ELSEVIER

Pattern Recognition Letters 18 (1997) 1159–1166

Pattern Recognition Letters

Experiments with a featureless approach to pattern recognition

Robert P.W. Duin *, Dick de Ridder, David M.J. Tax

Faculty of Applied Physics, Delft University of Technology, Delft, The Netherlands

Abstract

Traditionally automatic pattern recognition is based on learning from examples of objects represented by features. In some applications it is hard to define a small, relevant set of features. At the cost of large learning sets and complicated learning systems discriminant functions have to be found. In this paper we discuss the possibility to construct classifiers entirely based on distances or similarities, without a relation with the feature space. This is illustrated by a number of experiments based on the *support object classifier* (Duin et al., 1997), a derivative of Vapnik's *support vector classifier* (Cortes and Vapnik, 1995). © 1997 Elsevier Science B.V.

Keywords: Support vector classifier; Featureless recognition; Character recognition; Hilbert space

1. Introduction

Almost the entire field of statistical pattern recognition is based on the feature vector representation. This is a straightforward way to implement expert knowledge of measurable discriminative properties of the classes to be distinguished. In case this knowledge is not available this is often compensated by considering large numbers of possibly useful features.

A consequence of the lack of knowledge is that the dimensionality of the feature space increases. For most of the traditional methods this implies that large training sets are needed (curse of dimensionality, see (Jain and Chandrasekaran, 1987)). Intuitively this is the price one has to pay: if knowledge lacks, more examples have to be supplied.

One of the ways to overcome the problem of missing feature knowledge is to sample the object and use these samples as features. Now several normalizations might be necessary to obtain invariance for transformations like translation, rotation, resizing, etcetera. From this point of view, however, the demand for larger training sets for increasing feature sizes, here sample sizes, is counterintuitive. Why are more examples needed if in the character recognition problem the characters are not represented by 6×6 pixels, but by 1024×1024 pixels? A 36-dimensional feature space may reasonably be filled by a few thousand characters. There are, however, no feasible ways to obtain training sets large enough to fill a space with a dimensionality of about a million.

In this paper the possibility will be discussed of using a (dis)similarity representation instead of a feature vector representation. So the training set is represented by a similarity matrix and new objects are classified on the basis of their similarity with

* Corresponding author.

PII S0167-8655(97)00138-4

objects in the training set only. This way of classification using a given set of prototypes is traditionally known as template matching. Recently we discussed how these object similarities can be combined into a classification function and how such a classifier can be trained (Duin et al., 1997). This was based on the support vector classifier (SVC) developed by Cortes and Vapnik (1995) and Vapnik (1995), which we already studied experimentally (Tax et al., 1997). The SVC is still based on feature spaces, but has two interesting properties which we will briefly discuss.

First, as it is defined on vector dot products between test objects and training objects, its complexity (say the number of parameters in case of polynomial classifiers) is independent of the dimensionality.

Second, the complexity of the SVC is optimized with respect to the training set by reducing the so-called support set, i.e., the subset of the training set that is actually used for the classifier. This reduction might be compared with the condensing procedure for the nearest neighbor rule (NNR). In both cases the memory demand and the computational complexity are reduced. The generalization error, however, is expected to decrease by the reduction process in the SVC, while in case of the NNR reduction it has an unpredictable, but often increasing effect on the error.

For the SVC the feature size might grow to infinity as long as the vector dot products remain finite. There exists a clear mathematical foundation for this property. We thereby have a classifier that can operate in Hilbert space. In our proposal (Duin et al., 1997) we replace the dot products by arbitrarily defined similarities and thereby loose the relation with the feature vector space. What we gain on the other side, is a classifier that can be used in connection with any procedure for measuring (dis)similarities between objects. It is insensitive to the number of samples (pixels) used to measure these, except for a possible effect on the digitalization error. What we expect therefore is, the more samples the better, instead of the curse of dimensionality, causing some training set size dependent optimum.

As our classifier is a straightforward extension of the SVC from a vector representation to an object representation it is called the Support Object Classifier (SOC). An important issue for both, the SVC as well as the SOC, is the question whether the classes are overlapping or not. It is not possible to estimate a probability density function in an infinite-dimensional space. This is the reason behind the curse of dimensionality. The support classifiers not only avoid that, but are not based on the underlying Bayes theory at all. They are optimized by the minimum description length criterion instead of by criteria based on error minimization or on likelihood maximization. Consequently it is difficult to handle class overlap. The optimization criterion that has been used by Vapnik (1995) for that purpose is in our opinion rather ad hoc. For the moment we prefer to restrict ourselves to non-overlapping classes. In contrast to feature based approaches, this is not a severe restriction for classifiers based on object distances. It simply demands that two objects with distance zero always belong to the same class. If the distance measure is such that different objects cannot have a zero distance, the demand is fulfilled if different classes cannot generate objects that are entirely similar.

