Classifiers for dissimilarity-based pattern recognition

Elżbieta Pękalska 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
E-mail: ela@ph.tn.tudelft.nl

Abstract

In the traditional way of learning from examples of objects the classifiers are built in a feature space. However, alternative ways can be found by constructing decision rules on dissimilarity (distance) representations, instead. In such a recognition process a new object is described by its distances to (a subset of) the training samples.

In this paper a number of methods to tackle this type of classification problem are investigated: the feature-based (i.e. interpreting the distance representation as a feature space) and rank-based (i.e. considering the given relations) decision rules. The experiments demonstrate that the feature-based (especially normal-based) classifiers often outperform the rank-based ones. This is to be expected, since summation-based distances are, under general conditions, approximately normally distributed. In addition, the support vector classifier achieves also a high accuracy.

1. Introduction

Real-world data consist of measurements, which form a set of attributes describing objects. The objects are then represented in a feature space. For classification purposes, features should emphasize the differences between classes. In some real applications the set of features needed for a good discrimination can increase drastically and the problem is often hard due to the curse of dimensionality [7].

Dissimilarity-based pattern recognition (DPR) offers new possibilities for building classifiers on distance representations. It becomes especially useful when the original data is described by many features or when experts cannot formulate the attributes explicitly, but they are able to provide a dissimilarity measure, instead.

This paper investigates methods for building classifiers on distance representations. A few such approaches are introduced in section 2, in which some insight into the types of distance distributions and their consequences for classifiers is presented, as well. Section 3 describes the conducted experiments. The results are discussed in section 4 and the conclusions are summarized in section 5.

2. Dissimilarity-based pattern recognition

A dissimilarity value expresses a magnitude of difference between two objects and becomes zero only when they are identical. Such measures differ according to various datasets or applications and the study of their best characteristics is an issue of further research.

A straightforward way of dealing with dissimilarity representations is based on the distance relations between objects, which naturally leads to the rank-based methods, e.g. to the nearest neighbor rule. It is also possible to construct a support vector classifier based on distance information [3].

The distance data 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 it embeds the problem in a special case, when the number of samples n equals their dimensionality k (which is an example of the critical training size problem). In general, any arbitrary classifier operating on features can be used. In the learning process, the decision rules are constructed on the complete $n \times n$ pairwise distance matrix. The p test objects are then classified by using their distances to the n training samples (the test data consists of $p \times n$ dissimilarities).

When only n samples are available in an n-dimensional space, they are not sufficient for representing the real data distribution. It is known [9] that the feature-based classifiers can perform poorly. Therefore, reduction of the dimensionality is important, also because of the computational aspect when the test samples are considered. Besides the complete $n \times n$ distance matrices, also their $n \times m$ ($m < n$) reduced versions are studied, which are sets of dissimilarities computed between n training samples and m prototypes chosen from their collection. Our earlier experiments [4] show that objects selected in a pseudo-random way may not spoil the classification results much in comparison with more sophisticated methods. Therefore, as our objective is to investigate the behavior of classifiers and not to considerably improve their performance, we limit ourselves to a random selection.

2.1. Distributions of distances

Most of the widely-used distance measures, e.g. Euclidean or Hamming distances, are based on sums. The key issue is then to realize the central limit theorem (CLT), which applies to them. Under general circumstances, the mean of n random variables tends to be normally distributed in the limit. The main condition is that there is no dominant variance. The variables can be drawn from the same or different distributions, nevertheless, their sum approximates the Gaussian. In practice, the approximation can be already very good for small n, such as 20, for instance.

0-7695-0750-6/00 $10.00 © 2000 IEEE

The (squared) Euclidean distances computed for a few variables are approximately χ^2 distributed. With the growing number of variables (degrees of freedom), the distribution starts to resemble the normal one. The square root in the Euclidean distance is a modification in the spirit of the Box-Cox transformation, which also imposes normality [6]. The distances based on a sum of many variables are approximately normally distributed, provided that none of the variances dominates. This has a crucial effect on our classification task. It suggests that the normal-based classifiers built on dissimilarities should perform well, since the assumption on normality is fulfilled. However, one could expect [9] that such decision rules will not handle the given task well, since they deal with the critical learning size problem ($n = k$). In fact, they make use of the inverse of the sample covariance matrix, which becomes singular. Knowing, however, that data is approximately normally distributed, regularization should guarantee a way to find good classifiers.

