PERGAMON

Pattern Recognition 33 (2000) 1475–1485

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Combining multiple classifiers by averaging or by multiplying?

David M.J. Tax[a,*], Martijn van Breukelen[a,1], Robert P.W. Duin[a], Josef Kittler[b]

[a]*Pattern Recognition Group, Department of Applied Physics, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands*
[b]*Department of Electronic and Electrical Engineering, University of Surrey, Guildford, Surrey GU2 5XH, UK*

Received 25 January 1999; received in revised form 1 June 1999; accepted 1 June 1999

Abstract

In classification tasks it may be wise to combine observations from different sources. Not only it decreases the training time but it can also increase the robustness and the performance of the classification. Combining is often done by just (weighted) averaging of the outputs of the different classifiers. Using equal weights for all classifiers then results in the mean combination rule. This works very well in practice, but the combination strategy lacks a fundamental basis as it cannot readily be derived from the joint probabilities. This contrasts with the product combination rule which can be obtained from the joint probability under the assumption of independency. In this paper we will show differences and similarities between this mean combination rule and the product combination rule in theory and in practice. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Combination of classifiers; Classifier fusion; Neural networks; Handwritten digits recognition; Pattern recognition

1. Introduction

Sometimes observations from different independent sources are available for a classification task. Instead of training one large classifier on all data, it may be wise to combine just the outputs of the smaller classifiers trained on the individual data sources. Even when only one data source is available, a set of classifiers of different complexity and trained with different training algorithms can be designed and combined. It not only decreases the training time but also can increase the robustness and the performance of the classification [1].

A large number of combining schemes for classification exists. In general three types of situations in combining classifiers can be identified [2]. In the first type each classifier outputs a single class label and these labels have to be combined [3]. In the second type the classifiers output sets of class labels ranked in the order of likelihood [4] and the third type involves the combination of real valued outputs for each class by the respective classifiers (most often posterior probabilities [5], sometimes evidences [6]).

Commonly a combined decision is obtained by just averaging the estimated posterior probabilities. This simple algorithm already gives very good results [7,8]. This result is somewhat surprising, especially considering the fact that averaging of the posterior probabilities is not based on some solid (Bayesian) foundation. When the Bayes theorem is adopted for the combination of different classifiers, a product combination rule automatically appears under the assumption of independence: the outputs of the individual classifiers are multiplied and then normalized (this is also called a logarithmic opinion pool [9]).

In Refs. [10,11] a theoretical framework for combining classifiers is developed. For different types of combination rules, under which minimum and maximum rules, weighted averages, mean and product, derivations were obtained. When classifiers are used on identical data representations, the classifiers estimate the same class posterior probability. To suppress the errors in the estimates, the classifier outputs should be averaged. On the other hand, when independent data representations are available, classifier outcomes should be multiplied to gain maximally from the independent representations. In

* Corresponding author. Tel.: 00-31-(0)15-278-1845.
E-mail address: davidt@ph.tn.tndelft.nl (D.M.J. Tax).
[1] Now with TNO, Institute of Applied Physics, Delft, The Netherlands.

PII: S0031-3203(99)00138-7

Ref. [12] comparisons between the average combination rule and the product combination rule are made. It was confirmed that when independent data sets were available, the product combination rule should be used. Only in case of poor posteriori probability estimates, the more fault tolerant mean combination rule is to be used.

In this paper we summarize the differences and similarities between these two classifier combination rules. The combination rules will be derived from the posterior probabilities estimated by different classifiers and we will show that for two-class problems there is no difference between the two combination rules, independent of the fact if independent or dependent data representations are used. In multi-class problems the product combination rule is shown to be superior to the mean combination rule, only more noise sensitive. Also the use of rejection of objects which are classified with low confidence by the combination rules, will be shown. The problem which remains is when to use which combination rule. This choice depends on the fact if the problem is a two- or a multi-class problem, if independent representations are available and how well the classifiers estimate the posterior probabilities.

