Classification on dissimilarity data: a first look

Elżbieta Pękalska and Robert P.W. Duin

Pattern Recognition Group, Department of Applied Physics, Faculty of Applied Sciences,
Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands
email: ela@ph.tn.tudelft.nl

Keywords: dissimilarity representation, feature-based classifier, rank-based classifier, normal-based classifier, support vector classifier, normal distribution, central limit theorem

Abstract

In a dissimilarity (distance) data each pair of objects is characterized by a value which expresses the magnitude of difference between them. This type of data can be now classified using various approaches, provided that a new object is represented by its distances to the training samples.

This paper discusses a number of possibilities to tackle such a classification problem. Two types of methods are investigated: the feature-based (i.e. interpreting the distance data as a feature space) and rank-based decision rules. Experiments conducted on real datasets demonstrate that the feature-based classifiers often outperform the rank-based ones. The normal-based decision rules perform well, since summation-based distances (frequently appearing in practice) are, under general conditions, approximately normally distributed. In addition, also the support vector classifier achieves a high accuracy, particularly in distance spaces of a very high dimensionality.

1 Introduction

In the traditional approach to pattern recognition objects are represented in a feature space. However, this is not the only space where classification problems can be tackled. Alternative ways can be found by constructing decision rules on dissimilarity representations. Such representations become an option when the original data consists of a large set of attributes, originating either from a homogeneous source (e.g. spectra bands, images) or being a collection of heterogeneous characteristics.

For some applications it can be also easier or more natural to define a distance or similarity between objects than to formulate the features explicitly. Let us assume e.g. a syntactic way of treating patterns, in which a distance measure is defined as the smallest number of the operations necessary to transform one pattern into another [7]. This introduces a transformation system consisting of the structured objects (pattern representatives) and a finite set of modification operators. A concrete example is a system, which describes strings over a finite alphabet, and the substitution operations are some variants of deletion and insertion. It is also possible to assign non-negative weights to the particular operations, emphasizing those which refer to rare phenomena, as well. Chromosome classification is an application of such an approach. Each chromosome is composed of some primitives, where the fixed set of primitives defines our alphabet. The dissimilarity measure should now reflect how much the chromosomes differ. The number of necessary substitutions in order to transform one chromosome into another is such an intuitive value.

There exist many ways of defining a dissimilarity measure and this confirms the usefulness of studying recognition problems on distance representations. They can offer an alternative approach to pixel based image recognition, as well.

This paper investigates methods for building classifiers on dissimilarity representations. A number of approaches is introduced in section 2, which gives also some insight into the types of distance distributions and their consequences for classifiers. The experiments are performed on real datasets which alongside with the experimental set-up are described in section 3. The results are discussed in section 4 and the conclusions are summarized in section 5.

2 Dissimilarity-based pattern recognition

Dissimilarity measures differ according to various datasets or applications. It is assumed that the distances are non-negative and two objects have a zero distance only when they are identical. This allows us for defining classifiers on dissimilarity representations.

A straightforward way of dealing with such a problem is based on the distance relation between objects, which naturally leads to the rank-based methods, e.g. to the nearest neighbor (NN) rule or to the Parzen classifier.

Another possibility is to use the distance-based information to construct the support vector classifier (SVC) [3, 12].

The distances can be also treated as a description of a specific feature space, where each dimension corresponds to an object. This does not essentially change the classical feature-based approach, although a special case is considered, where the number of samples n equals their dimensionality k (which is an example of the critical training set size problem) and each data value expresses a dissimilarity between two objects. In general, any arbitrary classifier operating on features can be used. In the learning process, the pattern recognizers are built on the $n \times n$ distance matrix. The p test objects are classified using their distances to the n training samples (the size of the test data is $p \times n$).

Another alternative to tackle dissimilarity data refers to a problem of its embedding into a feature space by imposing that the structure, revealed in distances, is preserved. Having an arbitrary nonnegative and symmetric distance representation, it is in general possible to find a distance preserving mapping onto a pseudo-Euclidean space [6]. In such a space, classifiers can be then built. This approach remains topic for future research.

In this paper, we analyze the behavior of classifiers operating on distance representations (see also [10]). Our aim is not only to investigate, but also to provide some reasoning and general rules, which could suggest a reasonable classifier for a given type of distance measure. We examine the basic and commonly used decision rules which refer to distances either as features or as relations.

