KYBERNETIKA — VOLUME 32 (1996), NUMBER 4, PAGES 001 – 007

FEATURELESS CLASSIFICATION

Robert P.W. Duin [1], Dick de Ridder and David M.J. Tax

In this paper the possibilities are discussed for training statistical pattern recognizers based on a distance representation of the objects instead of a feature representation. Distances or similarities are used between the unknown objects to be classified with a selected subset of the training objects (the support objects). These distances are combined into linear or nonlinear classifiers. In this approach the feature selection problem is replaced by finding good similarity measures. The proposal corresponds with determining classification functions in Hilbert space using an infinite feature set. It is a direct consequence of Vapnik's support vector classifier [2].

1. INTRODUCTION

Research in statistical pattern recognition has traditionally been dominated by feature vector approaches: objects are represented by feature sets of equal size. These are represented in vector spaces followed by the development of classifiers separating as good as possible the feature vector sets of different classes.

An important drawback of this approach is that on apriori grounds (i.e. on the physical nature of the objects) features have to be defined that are strongly related with class differences. This set may not be too large, both, for computational reasons as well as to preserve the generalization power of the resulting classifiers. Feature spaces of increasing dimensionality finally deteriorate the recognition performance. This 'curse of dimensionality', also known as Rao's paradox [7] or as the peaking phenomenon [8] makes it necessary to have enormous numbers of training examples available for large feature sizes. Simple rules of thumb based on Cover's work [6] demand something like ten times the feature size. Worst case approaches based on the VC dimension [1] demand for almost all classifiers exponentially increasing training sets. Consequently much research is done in finding small sets of good features on apriori grounds or in statistical techniques to reduce initially too large feature sets.

In this study we will reinvestigate the possibility of avoiding the necessity of finding features. We will return to one of the most naive approaches: distances or similarities between direct sensor representations of the objects. So we don't look

[1]This work was supported by the Foundation for Applied Sciences (STW), the Foundation for Computer Science in the Netherlands (SION) and the Dutch Organization for Scientific Research (NWO).

for good features but directly use a similarity measure $S_x(x_i, x_j)$ between objects x_i and x_j. This measure should be such that it emphasizes class differences. Just like the feature definitions it has to be based on application knowledge. The object representations and the way these similarities are measured are not important for the remainder of this paper. They are application dependent. We will focus on the possibilities of building classifiers based on these similarity measures. So we are looking for classification functions of the type

$$C(x) = C(\lambda_1, S(x, x_1), \lambda_2, S(x, x_2), ..., \lambda_j, S(x, x_j), ...) \tag{1.1}$$

in which the objects $x_j \in L$ are members of the training set L and have labels l_j and in which x is the object to be classified. The traditional way to do this is the nearest neighbor rule, in this context often called template matching: Assign the object to the class of the nearest neighbor, i.e. the object with the highest similarity. The main drawback of this method is that it may require a large training set and thereby becomes computationally heavy. What is needed are condensing and editing techniques [9] for reducing the training set to a minimum subset and, moreover, a technique for building more general classification functions than maximum or minimum selectors.

Recently Vapnik proposed a support vector classifier [2] that computes a classification function on an automatically minimized training set, the *support set*. Although it is based on a vector space approach, it might be used for object similarity approaches as well. In this paper, we will go into the question whether a support *object* classifier based on Vapnik's support *vector* classifier might be useful for building featureless classifiers.

2. SUPPORT OBJECT CLASSIFICATION

Let $L = x_1, x_2, ..., x_m$ be a training set of objects with labels $\Lambda = \lambda_1, \lambda_2, ..., \lambda_m, \lambda_i \in \Omega$. Let $D(x_i, x_j)$ be a user defined distance measure, e.g. a simple measure like the Euclidean distance. More complicated measures can also be used provided that $D(x_i, x_j) = 0$ if and only if the objects x_i and x_j are identical. The traditional nearest neighbor classifier can be defined on these distances. A distance based classifier between two classes ω_1 and ω_2 can be defined as:

$$C(x) = \sum_{j=0}^{m} \alpha_j K(D(x, x_j)), \quad C(x) > 0 \text{ then } \omega_1, \text{ else } \omega_2 \tag{2.1}$$

in which $K(\cdot)$ is some potential function, e.g. $K(z) = \exp(-z/s)$, in which s is a free scaling factor. This is equivalent with the potential function approach as proposed more than 30 years ago by Aizerman et al. [5]. The coefficients α_j and the scaling parameters have to be optimized by the training procedure. The function $K(z)$ can be interpreted as a transformation from distances to similarities. It is also possible to define these classifiers directly on similarities: $S(x_i, x_j)$ if $S(x_i, x_j) = 1$ for $x_i = x_j$

