbors of both classes.*

1. Dissimilarity-based classification by Fisher's linear classifier using a randomly selected representation set, see section 2.2.
2. Dissimilarity-based classification by Fisher's linear classifier based on a systematically selected representation set, see section 2.2. Starting from a few objects in the set R, this selection is done in an iterative procedure. In each step, the classifier is trained and the training object that is the closest to the decision boundary is added to the representation set.
3. Embedding-based classification by Fisher's linear classifier defined by an interactively growing representation set as above. For the construction of the embedded space we used the eigenvectors corresponding to the largest eigenvalues, jointly explaining 70% of the variance.



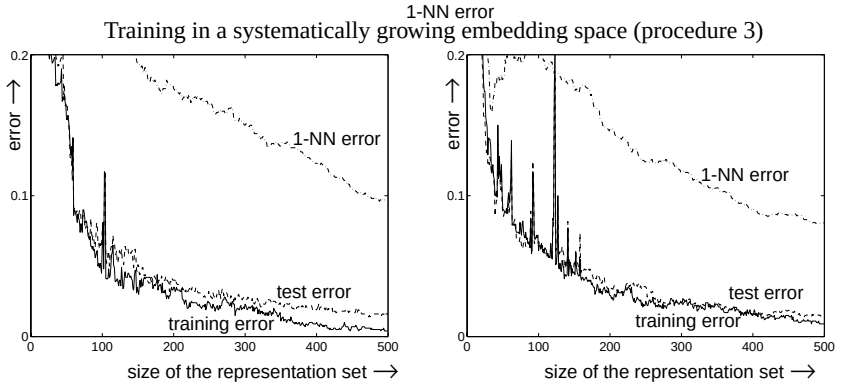




*Fig. 5 The errors of the training set and the test set as a function of the size of the representation set for the NIST digits '3' and '8' represented by their Hamming distances. The 1-NN error on the representations set is given as a reference.*

*Fig. 6 The errors of the training set and the test set as a function of the size of the representation set for two sets of polygons represented by their Hausdorff distances. The 1-NN error on the representations set is given as a reference.*

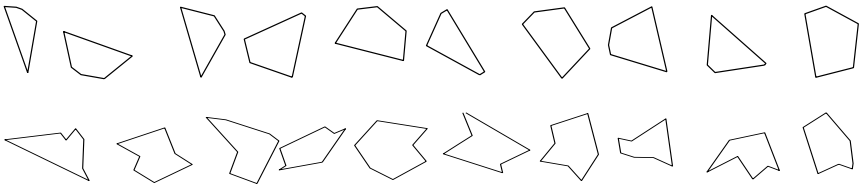


*Fig. 7 Some examples of the two classes of polygons generated. Upper row, polygons with 5 corners, f on the unit circle. Bottom row, polygons with 7 corners, having a standard deviation of 0.4 from the unit circle.*

In order to achieve a better generalization, *D(T,R)* was used for training all these classifiers instead of using only *D(R,R)* as in section 2.4. In figure 5, the errors on the training set and test set are shown as a function of the size of the representation set. The test errors for the 1-NN rule on the representation set are shown for comparison. This figure shows that we can construct a zero-error classifier for the training set, but that it appeared impossible to get this result also for the test set. This might be caused by the fact that this real world example did not fulfill our assumption number 4. Note the instability of the results for the embedding procedure (bottom figure) in case of small sizes of the representation set. It should also be noted that a systematic selection of objects for the representation set is not better at all than a random selection. This is in agreement with previous results [1], [3].

In an attempt to verify the statements in section 3 we constructed an artificial dataset based on polygons. Two classes, of 2000 objects each, are generated, one class with 5 corners on the unit circle and one with 7 corners, deviating from that circle by a standard deviation of 0.4. In figure 5 some examples are shown. In order to guarantee that these classes are really separable we made sure that the shortest edges of the 7-corner polygons are longer than 0.2. Thereby, they cannot degenerate to a 5-corner polygon. This determines the gap we discussed in section 3.