2.1 Distributions of distances

Most of the commonly-used distance measures, e.g. (squared) Euclidean, Minkowski or Hamming distances, are based on the sum of (absolute) differences between variables. The key issue is then to realize the importance of the central limit theorem (CLT), which applies to them. Under general circumstances, the CLT states that the mean of n random variables tends to be normally distributed in the limit. The main conditions are that the variances must exist and none of them should dominate. The variables can be drawn from the same or different distributions, nevertheless, their sum tends to approximate the Gaussian. In practical applications, the approximation can be already very good for small n, such as 20, for instance.

The (squared) Euclidean distances computed for a few variables can be considered as being approximately χ^2 distributed. With the growing number of variables (degrees of freedom), their distribution starts to resemble the normal distribution. The square root in the Euclidean distance is a modification, in the spirit of the Box-Cox transformation (see [8]), which also imposes the normality.

The conclusion is that distances based on a sum of many variables are normally distributed, provided that no variance of the sum components dominates. (Otherwise, they are described by χ^2 or gamma distribution, in general, with a few degrees of freedom.) This fact has a crucial effect on the classification task. It suggests that the normal-based classifiers applied to the distance data should perform well, as the assumption on normality is fulfilled.

When only n objects are available in an n-dimensional dissimilarity space, the training samples are not sufficient for representing the real data distribution (the curse of dimensionality [9]). On the other hand, one could expect [13] that such decision rules will not handle the given task well, since they have to deal with the critical learning size problem ($n = k$). In fact, they make use of the inverse of the estimated covariance matrix, which becomes singular. Knowing, however, that data is approximately normally distributed, a sort of regularization should guarantee a way to construct classifiers.

2.2 Classifiers

For feature-based classifiers, simple pattern recognition techniques are expected to generalize better, since less parameters are to be determined on the basis of the given samples. Therefore, linear or quadratic classifiers are of interest.

Although, reduction of the dimensionality is an important issue (see [4, 10]), it remains beyond the scope of this paper. Here, we focus on studying classifiers for the complete dissimilarity data.

Within the group of the feature-based and rank-based classifiers, the following are studied:

Regularized normal-based linear classifier

The normal-based linear classifier (NLC) assumes that all classes are characterized by multi-normal distributions with the same covariance matrix S. For a 2-class problem the NLC is given by:

$$\begin{aligned} f(\boldsymbol{x}) &= \left[\boldsymbol{x} - \frac{\overline{\boldsymbol{x}}_{(1)} + \overline{\boldsymbol{x}}_{(2)}}{2}\right]^T S^{-1} \left(\overline{\boldsymbol{x}}_{(1)} - \overline{\boldsymbol{x}}_{(2)}\right) \\ &+ \;\; 2 \log \frac{P_{(1)}}{P_{(2)}}, \end{aligned}$$

where $P_{(i)}$, $i = 1, 2$ are prior probabilities. Since the rank of the estimated covariance matrix S is not larger than $n-1$ for our dissimilarity data, it is impossible to determine its inverse. Therefore, a regularized version S_r is used instead and such a decision rule is called the regularized normal-based linear classifier (RNLC). Regularization takes care that the inverse operation is possible by emphasizing (e.g. enlarging) the diagonal values (variances) of the matrix S with reference to the off-diagonal elements (covariances).

Regularized normal-based quadratic classifier

The normal-based quadratic classifier (NQC) assumes that the classes have multi-normal distributions, each characterized by different covariance matrix. For a 2-class problem the NQC with the covariances matrices: $S_{(1)}$ and $S_{(2)}$ is given by:

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{i=1}^{2} (-1)^i \, (\boldsymbol{x} - \overline{\boldsymbol{x}}_{(i)})^T \, S_{(i)}^{-1} \, (\boldsymbol{x} - \overline{\boldsymbol{x}}_{(i)}) \\ &+ \;\; 2 \log \frac{P_{(1)}}{P_{(2)}} + \log \frac{|S_{(1)}|}{|S_{(2)}|} \end{aligned}$$

where $P_{(i)}$, $i = 1, 2$ are prior probabilities. When the estimated covariance matrices become singular the regularization versions are used for the dissimilarity data and the classifier is called the regularized normal-based quadratic classifier (RNQC).

Pseudo-Fisher linear discriminant (PFLD)

It uses a single covariance matrix to describe all classes. This classifier originates from the Fisher linear discriminant, obtained by maximizing the ratio of the between-scatter to the within-scatter (Fisher criterion [5]), which for 2 classes is basically the NLC, without regularization. When the estimated covariance matrix is singular, a pseudo-inverse operation is proposed instead and the resulting classifier is called the Pseudo-Fisher linear discriminant [11]. The pseudo-inverse relies on the singular value decomposition of the matrix S and it becomes the inverse of S in the subspace spanned by the eigenvectors corresponding to r non-zero eigenvalues. The classifier is found in this subspace and in the remaining $n - r$ directions is orthogonal to this subspace. The PFLD can perform badly [13], but it is used here as a reference for the RNLC.