and $S(x_i, x_j) \downarrow 0$ for decreasing similarity. So

$$C(x) = \sum_{j=0}^{m} \alpha_j \{S(x, x_j)\}^p, \quad C(x) > 0 \text{ then } \omega_1, \text{ else } \omega_2 \tag{2.2}$$

which defines a polynomial classifier of degree p. Note that the summations in 2.1 and 2.2 start for $j = 0$, referring to the constant contributions: $S(x_0, x) = 1, \forall x$.

For convenience we will restrict ourselves to similarity based classifiers. By using the right transformation, this covers distance based classifiers as well. A classifier like 2.2 has to be trained by optimizing the parameters α_j over the training set. Here the problem arises that there are as many parameters as there are objects in the training set. For a general set of objects this implies that the parameter values can always be given such values that all objects are classified correctly. For polynomial classifiers this has already been observed by Cover [6].

Vapnik has studied more generally the relation between classifier complexity and the size of the training set [1]. In his recent study [2] he follows an interesting approach in which simultaneously the classifier complexity is reduced by minimizing the set of training objects under consideration and the performance is maximized by optimizing the corresponding coefficients. Vapnik studies this approach for feature representations of objects in vector spaces. Here we will investigate the applicability to Hilbert spaces if just similarity matrices of objects are given.

There are several ways to do this. A simple criterion for two-class classifiers is

$$J_e = n_s/2 + n_e \tag{2.3}$$

In this expression n_s is the number of support objects that take part in 2.2 and n_e is the total number of erroneously classified objects over the entire training set. The first term can be interpreted as the classifier complexity contribution and the second term as the error contribution. This criterion demands a search over all combinations of training objects. For a given support set $L_s \subset L$, however, the computation of the classification function $C(x)$ is straightforward. Classifying all objects of the training set yields:

$$C(x_i) = \sum_{j=0}^{m} \alpha_j \{S(x_i, x_j)\}^p, \quad \forall x_i \in L \tag{2.4}$$

in which n_s is the size of L_s. If we demand that $C(x) = 1$ for $x \in \omega_1$ and $C(x) = -1$ for $x \in \omega_2$ and if these targets are summarized in a vector t, this can be rewritten as

$$t = \alpha S^p + \alpha_0 \tag{2.5}$$

The elements of the (n_s, n_s) matrix S are the similarities in the support set L_s. If rank$(S) < n$, α can directly be solved. It is possible, however, that the data (the set of similarities) is in a subspace causing S to be singular. In that case several solutions are possible. The Moore-Penrose pseudo-inverse defining the minimum norm classifier, may be used here as it is consistent with finding the most simple classifier. Moreover, it maximizes the object distances.

The search for the best set of support objects can be very time consuming. Vapnik [2] proposes a combined approach that automatically minimizes the support set while optimizing the weight vector α:

$$\alpha_{opt} = \arg\min_{\alpha}\{|\alpha| - \frac{1}{2}\alpha^T S\alpha\} \tag{2.6}$$

in which $|\alpha|$ is the sum of the coefficients α_j. See also [3]. By using a quadratic optimization procedure just those objects get values $\alpha_j \neq 0$ that are necessary for building the classifier.

This approach is particularly suited for finding classifiers in case a zero error solution exists. In case of class overlap it is always arbitrary how classification errors and object distances are combined in an optimization criterion. If for computational reasons another measure than an error count is used then certain distance measures and data distributions are favored.

Vapnik shows that the use of inner vector products for building the similarity matrix S, used in 2.4, 2.5 and 2.6 is consistent with determining polynomial classifiers in the original feature space. There is, however, no reason why we should not use differently constructed similarity matrices. As the relation with the feature vector space is lost this method should be called a support object classifier instead of a support vector classifier.

3. LEARNING POWER

We will assume that the classes are non-overlapping. For feature spaces this is a severe restriction. For object-distance representations, however, this is valid under the following two conditions, which are generally fulfilled:

1. The objects are unambiguously labeled.

2. Different objects have non-zero distances: $D(x_i, x_j) > 0$ if $x_i \neq x_j$ or: $S(x_i, x_j) < 1$ if $x_i \neq x_j$

The learning curve, the error ϵ of a classifier as a function of the size of the training set m can often be approximated by

$$\epsilon = am^{-V} \tag{3.1}$$

in which a is some arbitrary constant and V is the learning power. We will use this quantity for comparing different classifiers based on different (dis)similarity measures. In fig. 1(a) is shown what sizes of the training set are needed for a desired performance for given values of the learning power. Note that values of $V = 0.7$ and $V = 0.4$, for instance, result in training sizes that differ several orders of magnitude.