2.2. Classifiers

Our aim is not only to investigate the classifiers on distance representations, but also to provide with some general rules suggesting a reasonable classifier for a given type of distance measure. Within the group of the feature-based and rank-based decision rules, the following are studied:

Normal-based linear/quadratic classifiers (NLC/NQC)

The NLC/NQC assumes that the classes are described by multi-normal distributions with the same/different covariance matrices. Since for $n \times n$ dissimilarity representations the estimated covariance matrix S is singular, its inverse cannot be determined. Therefore, its regularized version is used instead and the resulting classifier is the regularized normal-based 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). Also, when the regularization versions are used in case of the NQC, the resulting classifier becomes the regularized normal-based quadratic classifier (RNQC).

Pseudo-Fisher linear discriminant (PFLD)

The PFLD uses a single covariance matrix to describe all classes. It 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. When the estimated covariance matrix is singular, a pseudo-inverse operation is proposed instead and such a classifier is called the PFLD [8]. 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. It can perform badly, but it is used here as a reference point for the (R)NLC.

Support Vector Classifier (SVC)

The SVC is a hyperplane maximizing the margin between 2 separable classes (the shortest object distance to the hyperplane) [3]. In case of overlap, the soft margin hyperplane is introduced, which handles the misclassified objects. 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 given by: $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + w_0 = \sum_{i=1}^{n} \alpha_i \lambda_i (\boldsymbol{x} \cdot \boldsymbol{x_i}) + w_0$, 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. In fact, many weights α_i appear to be zero, therefore in the end only the objects with non-zero weights, i.e. support vectors (SV), contribute to the classifier. A nonlinear decision function is obtained by a mapping Φ of the input objects to a high-dimensional space and finding a linear classifier there.

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 as: $D_\Phi : \boldsymbol{x} \rightarrow [D(\boldsymbol{x_1}, \boldsymbol{x}), \ldots, D(\boldsymbol{x_n}, \boldsymbol{x})]^T$, which maps a new object $\boldsymbol{x}$ into a vector consisting of the distances to the n training samples. The linear decision function, constructed in the distance space, is found then by: $f(\boldsymbol{x}) = \boldsymbol{w}^T D_\Phi(\boldsymbol{x}) + w_0$.

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.

Parzen classifier

The Parzen classifier models the class-conditional probabilities 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 (found e.g. by maximum likelihood [5]). Two approaches are here possible. The first one (rank-based) uses the given dissimilarities in the density function instead of the Euclidean distances, while the second (feature-based) - addresses them in a traditional way, computing the Euclidean distances from the given data.

Decision trees (DT)

We consider here the binary DT. The maximum entropy criterion (DT-max) and the Gini index (DT-Gini) are the most frequently used splitting rules [2]. In each node, they determine a feature, together with a threshold, to be used for the partition. When operating on a distance representation, selecting a feature means actually 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

In our experiments the behavior of classifiers operating on dissimilarity data is investigated. We study various distance measures (summation-based, normally or χ^2 distributed, or the max-norm) and some transformations applied to data. Thus, we focus on the DPR problem itself, i.e. how to deal with dissimilarity representations, in general.

Table 1 presents the characteristics of the datasets used. All datasets are randomly split into approximately equally-sized the training and testing sets, taking care that the prior probabilities are kept. In our study, based on 25 runs, two types of experiments are conducted. In the first one, the

Table 1. Datasets used in experiments.

	Iris [11]	**Crabs** [10]	**Sonar** [11]	**Face** [1]	**Pump** [13]	**Pump2** [13]	**Cband**	**Digit** [12]
Dimensionality	4	5	60	256× 256	640	500	Undefined	256 × 256
# classes/# per class	3/50	4/5	2/111,97	40/10	4/225	4/225	24/25	10/200
Considered	Euclidean	Euclidean	Euclidean	Hamming	Euclidean	City-block	Inner	Contour
distances			Exponent		Max-norm	Max-norm	Product	

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). In the second approach, the reduced $n \times m$ ($m \approx n/5$) distance sets are considered. Such sets express the dissimilarities between n training samples and m prototypes, randomly chosen from their collection, which are distributed over all classes according to the prior probabilities. The size of the test data is then $p \times m$.

The NLC in the Karhunen-Loéve (KL) projected space was also considered as a reference. However, this is not studied here, since it was a part of our previous research [4].