In Section 2 some ways to obtain posterior probabilities are discussed. The derivation of the two combination rules is shown in Section 3. In Section 4 differences between the mean combination and product combination rules are investigated in case of two- or multi-class problems, in case of very noisy posterior probabilities and in case of rejection of objects. Section 5 will show how this theory translates into practice. The last section will discuss the results.

2. Probabilities estimated by classifiers

Our goal in training classifiers is to find the classification rule that minimizes the probability of error. The optimal rule in classification is the Bayesian decision rule: assign a pattern to the class with the largest posterior probability. A drawback is that the true posterior probability has to be known. When this posterior probability is known (or is estimated) not only a classification can be obtained but also the confidence in this classification can be assessed. When two classes have almost the same posterior probability the classification can be rejected due to lack of confidence. This results in more reliable classifications.

Some classifiers immediately offer estimates of posterior probabilities, like the multilayer perceptron, trained with backpropagation (see Ref. [13]) or by maximizing the cross-entropy on the network outputs (see Ref. [14]). In other classification methods, probabilities are harder to obtain. Often class probabilities estimates are only reliable for large training sets, as for instance in the case of the k-nearest-neighbor classifier.

In this paper four different classifiers are used. The first is a Gaussian linear classifier, which assumes for each class a Gaussian probability distribution with equal covariance matrices. The posterior probability for one object to belong to class ω_j is

$$f_j(x) = \frac{p(x|\omega_j)P(\omega_j)}{\sum_k p(x|\omega_k)P(\omega_k)}, \tag{1}$$

where $p(x|\omega_j)$ is normally distributed.

The second linear classifier is Fisher linear classifier, where a pseudoinverse is used when the covariance matrix is close to singular (see also Ref. [15]). This classifier does not give direct estimations of the posterior class probabilities. Therefore we use the sigmoids of the distances to the decision boundary. These are optimally scaled using a logistic approach and are used as approximations of the posterior probabilities. In cases involving more classes, for each separate class a classifier is trained between that class and all the other classes combined. Using this method the posterior probabilities of the classes follow from the classifier for that class.

Third classification method is the quadratic classifier. It assumes like the Gaussian linear classifier Gaussian probability distributions for each of the classes, only the covariance matrices do not have to be the same. The same posteriori probabilities are used.

The last method used is the multilayer perceptron. Networks with different number of hidden units are trained with the Matlab Neural Network Toolbox. It is a gradient descent method with variable learning rate and stopping criterion based on the performance on an independent test set. The normalized network output is used as the estimate for the posteriori probability.

3. Combining rules

We assume that all objects are represented by feature vectors $x \in \chi$, each object belonging to one of C classes $\omega_j, j = 1, \ldots, C$. When R measurement vectors $x^1, \ldots, x^R$ from feature spaces $\chi^1, \ldots, \chi^R$ are available, the probability $P(\omega_j|x^1, \ldots, x^R)$ has to be approximated to make a classification (see also Ref. [10]).

In each of the R feature spaces a classifier can be constructed which approximates the true posterior class probability $P(\omega_j|x^k)$ in χ^k:

$$f_j^k(x^k) = P(\omega_j|x^k) + \varepsilon_j^k(x^k), \tag{2}$$

where $\varepsilon_j^k(x^k)$ is the error made by classifier k on the probability estimate that object x^k belongs to class ω_j. A combination rule combines these $f_j^k(x^k)$ to approximate $P(\omega_j|x^1, \ldots, x^R)$ as good as possible.

Two extreme cases can be distinguished, the first in which $\chi^1 = \chi^2 = \cdots = \chi^R$, the second where $\chi^1, \ldots, \chi^R$

are different and assumed to be independent. In the first case all the classifiers use the same data x: $P(x^1, \ldots, x^R|\omega_j) = P(x^1|\omega_j) \cdot \delta(x^1 - x^2) \cdot \ldots \delta(x^{R-1} - x^R)$. This trivially leads to

$$P(\omega_j|x^1, \ldots, x^R) = P(\omega_j|x^k) \quad \text{for any } k, 1 \leqslant k \leqslant R. \tag{3}$$

