FEATURELESS PATTERN 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 definition 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 [12].

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 a priori grounds (i. e. on the physical nature of the objects) features have to be defined that are strongly related to 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 or as the peaking phenomenon [7] makes it necessary to have enormous numbers of training examples available for large feature sizes. Simple rules of thumb demand something like ten times the feature size. Worst case approaches based on the VC dimension [11] demand for almost all classifiers exponentially increasing training sets. Consequently much research is done in finding small sets of good features on a priori grounds or in statistical techniques to reduce initially too large feature sets.

In this study the possibility will be re-investigated 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 for good features but directly use a similarity measure $S_x(x_i, x_j)$ between objects

[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).

x_i and x_j (these are not feature vectors but just symbolic representations of the objects). 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

$$C(x) = C(\lambda_1, S(x, x_1), \lambda_2, S(x, x_2), \ldots, \lambda_j, S(x, x_j), \ldots) \qquad (1.1)$$

in which the objects $x_j \in L$ are members of the training set L and have labels λ_j and in which x is the object to be classified. Traditionally this is done by the nearest neighbor rule, in`this context often called template matching: Assign the object to the class of its nearest neighbor, i.e. the object with the highest similarity. Its main drawback is that in case of a large training set it becomes computationally heavy. What is needed are condensing and editing techniques [2] 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 [12], see also [9], 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 discuss whether a support *object* classifier based on Vapnik's support *vector* classifier might be useful for building featureless classifiers. Parts of this paper have been presented before [4, 5, 6].

2. SUPPORT OBJECT CLASSIFICATION

Let $L = x_1, x_2, \ldots, x_m$ be a training set of objects with labels $\Lambda = \lambda_1, \lambda_2, \ldots, \lambda_m, \lambda_i \in \Omega = \{\omega_1, \omega_2\}$. Let $D(x_i, x_j)$ be a user defined distance measure, e.g. a simple measure like the Euclidean distance between pixel representations. 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 nearest neighbor rule can be based on these distances. A distance based classifier between two classes ω_1 and ω_2 can be defined as:

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

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 [1]. 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

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

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 appropriate 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.

Vapnik has studied more generally the relation between classifier complexity and the size of the training set [11]. In his recent study [12] he follows an interesting approach in which simultaneously the classifier complexity is reduced by minimizing the set of training objects 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 $C(x)$ using 2.2 is straightforward. 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

$$t = \alpha S^p + \alpha_0. \tag{2.4}$$

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 [12] proposes a combined approach that automatically minimizes the support set while optimizing the weight vector α:

$$\alpha_{opt} = \arg\min_{\alpha} \left\{ |\alpha| - \frac{1}{2}\alpha^T S \alpha \right\} \tag{2.5}$$

in which $|\alpha|$ is the sum of the coefficients α_j. See also [10]. 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 2.2, 2.4 and 2.5 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. See also [9]. 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.

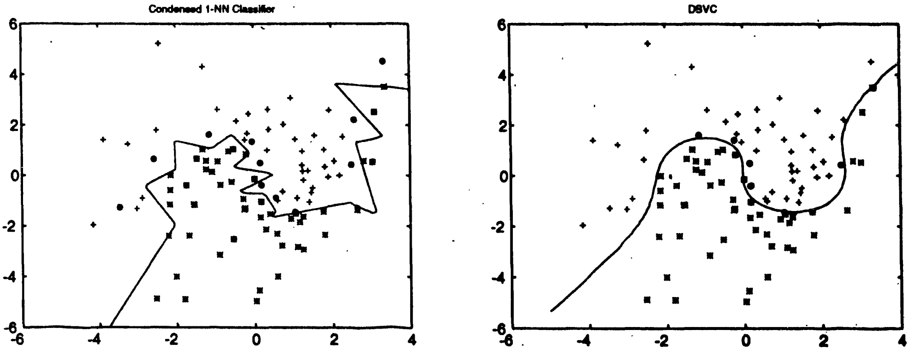

Fig. 1. (a) Condensed NN-classifier, (b) Support object classifier.

3. EXAMPLES

It is difficult to visualize datasets in infinite dimensional spaces. We will therefor use a 2D feature space example. It is important, however, to realize that in this example just object differences are used. In Figure 1 (a) the condensed nearest neighbor classifier [2] is shown for two non-overlapping classes. This classifier is computationally more efficient as it uses less objects (here just 20 out of 100). Condensing, however, might increase the generalization error.