4. Discussion

Table 2 presents the mean generalization errors (with their standard deviations) for various distance measures and data. In case of 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 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 3 classes, where only 2 are somewhat overlapping. In such a case, nearly all considered decision rules (except for the DT) perform well.

For the 4-class Crabs data, with a larger overlap, the linear classifiers perform the best. In order to reduce the influence of the dominant variance, distances for the standardized data were also considered. The results indicate that this significantly improves the performance of nearly all pattern recognizers. 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 normal-based classifiers. Also, the ranked-based decision rules become more accurate, since after standardization the discriminative power of all features is used in a similar way.

4.2. Normally distributed distances

The Sonar, Face, Pump, Pump2 and Digit distances are based on sums with no dominant variance. The RNLC, RNQC and PFLD [8] classifiers should perform well on such datasets, since the normality condition holds. The experiments confirm our expectations, both for the complete distance matrices and the reduced once. They show that such decision rules outperform in general the other ones.

The variances of the distance sum components for the Sonar data are similar, so standardization will not help in case of the normal-based classifiers. The tests demonstrate this clearly. However, both for the Crabs and Sonar data, it improves the performance of all rank-based classifiers. When the original features are of the same order of magnitude with equal variances, then each contributes to the summation-based distances in the same way. This gives more homogeneous description and allows distances to make use of the discriminative power of all features.

4.3. Non-normally distributed distances

The exponential equivalents of all non-zero Sonar Euclidean distances (normally distributed) are investigated. Our goal is to study the changes in the classification accuracy. In case of the rank-based classifiers, such a transformation has a minor influence (or not at all) since the rank is kept. The performance of the normal-based decision rules, as expected, is much worse due to imposed non-normalities.

The Pump and Pump2 datasets with max-norm distances and Cband with inner-product distances were also studied. The max-norm distances are definitely non-normally distributed, and Cband, although based on sums, are approximately χ^2 distributed. In such cases, the rank-based methods, like the NN, Parzen and DT-max mostly perform a bit better then the linear or quadratic classifiers.

4.4. Imposing the normality

One possibility to deal with the non-normally distributed distances is to impose the Gaussian distribution on them by using the Box-Cox transformation [6]. Such a transformation is applied to each element d of both the training and testing dissimilarity matrices by using the following formula: $(d^p - 1)/p$, and $p \in (0, 1]$. The classification problem solved for the transformed ($p = 0.25$) Cband dataset confirms that the normal-based classifiers have considerably better performance in case of both the complete and reduced distance matrices then without such a transformation.

4.5. Decision trees, NN and Parzen classifiers

What is surprising, is the poor performance of the DT classifiers, even in case of separable Pump classes. It should have been of use, especially when non-normal distances are considered. 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