Support Vector Classifier (SVC)

The SVC is a hyperplane maximizing the margin between two separable classes (the shortest object distance to the hyperplane) [3, 12]. In case of overlap, the soft margin classifier is introduced, which handles the misclassified points. For the training points: $\boldsymbol{x_1}, \ldots, \boldsymbol{x_n}$ with the labels $\lambda_1, \ldots, \lambda_n$, ($\lambda_i = \pm 1$), the linear SVC is found by:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \underbrace{\alpha_i \, \lambda_i \, (\boldsymbol{x} \cdot \boldsymbol{x_i})}_{w_i \, \boldsymbol{x}} + w_0, \text{ and } \boldsymbol{w} = \sum_{i=1}^{n} \alpha_i \, \lambda_i \, \boldsymbol{x_i},$$

where $(\boldsymbol{x} \cdot \boldsymbol{x_i})$ is the dot product operation and α_i are non-negative values. The $\boldsymbol{w}$ coefficients are expressed as weighted linear combinations of the training objects. In fact, many weights α_i appear to be zero, so in the end only some objects contribute to the classifier. The objects with non-zero weights are called support vectors (SV).

A nonlinear decision function is obtained by a mapping Φ of the input objects to a high-dimensional feature space and finding a linear classifier in that space. This classifier is expressed as [3]:

$$f(\boldsymbol{x}) = \sum_{i=1}^{n} \alpha_i \, \lambda_i \, (\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x_i})) + w_0,$$

where the dot product can be replaced by its generalized version: $K(\boldsymbol{x}, \boldsymbol{x_i}) = (\Phi(\boldsymbol{x}) \cdot \Phi(\boldsymbol{x_i}))$, called a kernel. Since in a high-dimensional space the SVC's function is based on dot products of vectors and support vectors only, this allows for defining explicitly the kernel operator instead of the map Φ. The kernel can be any symmetric and positive definite function fulfilling the Mercer theorem [3].

To introduce the SVC operating on a dissimilarity matrix D, a data dependent mapping D_Φ from the original space to a higher-dimensional space is defined by:

$$D_\Phi : \boldsymbol{x} \rightarrow [D(\boldsymbol{x_1}, \boldsymbol{x}), \ldots, D(\boldsymbol{x_n}, \boldsymbol{x})]^T,$$

which maps each new object $\boldsymbol{x}$ into a vector consisting of the distances to all n training samples. The linear decision function in the distance space becomes:

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \, D_\Phi(\boldsymbol{x}) + w_0$$

The kernel matrix K in the training process consists of the dot products of the form: $(D_\Phi(\boldsymbol{x_i}) \cdot D_\Phi(\boldsymbol{x_j}))$, and it is given by $K = D\,D^T$. It is positive definite by construction. In this approach a linear classifier in the n-dimensional distance space is constructed. We refer to this method as to the SVC-D - support vector classifier on distances.

Another way to define the linear SVC on dissimilarities is as follows. Let us consider the same data-dependent mapping D_Φ. The singular value decomposition of the distance matrix D is found as $D = U\,L\,V^T$ and the whitening-type of transformation W [15] is defined by: $W = L^{-\frac{1}{2}}\,U^T$. This transformation is then applied to the mapping D_Φ. Therefore, the kernel operator becomes:

$$K(\boldsymbol{x_i}, \boldsymbol{x_j}) = (\,(W\,D_\Phi(\boldsymbol{x_i})) \cdot (W\,D_\Phi(\boldsymbol{x_j}))\,),$$

which in the training process becomes the following:

$$K = D\,W^T\,W\,D^T = D\,U\,L^{-1}U^T\,D^T$$

We will refer to this method as to the SVC-D2.

Both the SVC-D and SVC-D2 construct linear classifiers in the distance space, however the SVC-D2 decorrelates the distance features and re-scales them. Therefore, the variances become more similar to each other.