This $P(\omega_j|x^k)$ is estimated by $f_j^k(x^k)$. When we assume zero-mean error for $\varepsilon_j^k(x^k)$ (i.e. zero bias), all $f_j^k(x^k)$'s can be averaged to obtain a less error-sensitive estimation. This leads to the mean combination rule

$$f_j(x^1, \ldots, x^R) = \frac{1}{R} \sum_{k=1}^{R} f_j^k(x^k). \tag{4}$$

In the second case all feature spaces are different and class conditionally independent. The probabilities can be written as $P(x^1, \ldots, x^R|\omega_j) = P(x^1|\omega_j) \cdot P(x^2|\omega_j) \cdot \ldots P(x^R|\omega_j)$. Using the Bayes rule, we derive

$$P(\omega_j|x^1, \ldots, x^R) = \frac{\prod_k P(\omega_j|(x^k)/P(\omega_j)^{R-1}}{\sum_{j'}\{\prod_{k'} P(\omega_{j'}|x^{k'})/P(\omega_{j'})^{R-1}\}}. \tag{5}$$

In case of equal a priori class probabilities ($P(\omega_j) = 1/$(number of classes)), this formula reduces to a product combination rule (Eq. (6)) with $\varepsilon_j^k(x^k) = 0$:

$$f_j(x^1, \ldots, x^R) = \frac{\prod_{k=1}^{R} f_j^k(x^k)}{\sum_{j'} \prod_{k=1}^{R} f_{j'}^k(x^k)}. \tag{6}$$

4. Differences between averaging and multiplying

From the derivation of Eqs. (4) and (6), we would expect that the two rules will be useful under different conditions. The mean combination rule will be especially useful in case of identical or very highly correlated feature spaces in which the classifiers make independent errors. The product combination rule is apt for different, class conditionally independent feature spaces where classifiers make small estimation errors. We will investigate this correlation dependance with a very simple model.

In this model data consist of two classes in an N-dimensional space, each normally distributed (see for a three-dimensional example Fig. 1). The first class is centered on the origin, the second on $(1, 1, \ldots, 1)/\sqrt{N}$. The covariance matrix can be adjusted. It can be changed from identity, in which case the data is uncorrelated for each component (see left picture in Fig. 1), to complete correlation, in which case all data are perfectly correlated for each component (right picture in Fig. 1). On each single feature a classifier is trained which means that the number of classifiers $R = N$. Each classifier has to estimate posterior probabilities and a decision boundary. Combining the predictions of the classifiers in the two extremes, perfect correlation and complete independence of the data, will indicate where one combination rule can be preferred over the other.

4.1. Two-class problems

In Table 1 classification errors for the (three) individual classifiers and the combination rules are shown. The Gaussian classifiers are trained on 20 training patterns, tested on 100 patterns. The average individual test error rates of the classifiers are listed in the second column. The error rates from the combination rules are shown in the third and fourth column. The values are averages over 50 runs.

In these very simple experiments we see that no difference between the combination rules exists, even when a large number of classifiers is used. In case of two-class problems, we can derive conditions for which the rules behave the same. Assume we have equal class probabilities for the classes. When the product rule classifies some object x to class ω_j, (j can be 1 or 2) then

$$\prod_k f_j^k(x^k) > \prod_k (1 - f_j^k(x^k)), \tag{7}$$

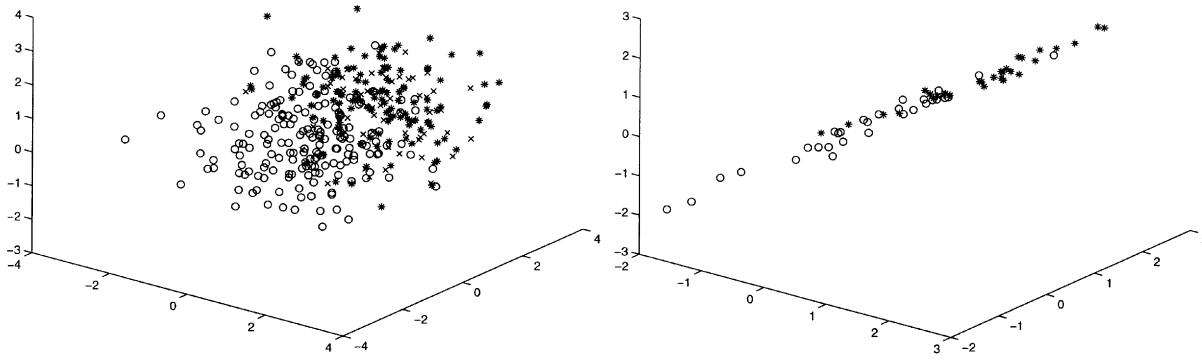

Fig. 1. Data distribution in three dimensions, (left) uncorrelated, (right) correlated.