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

Data	Iris	Crabs	Crabs	Sonar	Sonar	Sonar	Face
Distance	**Euclidean**	**Euclidean**	**Stand.**	**Euclidean**	**Exponent**	**Stand.**	**Hamming**
L-1-out NN	4.0	12.0	10.0	17.3	17.3	12.5	0.5
TR sizes	75 × 75	100 × 100	100 × 100	105 × 105	105 × 105	105 × 105	200 × 200
RNLC (f)	3.6 ± 0.3	10.7 ± 0.8	8.6 ± 0.6	15.7 ± 0.6	21.7 ± 0.6	17.5 ± 0.9	3.0 ± 0.3
RNQC (f)	3.3 ± 0.3	18.1 ± 1.0	14.2 ± 0.9	16.0 ± 0.6	19.3 ± 0.7	16.5 ± 0.8	10.7 ± 0.6
PFLD (f)	4.0 ± 0.3	14.5 ± 0.7	11.6 ± 0.7	15.8 ± 0.6	38.4 ± 1.5	16.6 ± 0.9	1.9 ± 0.2
SVC (f)	5.0 ± 0.5	11.3 ± 0.8	11.4 ± 0.7	18.2 ± 0.8	21.3 ± 0.7	17.5 ± 0.8	2.0 ± 0.2
# SV	12	64	73	61	57	60	200
1-NN (f)	4.3 ± 0.5	51.1 ± 1.0	40.4 ± 0.8	21.9 ± 0.7	25.6 ± 0.9	20.3 ± 1.2	3.0 ± 0.3
1-NN (r)	4.5 ± 0.4	20.2 ± 0.9	17.3 ± 0.8	18.9 ± 0.7	18.9 ± 0.7	14.9 ± 0.8	2.8 ± 0.3
Parzen (f)	3.4 ± 0.3	50.9 ± 1.0	40.4 ± 0.8	20.7 ± 0.7	24.7 ± 0.7	20.7 ± 0.7	3.0 ± 0.2
Parzen (r)	3.7 ± 0.3	20.0 ± 0.9	17.2 ± 0.8	18.6 ± 0.7	17.6 ± 0.7	18.6 ± 0.7	2.8 ± 0.3
DT-max (r)	6.6 ± 0.6	48.1 ± 1.1	41.1 ± 0.9	29.8 ± 0.8	29.8 ± 0.9	25.9 ± 1.0	40.5 ± 0.7
DT-Gini (r)	8.1 ± 0.7	58.6 ± 1.1	51.4 ± 0.8	30.1 ± 0.7	30.0 ± 0.8	24.6 ± 1.0	79.0 ± 0.8
TR sizes	75 × 15	100 × 20	100 × 20	105 × 21	105 × 21	105 × 21	200 × 40
(R)NLC (f)	3.5 ± 0.4	15.3 ± 1.0	13.3 ± 0.9	25.4 ± 1.0	24.2 ± 0.9	22.5 ± 0.7	14.3 ± 0.6
(R)NQC (f)	3.8 ± 0.4	27.6 ± 1.2	25.2 ± 1.0	24.0 ± 0.8	24.5 ± 0.9	22.1 ± 0.9	15.0 ± 0.6
FLD (f)	3.9 ± 0.4	21.2 ± 0.9	17.5 ± 0.8	25.4 ± 1.0	24.2 ± 0.9	22.5 ± 0.7	17.1 ± 0.5
SVC (f)	5.4 ± 0.5	13.1 ± 1.1	20.6 ± 1.2	26.7 ± 1.0	22.8 ± 0.8	23.8 ± 0.6	6.2 ± 0.3
# SV	12	74	91	77	55	47	185
1-NN (f)	4.6 ± 0.6	52.3 ± 1.0	42.0 ± 0.9	25.0 ± 0.8	29.0 ± 1.1	25.2 ± 1.0	7.3 ± 0.4
1-NN (r)	6.7 ± 0.8	51.9 ± 1.5	43.0 ± 1.3	34.1 ± 1.2	34.1 ± 1.2	30.0 ± 1.1	26.2 ± 0.5
Parzen (f)	3.7 ± 0.3	52.2 ± 0.9	42.2 ± 0.9	23.0 ± 0.7	25.5 ± 0.7	22.5 ± 0.7	7.3 ± 0.4
Parzen (r)	5.3 ± 0.8	51.5 ± 1.3	42.8 ± 1.3	34.1 ± 1.2	33.6 ± 1.2	29.1 ± 1.2	26.2 ± 0.5
DT-max (r)	6.0 ± 0.5	53.7 ± 0.9	44.8 ± 1.1	32.2 ± 0.7	32.2 ± 0.7	29.8 ± 1.2	41.0 ± 1.0
DT-Gini (r)	9.2 ± 0.9	59.2 ± 1.0	54.8 ± 1.0	32.4 ± 0.9	32.5 ± 0.8	28.2 ± 1.1	77.5 ± 1.3
KL - NLC	2.9 ± 0.3	12.8 ± 0.8	9.6 ± 0.6	22.2 ± 0.7	24.7 ± 0.8	22.4 ± 0.7	5.8 ± 0.3