In Section 2 we summarize the concept of object distance based classifiers. In Section 3 this is followed by an illustration using a set of character recognition experiments. We believe that these experiments clearly indicate the potential power of the object distance approach. In Section 4 we elaborate further on this and formulate some additional research questions.

2. Object based discriminants

Consider the linear classifier in a k-dimensional feature space R_k: $S(\boldsymbol{x}) = \boldsymbol{w}.\boldsymbol{x} + w_0 = W^{\mathrm{T}}X$. We will consider here the very small sample size problem in which the number of training samples m is less than the feature size, so $m < k$. Thereby the minimum norm W points into the subspace R_m defined by these samples and can be written as $W = \sum_i \alpha_i X_i$. Minimizing the mean square error in the training set,

$$\sum_j WX_j - y_j = \sum_j \left\{ \sum_i \alpha_i X_i X_j - y_j \right\}, \tag{1}$$

yields $\alpha = K^{-1} y$, with $K_{ij} = X_i^{\mathrm{T}} X_j$. This linear classifier is equivalent with Fisher's linear discriminant based on the Moore–Penrose pseudo-inverse, see also (Duin, 1995; Skurichina and Duin, 1996). As $m < k$ a zero-error solution is obtained. In these references it is also shown that if the training set is further selectively reduced a classifier with a larger generalization capability (smaller error on the test set) might be constructed based on just the training objects close to the margin between the classes. This procedure, based on an iterative reduction of the set of objects used in Eq. (1) is called the Small Sample-size Classifier (SSC).

The SVC developed by Cortes and Vapnik (1995) and Vapnik (1995) is based on just a slightly different viewpoint, resulting in an almost identical classifier. They add, however, two important observations. First it is noted that by almost the same procedure polynomials and other nonlinear classifiers can be found if the elements in the inner product matrix K are replaced by $K_{ij} = (X_i^{\mathrm{T}} X_j)^n$ for order-n polynomials and even by $K_{ij} = f(X_i^{\mathrm{T}} X_j)$ for a restricted set of functions like sigmoids. This can be done without increasing the complexity of the classifier as this is still determined by the set of coefficients stored in $\alpha = K^{-1} y$. Their number is equal to the reduced number of training objects: the support vectors.

The generalized support vector classifier as described above, opens an entirely new area for discriminant analysis. As it is solely based on inner products of feature vectors it also holds for objects represented by an increasing or possibly infinite number of features, i.e., a continuous description. Until now objects like non-digitized photographs (characters, faces) or time signals (speech) could only be handled by template matching techniques. This new approach makes it possible to build discriminant functions (linear as well as nonlinear) based on inner products defined as

$$K_{ij} = \left(X_i^{\mathrm{T}} X_j \right)^n = \int X_i(t)\, X_j(t)\, \mathrm{d}t, \tag{2}$$

in which t is a one or more dimensional parameter defining the object domain (e.g., time, 2D or 3D space).

We generalize this approach one step further. The whole procedure still works if the inner product matrix K is replaced by a similarity matrix K of which the elements are defined using some arbitrary similarity measure between object pairs. So $K_{ij} = S(X_i, X_j)$. For instance, for any distance measure D, e.g.,

$$D_{ij} = \int X_i(t) - X_j(t)^2\, \mathrm{d}t, \tag{3}$$

we can define a similarity $S = \exp(-\gamma D)$ in which γ is some normalization value.

By this process the correspondence with the feature space classifiers is now lost. What is gained, however, is that herewith a discriminant analysis approach is defined based solely on similarity measures.

For our implementation of the SOC we experimented with the above mentioned iterative minimization of the support set used in the SSC. In addition the quadratic minimization procedure proposed by Vapnik (1995) was used which automatically minimizes the support set while optimizing the weight vector α,

$$\alpha_{\mathrm{opt}} = \arg\min_{\alpha} \left\{ |\alpha| - \tfrac{1}{2} \alpha^{\mathrm{T}} S \alpha \right\}, \tag{4}$$

in which $|\alpha|$ is the sum of the coefficients α_j. In this procedure only those objects that are necessary for building the classifier obtain values $\alpha_j \neq 0$.