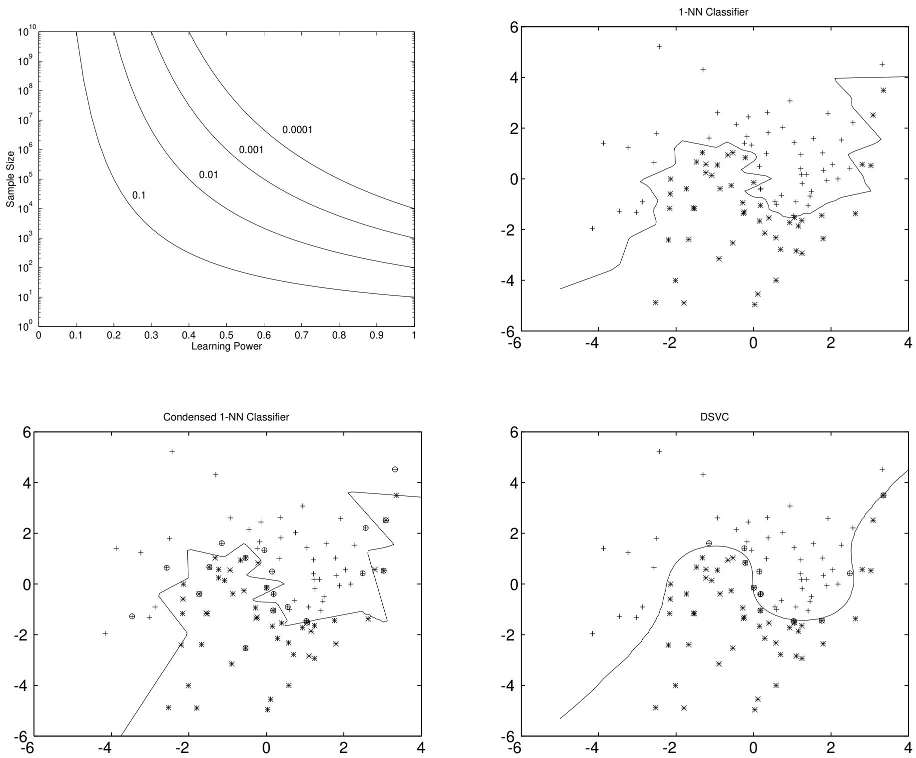

Figure 1: (a) Desired sample sizes as a function of learning power and accuracy, (b) NN-classifier for two non-overlapping classes, (c) Condensed NN-classifier, (d) Support object classifier

4. EXAMPLES

It is difficult to visualize datasets in infinite dimensional spaces. We will therefor fall back to a 2D feature space example. It is important, however, to realize that in this example just object differences are used. In fig. 1(b) the nearest neighbor classifier is shown for two non-overlapping classes. All 100 objects are used in order to compute this classifier.

A computationally more efficient classifier is obtained by condensing [9], see fig. 1(b). Here just 20 objects are used. The support object classifier based on 2.6 uses just 13 objects, see fig. 1(d). Moreover, it has, in this example, a better performance. In the next experiments the following classifiers will be used:

- NN: The nearest neighbor rule (template matching)
- CNN: The condensed nearest neighbor rule, i.e. using just those training samples that yield a zero error on the training set.
- SOCD: Support object classifier based on distances using 2.1 and 2.6.
- SOCS: Support object classifier based on similarities using 2.4, 2.5 and 2.6.

Method	V	e (m=100)	#sv (m=100)
NN	0.54	0.085	100
CNN	0.49	0.108	19
SOCD	0.59	0.056	29
SOCS	0.60	0.067	12

Table 1: Artificial dataset

In all experiments we used just 100 objects from each class resulting in distance (similarity) matrices of 200 x 200. This set was selected once and kept constant. The learning power was computed from 3.1 using averaged estimated errors over 50 experiments based on 60 and 100 arbitrarily selected training objects, using the remaining objects for testing. Results for the artificial classes of fig. 1(b) are summarized in table 1. In this table the mean value of the learning power (V) is given followed by the averages of the test errors and the numbers of support vectors that were found. We used $s = 0.1$ for the SOCD and $p = 3$ for the SOCS.