The Hausdorff distances [13] between polygon corners have been computed to build the dissimilarity representation. 500 objects per class are used for training and the remaining 1500 objects per class for testing. In its entirety we fulfilled the assumptions as discussed in section 3. The results in figure 6 show, however, that we did not reach our goal to construct a zero-error classifier for this problem. The error curves even look very similar to those of the digit recognition problem in figure 5, in spite of the fact that the polygon problem is intuitively much simpler and has definitely no class overlap.

In order to investigate the dependency of the results for the distance measure we computed for the same set of polygons the Modified Hausdorff distance. This distance measure is not metric but has to be preferred for a better class separability [13]. Instead of an Euclidean space, now a pseudo-Euclidean space has to be used for embedding, see section 2.3. The results, shown in figure 8, are much better than for the original Hausdorff distance (figure 6). Note, however, the difference in scale. Also in this case the error seems to be asymptotically constant and we fail to find a zero-error for large sizes of the representation set.

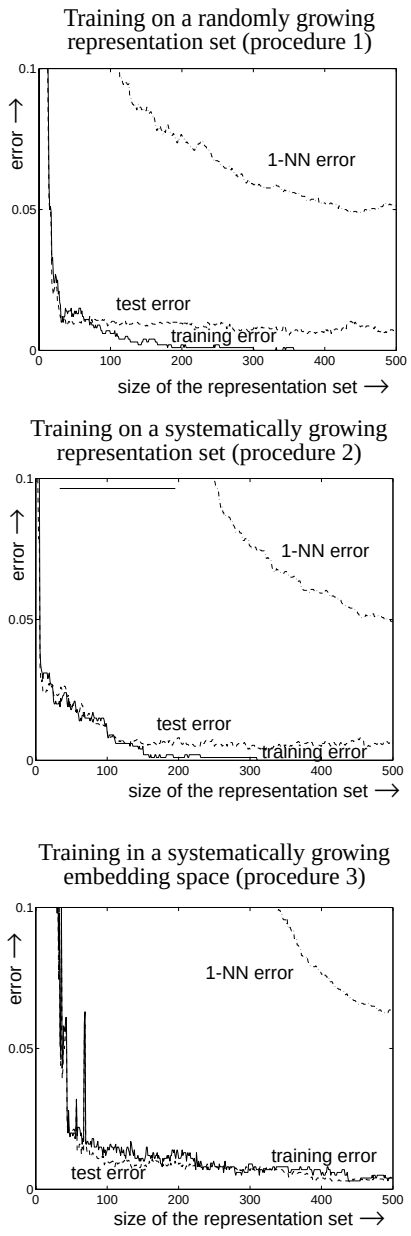



*Fig. 8 The errors on the training set and the test set as a function of the size of the representation set for two sets of polygons represented by their Modified Hausdorff distances. The 1-NN error on the representations set is given as a reference.*

## 5 Discussion and conclusions

The overlap of pattern classes may be avoided by a dissimilarity based representation constructed from the raw data if the assumptions as listed in section 3 are fulfilled. We showed that linear classifiers built for such representations can outperform the nearest neighbor rule, even for large training set sizes for which a good performance of the NN-rule may be expected. Although the classes are truly separable, we did not succeed in our attempts to construct a zero-error solution. This result certainly depends on the distance measure that is used in relation to the classifier. The linear classifier is able to separate large training sets (of 1000 objects in total), even in about 300 dimensions, but this does not generalize for the test set. Our blame is that the Fisher discriminant in combination with the dissimilarity representation is global sensitive: remote objects, having large distances, influence their exact position.

The challenge, we see for the future, is to construct more locally sensitive classifiers that still need just a fraction of the training set for representation. Further research is, therefore, needed to find out how distance measures may be constructed such that the potentially zero-error result can be obtained in practice.

## 6 Acknowledgments

This research was supported by the Dutch Organization for Scientific Research (NWO). The authors thank Professor A.K. Jain and Dr. D. Zongker of Michigan State University for supplying one of the datasets.