In preliminary experiments, the minimization of Eq. (4) appeared to be faster and to yield smaller support sets on the average compared to the iterative procedure. However, sometimes the minimization did not start due to accuracy problems. In the remaining of this paper we used Eq. (4).

Several other classifiers are also possible starting from a (dis)similarity matrix K. We mention:

1. The Pseudo-Fisher linear discriminant directly based on Eq. (1), using arbitrary similarities and using the entire training set. We will call this the Generalized Pseudo-Fisher Discriminant (GPFD).
2. The Nearest Neighbor Rule (NNR).
3. The Condensed Nearest Neighbor Rule (CNNR). In our implementation we minimized the support set under the condition of a constant classification (no label change) of the original training set using a leave-one-out approach.
4. Kernel based approaches in which objects are assigned to the class for which the sum of all kernels or the maximum of any kernel is the largest. This is possible as kernels can be entirely based on distances matrices. This method resembles the use of Parzen estimators. It differs from

it, however, in the sense that no densities are estimated.

3. Experiments

The above will be illustrated by a number of experiments on the recognition of handwritten numerals. We will use the NIST-3 database (Wilson and Marris, 1990) which contains about 2000 samples for each of the 10 classes 0,1, . . . ,9. In the raw data we used, the characters are represented in binary 128 × 128 images. In one of the experiments we also used the normalization software supplied with this dataset. It normalizes for position, size, angle and line-width, resulting in 16 × 16 gray value images.

Previous experiments with this dataset resulted in a classification error of about 1% using 1000 objects per class for training. As we intended to set up a large range of experiments using different types of distance measures and classifiers we had to restrict ourselves to small training sizes for computational reasons. This was also necessary due to the nature of the input data format of our methods: (dis)similarity matrices. A training set of 1000 objects per class yields for 10 classes matrices of $10000 \times 10000 =$ 100 million entries. This is still possible. Matrix inversion, however, which has to be done iteratively, is prohibitive.

For these reasons we restricted ourselves to datasets of just 20 samples per class from which we generated randomly training sets of 10 samples per class. Most of our experiments were repeated 100 times and we report the average error of test sets of 100 samples.

The following classifiers are used, see also Section 2.

- NNR: Nearest neighbor rule.
- CNNR: Condensed nearest neighbor rule. This classifier was used to obtain a reference for the size of the support sets.
- SOC: Support object classifier. For the 10-class problem we compute a classifier between each class and all other classes. Objects are assigned to the class for which the classifier shows the largest outcome. We did not try to combine the 10 support sets. In the tables below the average support set sizes over all 10 sets are reported.
- GPFD: Generalized Pseudo-Fisher Discriminant. This is the same classifier as SOC but now based on all objects, i.e., no support set reduction.

In studying the following experiments one should realize that our training sets are very small (10 objects per class) and that for a 10-class problem the a priori probability of error is 0.9. The best results reported for this data show an error of about 2.5%. This has been obtained using large neural networks. The results for the Nearest Neighbor rule (NNR) for training sizes of 1000 objects per class are close to this result, indicating dat for these sample sizes the NNR is close to optimal, see (De Ridder, 1996). We performed the following experiments.

3.1. Experiment 1. Resolution

In this experiment we computed distances by Eq. (3) for the NNR and the CNNR and similarities by Eq. (2) for GPFD and SOC on the raw 128 × 128 binary images and on several sub-samplings of that, see Table 1. We added the result on the 16 × 16 normalized gray value representations. From this table it can be concluded that the SOC performs much better than the NNR. The normalization makes the SOC and the NNR perform almost equal. For the SOC, however, it is much better to use the raw data instead.