The support object classifier based on 2.5 uses just 13 objects, see Figure 1 (b). Moreover, it has, in this example, a better performance. 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.
SOC: Support object classifier based

We used just 100 objects from each class and selected at random half for training and half for testing. Averaged results over 50 experiments are presented in Table 1.

The proposed technique of featureless classification will be illustrated on real data, using the hand-printed characters '0' to '9' from a NIST database [13]. The raw data is given in binary images of 128×128 pixels. We also investigated subsampled characters as well as normalization on mean, size, skewness and line-width. Two

Table 1. Artificial dataset.

Method	error	#sv
NN	0.085	100
CNN	0.108	19
SOC	0.056	29

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). 200 characters per class were used, randomly separated into 100 for training and 100 for testing.

The averaged results over 100 experiments are summarized in Table 2. These results are not optimized for p. We found however, that this scaling parameter might highly influence the results [5]. It can be observed that the support object classifier performs similar to the nearest neighbor rule. Condensing of the nearest neighbor rule, however, deteriorates the performances.

Table 2. Character recognition errors and support set sizes.

Data	NN	CNN	#sv	SOC	#sv
128*128	0.412	0.435	54	0.310	88
64 * 64	0.420	0.451	55	0.322	88
32 * 32	0.448	0.473	57	0.343	86
16 * 16	0.583	0.619	69	0.521	75
Normalized	0.129	0.220	33	0.130	73
Contours	0.160	0.242	37	0.149	33

4. DISCUSSION

The goal 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:

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.

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

The support object classifier we used reduces the training set to a small number of essentially needed examples. These support objects 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.

(Received December 18, 1997.)

REFERENCES

[1] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer: The probability problem of pattern recognition learning and the method of potential functions. Automat. Remote Control *25* (1964), 1175-1193.

[2] P. A. Devijver and J. Kittler: Pattern Recognition: A Statistical Approach. Prentice Hall, London 1982.

[3] R. P. W. Duin: Small sample size generalization. In: SCIA'95, Proc. 9th Scandinavian Conf. on Image Analysis (G. Borgefors, ed.), Volume 2, Uppsala 1995, pp. 957–964.

[4] R. P. W. Duin and D. de Ridder: Neural network experiences between perceptrons and support vectors. In: Proc. of the 8th British Machine Vision Conference (A. F. Clark, ed.), Volume 2, Colchester 1997, pp. 590–599.

[5] R. P. W. Duin, D. de Ridder and D. M. J. Tax: Experiments with object based discriminant functions; a featureless approach to pattern recognition. In: Pattern Recognition in Practice V, Vlieland, 1997, to be published in Pattern Recognition Letters.

[6] R. P. W. Duin, D. de Ridder and D. M. J. Tax: Featureless Classification. In: Proc. 1st International Workshop Statistical Techniques in Pattern Recognition (P. Pudil, J. Novovičová and J. Grim, eds.), Prague 1997, pp. 37–42.

[7] A. K. Jain and B. Chandrasekaran: Dimensionality and sample size considerations in pattern Recognition practice. In: Handbook of Statistics (P. R. Krishnaiah and L. N. Kanal, eds.), Vol. 2, North–Holland, Amsterdam 1987, pp. 835–855.

[8] S. Raudys: Evolution and generalization of a single neurone. I. Single layer perceptron as seven statistical classifiers. Neural Networks, to be published.

[9] B. Schölkopf: Support Vector Learning. Ph.D. Thesis, Techn. Universität Berlin 1997.

[10] D. M. J. Tax, D. de Ridder and R. P. W. Duin: Support vector classifiers: a first look. In: ASCI'97, Proc. Third Annual Conference of the Advanced School for Computing and Imaging, 1997.

[11] V. N. Vapnik: Estimation of Dependences Based on Empirical Data. Springer–Verlag, New York 1982.

[12] V. N. Vapnik: The Nature of Statistical Learning Theory. Springer–Verlag, Berlin 1995.

[13] C. L. Wilson, M. D. Marris: Handprinted Character Database 2. National Institute of Standards and Technology; Advanced Systems division, 1990.

Robert P. W. Duin, Pattern Recognition Group, Faculty of Applied Physics, Delft University of Technology, P. O. Box 5046, 2600 GA Delft. The Netherlands.
e-mail:bob@ph.tn.tudelft.nl

Dick de Ridder and David M. J. Tax, Faculty of Applied Sciences, Delft University of Technology, P. O. Box 5046, 2600 GA Delft. The Netherlands.