Parzen classifier

The Parzen classifier models the class-conditional probabilities, $P(X|c_i)$ for the class c_i, by kernel density estimation methods. It uses the multi-normal density function, with mean consisting of all training samples and the diagonal covariance matrix with the overall variance h^2:

$$p(\boldsymbol{x}|c_i) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{\sqrt{2\,\pi}\,h}\,e^{-\frac{||\boldsymbol{x}-\boldsymbol{x_i}||^2}{2\,h^2}}$$

The parameter h is determined by maximum-likelihood estimation [5]. Two approaches are then possible: the first one (rank-based) treats the given dissimilarities as the distances to be used in the density function instead of the Euclidean ones, while the second (feature-based) - addresses them in a traditional way, computing the Euclidean distances from the considered dissimilarities.

The nearest neighbor classifier (NN)

The nearest neighbor classifier assigns an object to the class of its nearest neighbor. Two possibilities are here considered. In the first approach (rank-based), the given dissimilarities are used directly, in the second (feature-based) - they are treated as a feature space for which the Euclidean distances are computed and then used for classification.

Decision trees (DT)

A decision tree (DT) is an example of a hierarchical decision process. It partitions the feature space of all possible objects into subregions described in leaves. Different subsets of the original feature space are used at different levels of the tree. Each sample is then classified by the label of the leaf it reaches. We consider here the binary decision trees.

The maximum entropy criterion (DT-max) and the Gini index (DT-Gini) are frequently used splitting rules [2]. In each node, they determine a feature, together with a threshold, to be used for the data partition. When a DT operates on a distance representation, selecting a feature actually means choosing an object. Splitting takes place by checking whether the sample under study lies in a neighborhood (given by the threshold in the considered distance measure) of the selected object or not.

3 Datasets and experiments

A number of real datasets are used in our study:

- **Iris** dataset characterizes 3 species of iris flowers, which are described by 4 attributes: sepal length/width and petal length/width. In total, 150 observations are available; 50 per class.
- **Crabs** dataset describes blue and orange Leptograpsus crabs with their male and female representatives. There are 200 samples given, each characterized by 5 length measurements.
- **Sonar** dataset consists of 111 patterns of a metal cylinder and 97 patterns obtained from rocks. There are 60 features, each representing the energy within a particular frequency band of the sonar signals.
- **Face** dataset consists of faces 256×256 images of 40 people. For each person 10 different images are given.
- **Pump** dataset consist of 900 objects and 500 features. Pump vibration was measured with 5 accelerometers mounted on a submersible pump operating in one abnormal and 3 abnormal states. The wavelet decomposition of the power spectrum was used. For each sensor the 100 coefficients with the largest variances were considered.
- Originally the **Texture** set consists of 7 256×256 filter images, where each is a composition of 5 textures. All points are expressed in a 7-dimensional space consisting of the filters' intensity values for the same pixel. This makes $256^2 = 65536$ objects in total. In our experiments, only 200 randomly chosen objects per texture were used. In Figure 1, two complete filter images are shown.
- **Cband** dataset describes $1D$-chromosome banding patterns. There are 600 objects evenly distributed over 24 classes.
- **Digit** dataset comes from the NIST database [17] and consists of 2000 images evenly distributed over 10 classes.

The characteristics of datasets is summarized in Table 1 and the examples of image data are shown in Figure 1. All datasets, but Sonar, have equally prior probabilities.

In the experiments, the behavior of classifiers built on dissimilarity data is studied. We investigate various distance measures (summation-based,

Table 1: Datasets used in experiments.

	Iris[16]	**Crabs**[14]	**Sonar**[16]	**Face**[1]	**Texture**	**Pump**[18]	**Cband**	**Digit**[17]
Dimens.	4	5	60	256 × 256	7	500	Undefined	256 × 256
# classes	3	4	2	40	5	4	24	10
# p. class	50	5	111,97	10	200	225	25	200
Distance measures	Euclidean	Euclidean	Euclidean Exponent Lp; p=0.5	Hamming	Euclidean	City-block Max-norm	Dot product	Contour