Table 1
Mean classification error and support set sizes for various resolutions of binary characters

Resolution	NNR error	CNNR error	CNNR size	GPFD error	SOC error	SOC size
128 × 128	0.412	0.435	54	0.314	0.310	88
64 × 64	0.420	0.451	55	0.323	0.322	88
32 × 32	0.448	0.473	57	0.359	0.343	86
16 × 16	0.583	0.619	69	0.562	0.521	75

Table 2
Mean classification error and support set sizes for various bounding box normalizations of 128 × 128 binary characters

Normalisation (128 × 128)	NNR error	CNNR error	CNNR size	GPFD error	SOC error	SOC size
No	0.412	0.435	54	0.314	0.310	88
Mean	0.381	0.415	52	0.299	0.304	88
Size	0.342	0.390	41	0.485		
Skewness, linewidth	0.129	0.220	33	0.146	0.130	73

3.2. Experiment 2. Normalization

Here we studied a simple normalization of the raw 128 × 128 data. In the mean normalization the bounding boxes of the characters were given equal means. In the size normalization they were given equal sizes as well by resampling. Finally a normalization on skewness and linewidth was used in addition to mean and size, resulting in 16 × 16 grey value images, see (De Ridder, 1996). Table 2 shows that these normalizations are useful in all circumstances with one exception. In the case of size normalization the GPFD gives a bad result and the SOC could even not be computed due to an unstable minimization procedure. This will be discussed below.

3.3. Experiment 3. Contours

In this experiment we computed distances using contours, equi-distantly sampled on 128 points. The modified Hausdorff (Dubuisson and Jain, 1994) distance is used in the NNR and similarities are computed according to $S = \exp(-\gamma D)$ with $\gamma = \mathrm{var}(D)^{1/2}$. Here we also included normalizations in order to obtain invariance to mean, size (contours are given equal variances in both directions) and rotation (using an angular contour description and rotating over all starting points). The latter normalizations include the first. The figures in Table 3 show that the NNR gives much better results and now outperforms the SOC in most cases. Note that bad results for the SOC correspond with a large support set. Obviously the similarity matrix we computed does not behave well. Therefore we decided to study the influence on the performance of monotonic transformations of the similarity matrix in a new experiment.

3.4. Experiment 4. Contours, different similarities

Now similarities are computed using $S = \exp(-sD/\gamma)$ for the contour distances including size normalization, compare Table 3. Again, γ is the standard deviation of all values in D. This is varied, however, with a multiplicative scaling factor s, which corresponds to the polynomial degree in the finite vector spaces studied by Vapnik (1995).

In Figs. 1 and 2 the average classification error and the average size of the support set are given as a function of the scaling factor. Especially the size of the support set is very sensitive to scaling. This corresponds to the minimizations problems we had in some of the experiments in Experiment 2. It is clear that the overall use of the value $s = 1$ for the scaling factor (i.e., use just the standard deviation of

Table 3
Mean classification error and support set sizes for 32 point contour normalizations of 128 × 128 binary characters

Normalisation	NNR error	CNNR error	CNNR size	GPFD error	SOC error	SOC size
No	0.261	0.337	47	0.682	0.327	100
Mean	0.230	0.300	44	0.524	0.268	100
Size	0.160	0.242	37	0.149	0.138	33
Rotation	0.311	0.398	50	0.276	0.281	97

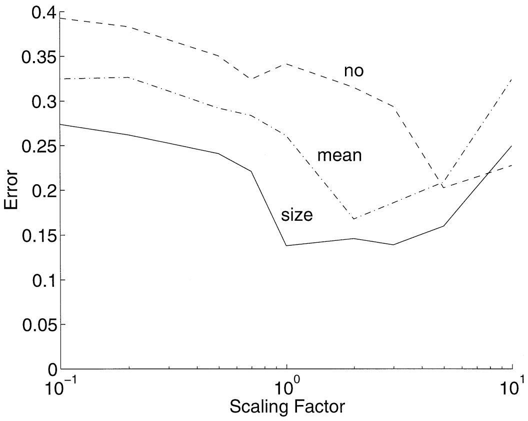

Fig. 1. The average error as a function of the scaling factor for various normalizations.

D for normalization) is a too simple method. Something more sophisticated has to be developed.

Finally, for a set of values of *s* the average (here over 25 experiments) error was computed as a function of the training size, see Fig. 3. In this experiment only the characters ‘‘3’’ and ‘‘8’’ are used. The sensitivity of the scaling factor demonstrates itself clearly, again.

More interesting is that here a deterioration of the performance can be observed for increasing sample sizes similar to what has been found earlier for the pseudo-Fisher classifier (Duin, 1995). An explanation for the case of Gaussian distributed feature vector data given by Raudys and Duin (1997) is based on the large noise for classification problems in which the feature size equals the sample size. It has to be investigated further how that relates to the scaling factor in the example discussed here.