Table 1
Results of combining three and ten classifiers with the two combining rules

Data correlation	Average test error indiv. classifiers (%)	Test error mean combination (%)	Test error product combination (%)	Average improvement (%)
$R = 3$				
0.0	0.40 ± 0.06	0.32 ± 0.04	0.32 ± 0.04	0.00 ± 0.01
0.5	0.37 ± 0.04	0.32 ± 0.03	0.32 ± 0.03	0.00 ± 0.01
1.0	0.32 ± 0.03	0.32 ± 0.03	0.32 ± 0.03	0.00 ± 0.00
$R = 10$				
0.0	0.46 ± 0.05	0.37 ± 0.04	0.37 ± 0.04	0.00 ± 0.01
0.5	0.43 ± 0.05	0.34 ± 0.03	0.34 ± 0.03	0.00 ± 0.01
1.0	0.32 ± 0.03	0.32 ± 0.03	0.32 ± 0.03	0.00 ± 0.00

We can rewrite $f_j^k(x^k) = \bar{f}_j(x) + \xi_j^k$ with $\bar{f}_j(x) = 1/R\sum_k f_j^k(x^k)$ and therefore $\sum_k \xi_j^k = 0$. This is basically the bias-variance decomposition (see Ref. [16]). The different values for ξ_j^k account for the variance, while $\sum_k \xi_j^k = 0$ indicates that there is no bias (the posterior probabilities are estimated well). We can expand the terms in Eq. (7):

$$\prod_k f_j^k = \bar{f}_j^R\left[1 + \sum_{k,k'} \frac{\xi_j^k \xi_j^{k'}}{\bar{f}_j^2} + \cdots\right]$$

$$= \bar{f}_j^R + \bar{f}_j^{R-2}\sum_{k,k'} \xi_j^k \xi_j^{k'} + \bar{f}_j^{R-3}\sum_{k,k',k''} \xi_j^k \xi_j^{k'} \xi_j^{k''} + \cdots. \quad (8)$$

For two-class problems $\xi_1^k = -\xi_2^k$. All sums over ξ_j^k's in the expansion of $\prod_k f_1^k$ and $\prod_k f_2^k$ will be equal, except for the sign in summations over an odd number of classifier outputs and for the factors $\bar{f}_j^{R-n}$. Using this in Eq. (7) results in

$$\bar{f}_j^R + \bar{f}_j^{R-2}\sum_{k,k'} \xi_j^k \xi_j^{k'} + \bar{f}_j^{R-3}\sum_{k,k',k''} \xi_j^k \xi_j^{k'} \xi_j^{k''} + \cdots$$

$$> (1-\bar{f}_j)^R + (1-\bar{f}_j)^{R-2}\sum_{k,k'} \xi_j^k \xi_j^{k'} - (1-\bar{f}_j)^{R-3}$$

$$\times \sum_{k,k',k''} \xi_j^k \xi_j^{k'} \xi_j^{k''} + \cdots. \quad (9)$$

When there are no outliers and ξ_j^k is smaller than $\bar{f}_j^k$, then $\sum_{k,k',k''} \xi_j^k \xi_j^{k'} \xi_j^{k''}$ will stay small. In Table 2 the size of $\bar{f}_j^{R-3}\sum_{k,k',k''} \xi_j^k \xi_j^{k'} \xi_j^{k''}$ relative to the previous two terms is shown for different number of classes. These values are the largest absolute values over all classes. The relatively large value of the second term does not influence the classification, because the signs of these terms are equal for all classes.