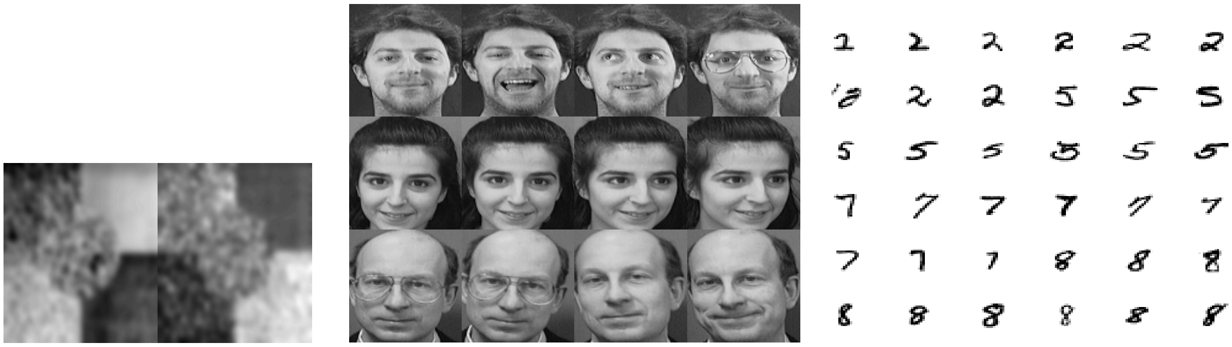

Figure 1: Examples from the Texture, Face and Digit image data sets.

normally or χ^2 distributed, or the max norm distance - the largest absolute difference between objects) and some transformations, as well. We focus on the dissimilarity-based pattern recognition problem itself, i.e. how to deal with such classification problems on distance representations, in general. The Iris, Crabs and Texture datasets come from low-dimensional spaces and their distance representations are considered here as references.

All datasets are randomly split into approximately equally-sized the training and testing sets, taking care that prior probabilities remain equal (except for the Sonar case). In our study, based on 10 runs, all the classifiers are firstly built on the complete $n \times n$ distance matrices and then applied to the test data consisting of $p \times n$ dissimilarities (computed between the p test and n training objects).

4 Discussion

Table 2 presents the mean generalization errors obtained for different distance measures and data. For the RNLC and RNQC only the lowest errors, obtained in a few trials with different regularization parameters, are reported. Although the PFLD does not assume any parametric model, it still makes use of the sample covariance matrix. Thereby, it remains in the spirit of the normal-based classifiers. On the contrary, the SVC is found without any estimation of the class conditional densities, which makes it attractive for our problem.

The most important results and observations are discussed below.

4.1 Distances of originally low-dimensional data

The Iris and Crabs Euclidean distances, based on a small number of variables, are χ^2-distributed, especially since in both cases one variance of sum components slightly dominates. The Iris data consists of three classes, where only two are somewhat overlapping. In such a case, nearly all considered decision rules (except for the DT) perform well.

For the Crabs data with four classes there is a larger overlap and the problem becomes harder. The linear (RNLC or PFLD) or quadratic classifiers perform the best. In order to reduce the influence of the dominant variance, distances for the standardized Crabs data were also considered. The results indicate, that this significantly decreases the error, not only for the RNLC and RNQC, but the other classifiers, as well. For the RNLC and RNQC discrimination functions, although the distances are still χ^2 distributed, they have a larger number of degrees of freedom, which seems to have already a positive effect on the classification. For other classifiers, the improvement in accuracy is probably due to a fact that all features contribute to the overall distance in the same way.

The Texture Euclidean distances are based on 7-component sums and are also χ^2-distributed. There is no dominant variance, so the approximation of the Gaussian distribution starts to be reasonable. The RNLC and PFLD perform well, however the SVC's functions achieve the same or even better results.

Table 2: Mean generalization error with its standard deviation (in %) for different distance data.

Data	**Iris**	**Crabs**	**Crabs**	**Sonar**	**Sonar**	**Sonar**	**Sonar**
Distance	**Euclidean**	**Euclidean**	**Stand.**	**Euclidean**	**Exponent**	**Stand.**	**Lp; p=0.5**
L-1-out NN	4.0	12.0	10.0	17.3	17.3	12.5	15.4
TR sizes	75 × 75	100 × 100	100 × 100	105 × 105	105 × 105	105 × 105	105 × 105
RNLC (f)	3.5 ± 0.4	10.7 ± 0.7	8.7 ± 0.6	17.0 ± 0.8	23.7 ± 1.3	18.4 ± 1.1	21.6 ± 0.9
RNQC (f)	3.1 ± 0.6	18.7 ± 1.3	15.2 ± 1.3	16.9 ± 1.3	21.0 ± 1.1	17.0 ± 1.0	21.4 ± 1.0
PFLD (f)	3.7 ± 0.4	15.1 ± 1.1	11.7 ± 0.8	17.3 ± 0.8	42.2 ± 3.0	18.0 ± 1.1	46.0 ± 1.9
SVC-D (f)	4.1 ± 0.5	11.2 ± 0.8	10.3 ± 0.6	18.5 ± 1.2	24.2 ± 1.2	20.3 ± 1.0	19.6 ± 1.2
# of SV	12	63	60	58	54	56	41
SVC-D2 (f)	3.6 ± 0.5	12.5 ± 0.8	10.0 ± 0.7	17.0 ± 1.2	25.3 ± 1.9	18.1 ± 1.1	23.6 ± 1.2
# of SV	29	92	87	86	70	83	56
1-NN (f)	4.0 ± 0.6	52.5 ± 1.5	41.4 ± 1.5	26.0 ± 1.7	29.1 ± 1.9	22.3 ± 1.6	20.9 ± 1.2
1-NN (r)	4.7 ± 0.5	21.3 ± 1.3	16.3 ± 1.1	18.6 ± 0.9	18.6 ± 0.9	15.2 ± 1.0	18.5 ± 1.0
Parzen (f)	3.1 ± 0.5	51.8 ± 1.6	41.0 ± 1.4	22.8 ± 0.8	25.9 ± 1.1	18.1 ± 0.9	20.5 ± 1.5
Parzen (r)	3.9 ± 0.5	20.8 ± 1.4	16.3 ± 1.1	18.2 ± 1.0	17.7 ± 0.8	14.0 ± 1.1	16.1 ± 0.8
DT-max (r)	6.7 ± 0.6	50.0 ± 2.1	43.1 ± 1.5	29.8 ± 1.2	29.8 ± 1.2	27.1 ± 1.7	27.1 ± 1.6
DT-Gini (r)	10.8 ± 1.5	58.7 ± 1.7	53.2 ± 2.2	28.0 ± 1.4	28.2 ± 1.3	25.1 ± 1.2	27.8 ± 1.3