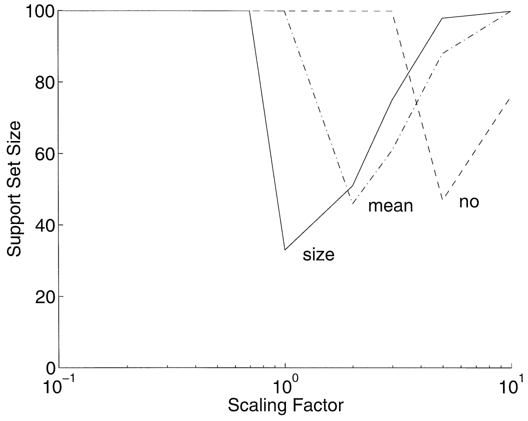

Fig. 2. The average size of the support set as a function of the scaling factor for various normalizations.

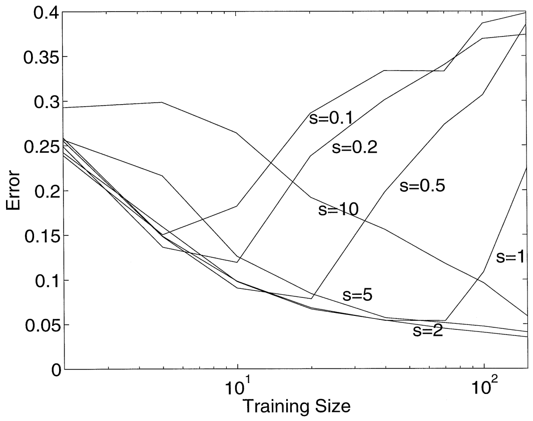

Fig. 3. Mean classification error as a function of the training size in discriminating size-normalized 128 point contour representations of 128 × 128 binary characters ‘‘3’’ and ‘‘8’’ as a function of the scaling factor *s*.

4. Conclusions

Object based discriminant analysis will be of importance in applications where no natural discriminative features are given, but instead some object similarity measure can be supplied. It thereby widens the application area of statistical (in the sense of data based) pattern recognition. Other, different types of expert knowledge will be needed or might be used. The automatic search for good features is replaced by the search for the optimal set of support objects: the objects close to the discriminant boundary that define its final position. The guidelines and criterion functions given by Vapnik (1995) are certainly helpful, but, as we observed, might yield some numerical and computational problems.

One of the most promising aspects of object based discriminant analysis is that it removes the need for small sets of good features in case of small training sets. For well separable classes well performing classifiers might be based on just a few tens of training samples, provided that a useful similarity measure can be supplied.

Several questions arise at the threshold of this new area. What is the performance compared to feature based approaches? How do we define similarity measures that are invariant for shifts, rotations, and for particular types of object deformations? How fast is it? The support vector classifier does not make use of Bayes rule, explicitly nor implicitly. Is hereby the need for a randomly generated training set removed? Is it useful if more boundary objects are supplied? This paper is just a first acquaintance with this new area. Much has to be done. In any case, it seems to promise an increased applicability of the pattern recognition field in practical problems.

For further reading, see (Aizerman et al., 1964; Fogelman-Soulie et al., 1993).

Discussion

Mao: If your similarity measure is replaced by a radial basis function, then your object classifier is equivalent to a radial basis function network from the functional point of view.

Duin: But where is the radial basis function defined; in what type of space, in your proposal?

Mao: In the feature space.

Duin: Yes, but my proposal includes featureless representations, so this is somewhat more general.

Mao: So, do you consider a radial basis function network as a special case of your support object classifier?

Duin: Yes, but a radial basis function network does not have this support object idea, where you discard unnecessary objects. A radial basis network may have its kernels in any place.

Mao: Suppose I use the same training algorithm as your support object classifier and I select the locations of the prototype as support objects.

Duin: Yes, then we are back to one of the variants proposed already by Vapnik. He also proposes, or he mentions just the possibility of using radial basis functions in his support vector classifier. And I wanted to make the extension to support objects, where I leave the feature space.

Kappen: How do the distance measures that you propose improve the performance compared to the original Vapnik approach that just takes the inner product?

Duin: But then you have to go back to the feature space. And then you have to define what features you are using.

Kappen: Well, the inner product of the training patterns perhaps.

Duin: Yes, but in that case, the training patterns have to be defined in the feature space. I did not deal with feature spaces, but in this talk the numbers are based on an exclusive-or distance, so if we define as our features the pixels, then we can compare this with the Vapnik approach. If he defines his features as pixels then the support object classifier is identical to the support vector classifier. But that is no longer possible for the contours, because here we are using the modified Hausdorff distances between contours which are not represented by features. So then it is no longer equivalent to a feature based approach.