We will illustrate the proposed technique of featureless classification on real data, using the handprinted characters '3' and '8' from the NIST-3 database [10]. The raw data is given in binary images of 128 x 128 pixels. We used 32 x 32 subsampling. Two distance measures between characters are used: Hamming (counting the number of different pixels) and modified Hausdorff on the contour (mean nearest neighbor distance between contour points).

Results are summarized in the tables 2 and 3. We experimented with several values for s and p and discovered that performances and learning power may be highly dependent for these scaling factors. The best results are shown. It can be observed that in the first experiment the support object classifier performs much better than the nearest neighbor rule. In the contour distance experiment (table 3) the support object classifier performs somewhat worse. In this experiment the learning power is very low, indicating that it will be difficult to gain much performance by increasing the training set. More experiments will be presented elsewhere [4].

Method	V	e (m=100)	#sv (m=100)
NN	0.53	0.086	100
CNN	0.44	0.14	21
SOCD	0.72	0.036	64
SOCS	0.70	0.040	65

Table 2: Character recognition using Hamming distances

Method	V	e (m=100)	#sv (m=100)
NN	0.24	0.048	100
CNN	0.26	0.075	13
SOCD	0.17	0.055	43

Table 3: Character recognition based on contours

5. DISCUSSION

The main purpose of this study is to argue and illustrate that it is possible to build classifiers on object (dis)similarities. This opens a new type of applications in which feature representations are replaced by distance measures. This has several consequences:

1. The type of application knowledge for specifying features might be entirely different from the knowledge to define distance measures. In some areas feature descriptions do not arise naturally. Character recognition might be a good example as during the years many different types of features have been proposed and tried. Distance measures might be a good alternative.

2. While we leave the vector space approach, we also leave the possibility of using density functions and thereby we the Bayes theory. A new type of probabilistic theory has to be developed, if possible.

3. The support object classifier we used reduces the training set to a small number of essentially needed examples. These support vectors are really different from the classically used prototypes. Prototypes can be considered as cluster centers: typical examples. Support vectors support the classification boundary, they are the typical boundary objects: the last objects before a new class region is entered. It is thereby to be expected that the support objects are close to confusion. Erroneously labeled objects and outliers are likely to become support objects. In applying the support object classifier it might be advantageous to reconsider the labeling of the support vectors.

4. Experiments with the support object classifier indicate that it may have a large learning power compared with feature based approaches. It thereby may be a good classifier for small sample size problems, especially in case of weakly defined features.

(Received Oktober 15, 1997.)

REFERENCES

[1] V. Vapnik 1982. Estimation of Dependences based on Empirical Data. *Springer-Verlag*, New York.
[2] V.N. Vapnik 1995 The nature of statistical learning theory, *Springer Verlag*, Berlin.
[3] D.M.J. Tax, D. de Ridder, and R.P.W. Duin 1997. Support vector classifiers: a first look, *ASCI'97, Proc. Third Annual Conference of the Advanced School for Computing and Imaging.*
[4] R.P.W. Duin, D. de Ridder, D.M.J. Tax 1997. Experiments with object based discriminant functions; a featureless approach to pattern recognition, Pattern Recognition in Practice V, Vlieland 1997, to be published in Pattern Recognition Letters.
[5] M.A. Aizerman, E.M. Braverman, and L.I. Rozonoer 1964. The probability problem of pattern recognition learning and the method of potential functions. *Automation and Remote Control*, vol. 25, September 1964, 1175-1193.
[6] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, *IEEE Trans.Elec.Comp*, vol. EC-14, 1965, 326-334.
[7] C.R. Rao 1946. Tests with discriminant functions in multivariate analysis. *Sankhya: The Indian Journal of Statistics*, vol. 7, 407 - 414.
[8] A.K. Jain and B. Chandrasekaran, Dimensionality and Sample Size Considerations in Pattern Recognition Practice, in: P.R. Krishnaiah and L.N. Kanal (eds.), Handbook of Statistics, vol. 2, North-Holland, Amsterdam, 1987, 835 - 855.
[9] P.A. Devijver and J. Kittler 1982. *Pattern Recognition: a Statistical Approach*, Prentice Hall, London.
[10] C.L. Wilson, M.D. Marris 1990. Handprinted character database 2, april 1990. National Institute of Standards and Technology; Advanced Systems division.

Robert P.W. Duin, Pattern Recognition Group, Faculty of Applied Physics, Delft University of Technology, P.O.Box 5046, 2600 GA Delft, The Netherlands