Data	**Face**	**Texture**	**Pump**	**Pump**	**Cband**	**Cband**	**Digit**
Distance	**Hamming**	**Euclidean**	**City-block**	**Max-norm**	**Product**	**Box-Cox**	**Contour**
L-1-out NN	0.5	5.4	74.4	64.6	27.8	27.8	18.8
TR sizes	200 × 200	500 × 500	452 × 452	452 × 452	600 × 600	600 × 600	1000 × 1000
RNLC (f)	4.1 ± 0.6	7.3 ± 0.3	34.6 ± 0.8	57.8 ± 0.8	25.6 ± 0.6	22.5 ± 0.5	13.6 ± 0.4
RNQC (f)	11.0 ± 0.6	8.5 ± 0.2	37.8 ± 0.8	71.4 ± 0.2	65.7 ± 0.7	26.7 ± 0.5	24.8 ± 0.3
PFLD (f)	2.6 ± 0.4	7.3 ± 0.3	36.9 ± 0.6	73.4 ± 0.8	50.8 ± 0.7	24.2 ± 0.6	14.0 ± 0.4
SVC-D (f)	2.7 ± 0.5	7.2 ± 0.4	36.3 ± 0.3	46.2 ± 0.5	30.2 ± 0.6	23.3 ± 0.5	11.5 ± 0.3
# of SV	200	130	414	406	553	554	729
SVC-D2 (f)	2.5 ± 0.5	6.6 ± 0.2	36.3 ± 0.6	50.0 ± 0.3	32.3 ± 0.6	21.7 ± 0.4	12.5 ± 0.3
# of SV	200	220	447	438	594	599	929
1-NN (f)	3.5 ± 0.5	14.2 ± 0.5	38.2 ± 0.7	46.6 ± 0.7	44.8 ± 0.7	40.1 ± 0.5	15.3 ± 0.3
1-NN (r)	2.9 ± 0.4	9.9 ± 0.2	74.7 ± 0.1	64.0 ± 0.3	31.2 ± 0.3	31.2 ± 0.3	20.7 ± 0.4
Parzen (f)	3.4 ± 0.5	13.4 ± 0.3	22.3 ± 0.6	27.5 ± 0.6	41.6 ± 0.5	37.5 ± 0.3	——
Parzen (r)	3.0 ± 0.6	9.3 ± 0.2	46.4 ± 0.3	40.5 ± 0.8	24.1 ± 0.3	22.7 ± 0.4	20.1 ± 0.3
DT-max (r)	43.6 ± 1.1	15.8 ± 0.5	42.5 ± 0.6	54.1 ± 0.8	50.6 ± 0.9	50.6 ± 0.9	29.1 ± 0.4
DT-Gini (r)	78.7 ± 2.2	22.5 ± 1.1	44.1 ± 0.4	54.7 ± 0.7	79.8 ± 0.6	79.7 ± 0.7	53.4 ± 1.0

——	–	no result, because of numerical instabilities
L-1-out NN	–	the leave-one-out NN rule for the given distance matrix
(f) / (r)	–	feature-based approach (distances as features) / rank-based approach (using given distances)