Kappen: So I should compare the SOC using exclusive-or distances to the SOC based on the modified Hausdorff distances.

Duin: Yes, for instance, but that is of course not completely fair, because the first one is based on the pixels used as features and here I use contours as a representation and then use a distance representation on the basis of contours. So, you can try to do a comparison, but finally I wanted to present a method that also works in cases where we don’t have features, but just have some type of a distance measure between objects.

Kappen: Perhaps you can explain again how this scaling factor enters into the selection of your support set.

Duin: The scaling factor is a number similar to the power, the degree of the classifier in the Vapnik approach. So, if you go back to the feature space than this scaling factor defines the degree of the classifier.

Raghavan: I would like to say two things. In our work on image retrieval, our goals are somewhat similar to yours, except that we did not call that featureless object recognition. I think that the expression ‘‘featureless’’ is somehow disturbing because I think it is very difficult to say where the deviding line is between what is a feature and what is not. Do you think that if I use a bitmap representation it is not using features, but if I use intensities it is using features? Another point is I think that you should also look at the earlier work like the work of Lev Goldfarb. He has some very interesting results in this area.

Duin: I think there is a clear distinction between what I call a featureless approach and a feature vector approach. If you use a representation of the objects in a feature space, for me that is a feature vector approach. If you try to avoid that representation by a direct measurement of the distances between objects, then you no longer have a representation in a feature space and you do not have the problems of feature spaces such as a too high dimensionality and having to estimate large numbers of parameters, or having to deal with large neural networks. So for me there is a clear distinction between feature-based and non-feature based distances.

Raghavan: The work of Goldfarb, for example, tries to make a link between syntactic pattern recognition and statistical pattern recognition. In his work he uses syntactic approaches, defining primitives and using distances between primitive strings. By your definition you seem to be restricting yourself too much; I think that you could have much broader ways to obtain those initial distances.

Duin: I agree, it can be further generalized, you can define distances based on primitives. But then again you are back to the problem of defining primitives, in some way or another. I wanted to start from a point where some expert knows, for instance from the physics of the problem, what a natural distance measure between objects is. For that type of application I am trying to find a classifier.

References

Aizerman, M.A., Braverman, E.M., Rozonoer, L.I., 1964. The probability problem of pattern recognition learning and the method of potential functions. Automation and Remote Control 25, 1175–1193.

Cortes, C., Vapnik, V., 1995. Support-vector networks. Machine Learning 20 (3), 273–297.

De Ridder, D., 1996. Shared weights neural networks in image analysis. Master Thesis.

Dubuisson, M.P., Jain, A.K., 1994. A modified Hausdorff distance for object matching. In: Proc. 12th IAPR Internat. Conf. on Pattern Recognition, Conference A, Jerusalem, Israel, 9–13 October 1994, IEEE Computer Society Press, Los Alamitos, CA, pp. 566–568.

Duin, R.P.W., 1995. Small sample size generalization. In: Borgefors, G. (Ed.), SCIA’95, Proc. 9th Scandinavian Conference on Image Analysis, Vol. 2, Uppsala, 6–9 June 1995, pp. 957–964.

Duin, R.P.W., De Ridder, D., Tax, D.M.J., 1997. Featureless classification. In: Proc. Workshop on Statistical Pattern Recognition, Prague, June 1997.

Fogelman-Soulie, F., Viennet, E., Lamy, B., 1993. Multi-modular neural network architectures: Applications in optical character and human face recognition. Internat. J. Pattern Recognition and Artificial Intelligence 7 (4), 721–755.

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

Raudys, S., Duin, R.P.W., 1997. On expected classification error of the Fisher Linear Classifier with pseudo-inverse covariance matrix, submitted.

Skurichina, M., Duin, R.P.W., 1996. Stabilizing classifiers for very small sample sizes. In: Proc. ICPR13, Vienna, Austria, 25–29 August 1996, Vol. 2, Track B, pp. 891–896.

Tax, D., De Ridder, D., Duin, R.P.W., 1997. Support vector classifiers: A first look. In: ASCI’97, Proc. 3rd Ann. Conf. of the Advanced School for Computing and Imaging, 1997, submitted.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer, Berlin.

Wilson, C.L., Marris, M.D., 1990. Handprinted character database 2. National Institute of Standards and Technology, Advanced Systems Division.