Data	Pump	Pump	Pump2	Pump2	Cband	Cband	Digit
Distance	**Euclidean**	**Max-norm**	**City-block**	**Max-norm**	**In. Product**	**Box-Cox**	**Contour**
L-1-out NN	0.0	0.0	74.4	64.6	27.8	27.8	18.8
TR sizes	452 × 452	452 × 452	452 × 452	452 × 452	600 × 600	600 × 600	1000× 1000
RNLC (f)	0.0 ± 0.0	0.4 ± 0.1	34.4 ± 0.4	52.3 ± 0.3	26.1 ± 0.3	22.3 ± 0.3	13.9 ± 0.2
RNQC (f)	0.0 ± 0.0	0.7 ± 0.1	38.2 ± 0.4	74.9 ± 0.0	65.9 ± 0.5	25.9 ± 0.3	25.5 ± 0.5
PFLD (f)	0.0 ± 0.0	3.2 ± 0.3	36.9 ± 0.4	72.3 ± 0.5	50.5 ± 0.3	24.1 ± 0.3	15.0 ± 0.5
SVC (f)	0.0 ± 0.0	0.4 ± 0.1	36.8 ± 0.3	47.2 ± 0.2	28.3 ± 0.3	24.5 ± 0.3	12.2 ± 0.2
# SV	75	213	413	405	524	554	747
1-NN (f)	0.5 ± 0.1	2.8 ± 0.1	38.2 ± 0.4	47.6 ± 0.3	44.2 ± 0.3	39.4 ± 0.4	15.8 ± 0.2
1-NN (r)	0.0 ± 0.0	0.1 ± 0.0	74.5 ± 0.1	64.9 ± 0.3	30.8 ± 0.3	30.8 ± 0.3	21.5 ± 0.3
Parzen (f)	0.5 ± 0.1	2.7 ± 0.1	37.3 ± 0.4	49.9 ± 0.3	46.1 ± 0.3	37.0 ± 0.3	———
Parzen (r)	0.0 ± 0.0	0.1 ± 0.0	72.2 ± 0.1	71.2 ± 0.2	26.1 ± 0.3	34.2 ± 0.4	20.4 ± 0.2
DT-max (r)	6.8 ± 0.3	9.6 ± 0.4	43.6 ± 0.5	54.0 ± 0.6	51.4 ± 0.4	51.3 ± 0.4	29.2 ± 0.4
DT-Gini (r)	13.1 ± 0.4	20.3 ± 0.6	43.9 ± 0.4	55.1 ± 0.4	80.3 ± 0.5	80.3 ± 0.5	53.9 ± 0.8
TR sizes	452 × 92	452 × 92	452 × 92	452 × 92	600 × 120	600 × 120	1000× 200
(R)NLC (f)	0.0 ± 0.0	1.7 ± 0.2	38.1 ± 0.4	51.7 ± 0.4	25.7 ± 0.3	25.1 ± 0.4	16.1 ± 0.2
(R)NQC (f)	0.2 ± 0.1	3.8 ± 0.3	59.6 ± 0.5	66.3 ± 0.3	51.6 ± 0.5	27.1 ± 0.3	35.1 ± 0.8
FLD (f)	0.0 ± 0.0	2.0 ± 0.2	40.6 ± 0.4	51.9 ± 0.4	50.5 ± 0.3	25.8 ± 0.4	16.8 ± 0.2
SVC (f)	0.0 ± 0.0	5.1 ± 0.3	41.9 ± 0.4	49.6 ± 0.3	56.4 ± 0.9	29.9 ± 0.3	17.6 ± 0.2
# SV	67	318	377	412	560	588	578
1-NN (f)	0.6 ± 0.1	3.7 ± 0.2	38.8 ± 0.3	48.7 ± 0.4	44.4 ± 0.4	42.1 ± 0.4	16.5 ± 0.1
1-NN (r)	0.4 ± 0.2	2.5 ± 0.3	74.8 ± 0.1	66.3 ± 0.4	41.9 ± 0.4	41.9 ± 0.4	29.2 ± 0.4
Parzen (f)	0.5 ± 0.1	3.4 ± 0.2	37.7 ± 0.3	49.8 ± 0.4	42.5 ± 0.3	40.1 ± 0.3	15.6 ± 0.2
Parzen (r)	0.4 ± 0.2	2.5 ± 0.4	74.1 ± 0.1	66.2 ± 0.4	39.2 ± 0.4	37.9 ± 0.4	28.4 ± 0.3
DT-max (r)	5.7 ± 0.3	11.0 ± 0.4	44.2 ± 0.5	54.6 ± 0.5	53.2 ± 0.4	53.2 ± 0.4	30.3 ± 0.2
DT-Gini (r)	12.4 ± 0.5	19.3 ± 0.6	44.0 ± 0.4	55.3 ± 0.5	78.9 ± 0.5	78.9 ± 0.5	52.3 ± 0.5
KL - NLC	0.0 ± 0.0	0.4 ± 0.1	37.4 ± 0.3	49.1 ± 0.3	25.6 ± 0.4	23.2 ± 0.2	12.7 ± 0.2

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

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, however, that the objects chosen for the splits are not good reference points. There is often a number of them, equally good for a split (according to a criterion), and one is randomly chosen. It turns out that the selected objects are representatives of mostly the first few classes. When many classes are present (e.g. Cband or Face data), not every class is represented by an object in the hierarchical decision process. Thereby, this does not positively in-

fluence the classification results. Possibly, some variants of DTs could be considered, which take into account objects which are more evenly distributed over classes.