4.2 Normally distributed distances

The Sonar, Face and Digit sets refer to distances based on a sum of many variables with no dominant variance. The RNLC, RNQC and PFLD [11] should then perform well, since the normality condition holds. The experiments confirm our expectations. They show that such classifiers outperform in general the other ones, provided that the dimensionality is not very high. In case of the Digit data, the distance representation describes a 1000-dimensional space. In such a space, the SVC, although based on many support vectors (73% or 93% of all objects), seems to discriminate more accurately then the normal-based decision rules. The normal-based classifiers make use of the inverse of a large sample covariance matrix, and it is possible that in the computational process of its determination some numerical instabilities can appear more easily.

The variances of the distance sum components of the Sonar data are similar, so the standardization will not help in case of the normal-based classifiers. The tests demonstrate this clearly. However, the standardization improves the performance of all rank-based decision rules. One possible explanation can be that when the original features are of the same order of magnitude with equal variances, then each of them contributes to the summation-based distances in the same way. This gives more a homogeneous description and allows dissimilarities to make use of the discriminative power of all features.

4.3 Non-normally distributed distances

For the Sonar Euclidean distances (normally distributed), the exponential equivalents for all nonzero distances were investigated. The aim of this operation is to study the changes in accuracy of classifiers. In case of the rank-based pattern recognition techniques, such a transformation has a minor influence or no influence at all (because the rank is kept). The performance of the normal-based classifiers, as expected, became much worse due to imposed nonnormalities. Also both SVC's functions give significantly worse results.

The Pump dataset with max-norm distances and the Cband with inner-product distances were also studied. The max-norm distances are definitely not a good measure for the Pump classification. They are studied here only in order to get an illustrative example of non-normally distributed data. The Cband dissimilarities, although based on sum, are approximately χ^2 distributed. In those two cases, the rank-based methods, like NN, Parzen and DT (Gini index) perform a bit better then the normal-based classifiers. However, both SVC's discrimination functions seem to approach the best accuracy obtained by the rank-based decision rules.

4.4 Imposing the normality

One possible way to deal with the non-normally distributed distances is to impose the Gaussian distribution on them by using the Box-Cox transformation [8]. Such a transformation is applied for each element d of both the training and testing distance matrices by using the following formula: $\frac{d^p-1}{p}$, for $p \in (0,1]$. The classification problem solved for the transformed ($p = 0.25$) Cband dataset confirms that the feature-based pattern recognizers have considerably better performance than without such a transformation (inner-product distances).

4.5 Non-metric distances

The $L_{0.5}$ distance (for which a triangle inequality does not hold) was considered as an example of a non-metric distance measure. For two vectors: $\boldsymbol{x}$ and $\boldsymbol{y}$ it is given by:

$$d_{L_{0.5}}(\boldsymbol{x},\boldsymbol{y}) = \left(\sum_{i=1}^{n} \sqrt{|x_i - y_i|}\right)^2$$

Such summation-based dissimilarities were applied to Sonar data. The sums themselves are approximately normally distributed, however the square operation disturbs it. Thereby, the approximation is not so good any longer. This is confirmed also by the RNLC and RNQC results which are significantly worse than in case of the Euclidean distances. The rank-based NN classifier and DT seem to perform in a similar way, as for the Euclidean distances, but the rank-based Parzen classifier achieves much worse accuracy. However, all those results indicate that pattern recognizers can be applied to non-metric representations, as well.

4.6 SVC

For the Iris, Crabs and Sonar data, both SVC's functions perform mostly worse than the parametric classifiers. In case of other datasets, the SVC-D and SVC-D2 often outperform the normal-based decision rules. One possible explanation of this fact is that the dimensionality is of not much importance for the SVC, while it has a major effect on the normal-based classifiers. It seems that it is easier to successfully regularized such classifiers in a lower-dimensional space (e.g. 75D Iris or 100D Crabs distance data) then in a high-dimensional space (e.g. 500D distance Texture data).

4.7 Decision trees, NN and Parzen classifiers

What is very surprising, is the poor performance of DT classifiers, even in case of a little overlap between the Iris classes. When the best features, equal to objects in the distance space, are sought for a split, such pattern recognizers should give good results. Choosing a feature for a split, in the process of building a DT, stands for finding a good prototype and considering objects lying in its sphere-neighborhood (in the given metric). The DT gives an error-free result on the training data (no pruning used), but it drastically increases the test error. It seems that the objects chosen for the splits are definitely not good reference points. There is often a number of them equally good for a split (according to a criterion) and in such cases one is randomly chosen. It turns out that the selected objects are mostly representatives of the first few classes. When many classes are present (e.g. Cband or Face data), not every class is represented by an object. One must then realize that this way of splitting does not positively influence the classification results. Possibly, some variants of DT could be considered, which take into account objects which are more evenly distributed over classes.