In case of a two-class problem the third term is very small and can be ignored. This means that when we start with the product combination rule (given by Eq. (7)) and we apply the approximation given by Eq. (8), we get the new combination rule: classify object x to class ω_j, (j can be 1 or 2) when

$$C\bar{f}_j^R > C(1-\bar{f}_j)^R. \quad (10)$$

This is a rescaled version of the mean combination rule for a two-class problem. In the product combination rule the output values are just shifted to the extremes; for $\bar{f}_j < 0.5$ to 0 and for values $\bar{f}_j > 0.5$ to 1.

Table 2
Sizes of higher-order sums in Eq. (9) relative to the first term, using $R = 3$ classifiers

No. Classes	Term 1	Term 2	Term 3
2	1.0	0.16	0.00016
3	1.0	0.22	0.00971
4	1.0	0.23	0.01862
5	1.0	0.30	0.02231

4.2. Multiclass problems

For multiclass problems the artificial two-class problem from the preceding section is extended. All classes still have the same covariance matrix, only their means are located at $n*(1,1,\ldots,1)/\sqrt{R}$, $n = 1,\ldots,C$. Training three Gaussian linear classifiers on this data sets for different number of classes, results in Fig. 2.

Using the combination rules for more than two-class problems, differences between the combination rules appear. Here the situation is much more complicated. Now not only the mean value of the estimated probabilities are most important, but also all individual class probabilities. For instance, when in the mean combination rule in a two-class problem, an object is assigned to a particular class, then the mean combination posterior probability is larger than 0.5. In a three-class problem, a mean value of 0.33 does not guarantee that the object will be assigned to

Fig. 2. Results of combining three classifiers for different number of classes. (The standard deviation for the individual classifiers is about 4%, for the combination rules about 2%.)

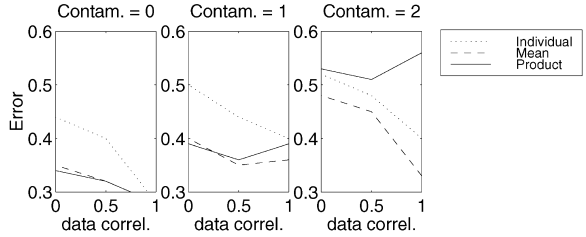

Fig. 3. Results of combining three classifiers for a three-class problem with one feature contaminated with noise, averaged over 10 runs. The left figure shows results without noise contamination, middle figure with noise between 0 and 1, right figure with noise between 0 and 2. The data correlation is shown on the x-axis. The dotted line is the average individual classifier error, the dashed line is the average rule, the solid line is the product rule. Standard deviation is about 5% for the classifiers, 2.5% for the improvement.

that class. Far away from the third class the decision boundary between two classes is still where the posterior probabilities of these two classes equal 0.5. For multi-class problems the decision boundaries can be in regions with posterior probabilities between $1/C$ and 0.5. This increase in uncertain areas for a growing number of classes also increases the chance that the classification will change when the product combination rule is used instead of the mean combination rule.

4.3. Error sensitivity

To investigate the error sensitivity of the combination rules, the training data used by one of the three classifiers, is contaminated by noise. The results are shown in Fig. 3. In the left picture, no noise is added. In the middle figure noise values equally distributed between zero and one are added to one feature, in the right picture noise between zero and two is added. It can be observed that when classifiers have poor decision boundaries and poor estimated probabilities the mean combination rule is more robust than the product rule. Especially when one of the classifiers is an outlier which outputs probabilities of 0 and 1, the product combination rule acts as a veto and the solution severely deteriorates.

The robustness of the mean combination rule with respect to the product combination rule is shown by Kittler [10] in which Eq. (2) is expanded, comparable with the expansion in Eq. (8). In that paper it is shown that the combined classifiers using a product-combination rule approximates the error free classifier up to a factor $[1+\sum_k(\varepsilon_j^k/P(\omega_j|x^k))]$ while in the mean-combination-rule the factor is $[1+(\sum_k\varepsilon_j^k/\sum_k P(\omega_j|x^k))]$. Note that $P(\omega)_j|x^k \leqslant 1$, so errors are amplified by the product rule. In the mean-combination rule the errors are divided by the sum of a posterior probabilities and especially for the winning class, where these probabilities are large, the errors are severely dampened.