On the contrary, the NN and Parzen classifiers operating on distance representations in a rank-based way give reasonable results when the complete sets are considered, however they are often not better than the RNLC or the SVC.

4.6. SVC

The SVC is often slightly less accurate than the parametric classifiers for the lower-dimensional distance data. However, it can outperform the other decision rules when many features are given. The dimensionality is not very important for the SVC construction, while it has a major effect on the normal-based classifiers. It also seems that it is easier to regularize them in a lower-dimensional space (e.g. 75D Iris or 100D Crabs distance data) then in a very high-dimensional space (e.g. 1000D Digit contour data).

4.7. The reduced sets versus complete distance sets

When the reduced sets are considered, most decision rules decrease their generalization capability compared to the complete distance representations. However, in case of separable or slightly overlapping classes (e.g. Iris or Pump data), the performance, especially of the feature-based classifiers, is nearly the same. Also, in case of many, only partly overlapping, classes (e.g. Digit data) or classes with a larger overlap (e.g. Pump2, Cband sets), in most cases the decision rules have slightly worse results. The exception is the NN for which the accuracy decreases much. This is understandable since the prototypes are chosen randomly and they may be not the most suitable for the NN. The performance of the DT classifiers does not change much, also because it is already quite poor. In nearly all cases the NLC in the KL space (being a linear combination of all features) gives better results than the decision rules based on a random object selection. However, all these experiments indicate that it is worth looking for a selection method that is both good and cheap, since the same (or even better) results can be achieved. This is a topic for further research.

5. Conclusions

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

Summation-based distances of many variables are approximately normally distributed, provided that no variance dominates. In such cases, the normal-based classifiers significantly outperform the rank-based ones.

When there is a dominant variance or the distances describe a low-dimensional representation, they are χ^2 distributed. The Box-Cox transformation imposes normality of the distribution, which has a positive effect on the feature-based classifiers (while the rank-based classifiers give the same results). Another possibility is standardization in the original space (before distances are computed) improves significantly the performance of all decision rules. However, this is not often the case when only distances are given.

For the distances 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 feature-based classifiers.

The SVC performs in general well. It often reaches one of the best accuracy, especially when the distance space is high-dimensional, e.g. 200D or more. This classifier maximizes the (soft) margin between classes and thereby it does not suffer from the curse of dimensionality [7]. Nearly all dissimilarity representations need at least 50% of the training objects to be the support vectors. This also suggest how difficult, especially in a critical learning size problem, is to establish the class boundaries.

The decision trees perform worse than expected. One possible reason is that no pruning was used. Also, the objects, chosen for a split during the process of building a DT, are often representatives of only first few classes, thereby in case of many classes much worse performance is observed (e.g. Cband or Face versus Iris or Sonar sets). Finding a suitable solution is a topic for further research.

Finally, the classifiers built on the reduced sets perform often (slightly) worse than on the complete distance representations. However, they give similar results in case of either a small or a large overlap, or separable classes. Better results can be achieved by finding an appropriate selection method. This is an open question for further investigation.

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. Friedman, R. Olshen, and C. 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 Rec. Letters*, 20(11-13):1175–1181, 1999.
[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.
[6] R. v. d. Heiden. The Box-Cox metric for nearest neighbour classification improvement. *Pattern Recogn.*, 30(2), 1997.
[7] 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, 1987.
[8] S. Raudys and R. P. W. Duin. On expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix. *Pattern Rec. Letters*, 19(5-6):385–392, 1998.
[9] M. Skurichina and R. P. W. Duin. Bagging for linear classifiers. *Pattern Recognition*, 31(7):909–930, 1998.
[10] StatLib, http://lib.stat.cmu.edu/.
[11] UCI Machine Learning, http://www.ics.uci.edu/ mlearn.
[12] C. Wilson and M. Garris. Handprinted character database 3. Technical report, National Insitute of Standards and Technology, February 1992.
[13] 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 Int. Workshop on NN for Signal Processing*, pages 67–76, USA, 1999.