## 7 References

[1]R.P.W. Duin, E. Pekalska, and D. de Ridder, Relational discriminant analysis, *Pattern Recognition Letters*, vol. 20, no. 11-13, 1999, 1175-1181.

[2]E. Pekalska and R.P.W. Duin, Automatic pattern recognition by similarity representations - a novel approach, *Electronic Letters*, vol. 37, no. 3, 2001, 159-160.

[3]E. Pekalska and R.P.W. Duin, Classifiers for dissimilarity-based pattern recognition, in:

A. Sanfeliu, J.J. Villanueva, M. Vanrell, R. Alquezar, A.K. Jain, J. Kittler (eds.), *ICPR15, Proc. 15th Int. Conference on Pattern Recognition*, vol. 2, IEEE Computer Society Press, Los Alamitos, 2000, 12-16.

[4]E. Pekalska, P. Paclik, and R.P.W. Duin, A Generalized Kernel Approach to Dissimilarity-based Classification, *Journal of Machine Learning Research*, Special Issue on Kernel Methods, vol. 2, no. 2, 2001, 175-211.

[5]R.P.W. Duin, Classifiers in Almost Empty Spaces, in: A. Sanfeliu, J.J. Villanueva, M. Vanrell, R. Alquezar, A.K. Jain, J. Kittler (eds. ) *ICPR15, Proc. 15th Int. Conf. on Pattern Recognition* , vol. 2, IEEE Comp. Soc. Press, Los Alamitos, 2000, 1-7.

[6]A.K. Jain, R.P.W. Duin, and J. Mao, Statistical Pattern Recognition: A Review, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, 2000, 4-37.

[7]J.S. Sanchez, F. Pla, and F.J. Ferri, Prototype selection for the nearest neighbor rule through proximity graphs, *Pattern Recognition Letters*, vol. 18, 1997, 507-513.

[8]Uri Lipowezky, Selection of the optimal prototype subset for 1-NN classification, *Pattern Recognition Letters*, vol. 19, no. 10, 1998, 907-918.

[9]T.M. Cover and P.E. Hart, Nearest neighbor pattern classification, *IEEE Trans.Info.Theory*, vol. IT-13, 1967, 21-27.

[10]T.M. Cover, Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition, IEEE *Transactions on Electronic Computers*, vol. EC-14, June 1965, 326-334.

[11]K. Fukunaga, *Introduction to statistical pattern recognition*, Academic Press,1990.

[12]B. Ripley, *Pattern Recognition and Neural Networks*, Cambridge University Press, Cambridge, 1996.

[13]M.-P. Dubuisson and A.K. Jain, Modified Hausdorff distance for object matching, Proc. *12th IAPR International Conference on Pattern Recognition* (Jerusalem, October 9-13, 1994), vol. 1, IEEE, Piscataway, NJ, USA,94CH3440-5, 1994, 566-568.

[14]D.W. Jacobs, D. Weinshall, and Y. Gdalyahu, Classification with Nonmetric Distances: Image Retrieval and Class Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 6, 2000, 583-600

[15]A.K. Jain and D. Zongker, Representation and recognition of handwritten digits using deformable templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 12, 1997, 1386-1391.

[16]S. Edelman, *Representation and Recognition in Vision*, MIT Press, 1999.

[17]S. D. Connell and A. K. Jain, Template-based online character recognition, *Pattern Recognition*, vol. 34, no. 1, 2001, 1-14.

[18]L. Goldfarb, A unified approach to pattern recognition, *Pattern Recognition*, vol. 17, 1984, 575-582.

[19]L. Goldfarb, A New Approach to Pattern Recognition, in: Kanal, L.N. and Rosenfeld, A. (eds.), *Progress in Pattern Recognition*, vol. 2,, Elsevier Science Publishers B.V., 1985, 241-402.

[20]I. Borg and P. Groenen, *Modern Multidimensional Scaling*, Springer Verlag, Berlin, 1997.

[21]W. Greub, *Linear Algebra*, Springer-Verlag, 1975.

[22]C.L. Wilson, M.D. Marris, *Handprinted character database 2*, april 1990. National Institute of Standards and Technology; Advanced Systems division.