4.4. Rejection

When a higher confidence is required for classification tasks, an acceptance threshold on the class probability can be introduced. Then probabilities around the decision boundary are excluded from classification. When several independent classifiers classify an object each with probability 0.6 to the same class, the sum rule also assumes that this probability is 0.6 while the product rule increases the confidence in this classification by a value depending on the number of other classes and the number of classifiers involved. From this observation it can be expected that when rejection thresholds are introduced, the mean-combination rule will reject objects which in the product combination rule are confidently classified.

In Fig. 4 the error is plotted versus the rejection rate for a two-class problem. Both, the mean-combination rule and the product rule give approximately the same curves. The amplification of the noise deteriorates all that has been gained by the increase of the confidence. In case of a problem with more than two classes (see Fig. 5) an improvement is obtained when the threshold is not set too high. For a rejection rate smaller than 80% the product combination rule outperforms the mean combination rule.

5. Experiments

5.1. Data

We tested the combination rules using real data sets. This data set is also used and explained by Van

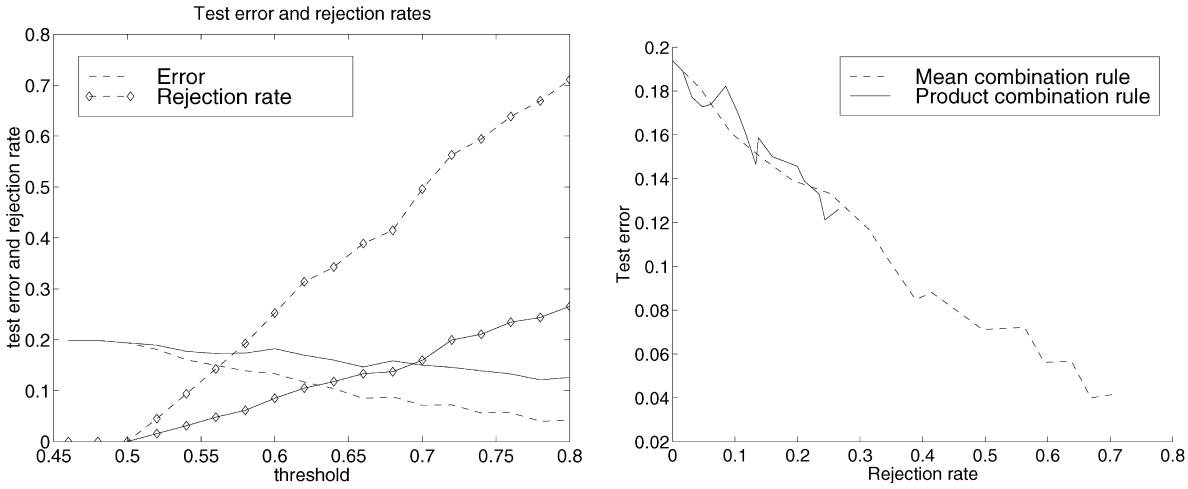

Fig. 4. (left) The error-rate and the rejection-rate versus the threshold on the confidence for the mean-combination (dashed) and the product-combination (solid) for two-class problems: (right) The error-rate versus the rejection rate for the mean-combination rule (dashed) and the product-combination-rule (solid).