On the contrary, the NN rule and the Parzen classifier applied to the distances in a rank-based way give reasonable results, however often not better than those obtained by the RNLC or the SVC.

5 Conclusions

There are important conclusions which can be drawn both from the CLT and our experiments.

First of all, summation-based distances of many variables are approximately normally distributed, provided that none of the variance component dominates. In such cases, the normal-based classifiers significantly outperform the rank-based ones.

Secondly, when there is a dominant variance or the distances describe a low-dimensional representation, they are χ^2 distributed. In this situation, two approaches are possible. The Box-Cox transformation imposes normality on the distance distribution, which has a positive effect on the feature-based classifiers, while the rank-based classifiers give the same results. Standardization in the original space (before distances are computed) significantly improves the performance of both types of classifiers. However, it is not often possible, when only distances are given.

Thirdly, for the dissimilarities which are not based on sums, or which distributions are far from the Gaussian, the rank-based classifiers, i.e. the NN rule, Parzen classifier and DT, give results which are not worse than those obtained by the normal-based decision rules.

Next, the linear support vectors classifiers perform in general well. They often reach one of the best accuracy, especially when the dissimilarity space is high-dimensional, e.g. 400D or more. Those classifiers do not estimate the class conditional density functions, but they try to maximize the (soft) margin between classes, instead. This way of proceeding does not suffer from the curse of dimensionality [9]. In most cases, the SVC-D, constructing a linear classifier in the distance space, performs better than the SVC-D2, constructing also a linear classifier but in the transformed (decorrelated) space. Nearly all dissimilarity representations need at least 50% (up to 100%) of all objects to become the support vectors. This suggest also how difficult in such a critical learning set size problem is to establish the boundary between classes.

Finally, decision trees perform here badly, much worse than expected. One possible reason is that no pruning was used. Also, the objects (features) chosen for splitting during the process of building a decision tree can be often representatives of only first few classes, so in case of many classes much worse performance is observed (e.g. Cband or Face sets versus Iris or Sonar). This remains a topic for further research to find a suitable solution.

6 Acknowledgments

This work was partly supported by the Foundation for Computer Science Research in The Netherlands (SION) and the Dutch Organization for Scientific Research (NWO).

References

[1] AT&T Labs,
http://www.cam-orl.co.uk/facedatabase.html.

[2] L Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and regression trees.* Wadsworth & Brooks, 1984.

[3] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

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

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition.* Academic Press, 1990.

[6] L. Goldfarb. A unified approach to pattern recognition. *Pattern Recognition*, 17(5):575–582, 1984.

[7] L. Goldfarb. What is distance and why do we need the metric model for pattern learning? *Pattern Recognition*, 25(4):431–438, 1992.

[8] R. van der Heiden and F. C. A. Groen. The Box-Cox metric for nearest neighbour classification improvement. *Pattern Recognition*, 30(2), 1997.

[9] A. K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics*, volume 2, pages 835–855. North-Holland, Amsterdam, 1987.

[10] E. Pękalska and R. P. W. Duin. Classifiers for dissimilarity-based pattern recognition. In *ICPR*, Barcelona (Spain), 2000, accepted.

[11] S. Raudys and R. P. W. Duin. On expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix. *Pattern Recognition Letters*, 19(5-6):385–392, 1998.

[12] B. Schölkopf. Support vector learning. Published by: R. Oldenbourg Verlag, Munich, 1997, 1997. PhD thesis,.

[13] M. Skurichina and R. P. W. Duin. Bagging for linear classifiers. *Pattern Recognition*, 31(7):909–930, 1998.

[14] StatLib, http://lib.stat.cmu.edu/.

[15] K. Tsuda. Support vector classifiers with asymetric kernel functions. Technical Report TR-98-31, Machine Understanding Division, Electrotechnical Laboratory, Japan, 1998.

[16] UCI Machine Learning Repository,
http://www.ics.uci.edu/ mlearn.

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

[18] A. Ypma, D. M. J. Tax, and R. P. W. Duin. Robust machine fault detection with independent component analysis and support vector data description. In *IEEE International Workshop on Neural Networks for Signal Processing*, pages 67–76, Wisconsin (USA), August 1999.