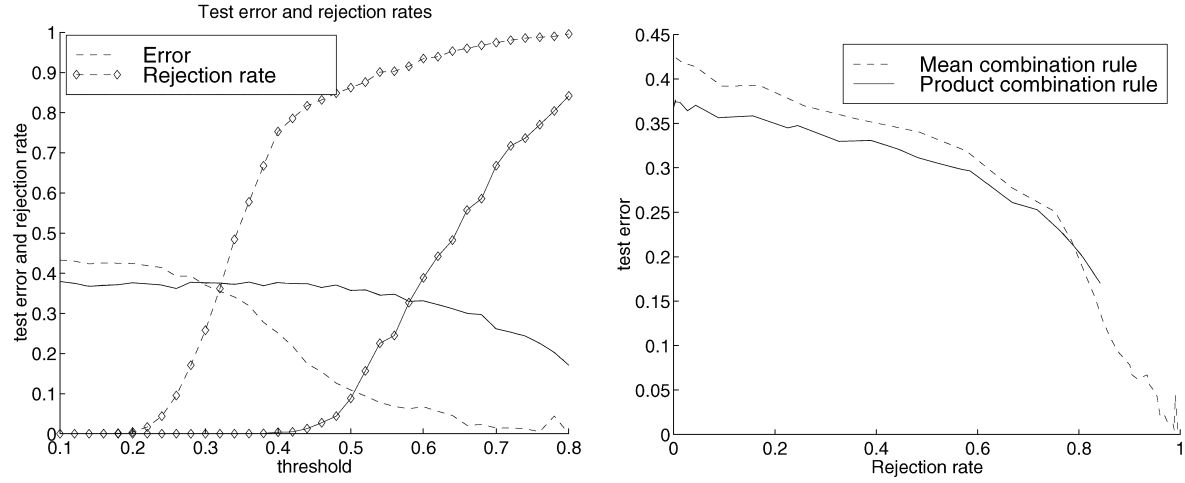

Fig. 5. The error-rate and the rejection-rate versus the threshold on the confidence for the mean-combination (dashed) and the product-combination (solid) for 10 classes: (right) The error-rate versus the rejection rate for the mean-combination rule (dashed) and the product-combination-rule (solid).

Breukelen [17]. From nine original maps from a Dutch public utility digits were extracted. The maps represent the position of a conduit system with respect to certain landmarks and were hand drawn by a large group of drawing engineers over a period of more than 25 yr. The data set is composed of separate dimensioning digits. The digits were automatically extracted from a binary image of the map, deskewed and normalized to fit exactly into a 30 by 48 pixel region. Finally, the digits were labelled manually.

From the set of 2000 digits four types of feature sets were extracted: Zernike moments, Karhunen–Loève features, Fourier descriptors and image vectors. Zernike moments are the projection of the image function onto a set of orthogonal basis functions. There are 13 orders of Zernike moments with 49 moments associated in total. For the feature extraction only the last 11 orders were used resulting in a subset of 47 Zernike moments. As Zernike moments are rotation invariant no distinction was made between digits with values 6 and 9. Thus, only nine classes were available for the Zernike features.

The Karhunen–Loève transform is a linear transform and corresponds to the projection of images onto the

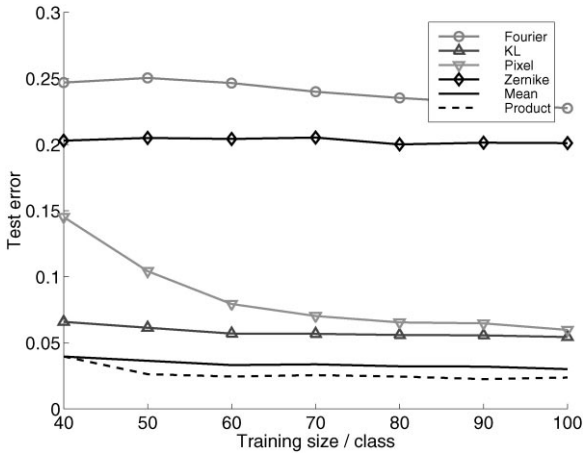

Fig. 6. Performance of the individual Gaussian linear classifiers and the combining rules.

eigenvectors of a covariance matrix. The covariance matrix is created from images of training data.

The fourth feature set is based on a simple transform of the binary image to a pixel vector. To reduce both the number of features and the possible loss of information the normalized image was divided into tiles of 2×3 pixels resulting in a total of 240 tiles. Each tile represents an element of the feature vector and the value corresponding to each tile was calculated simply by counting the number of object pixels within the tile.

Although the origin of the objects in all data sets is the same, applying different preprocessing methods results in independent measurements of these objects and thus in independent data sets. Therefore, we might expect that combining rules are able to improve upon individual classifiers considerably.

5.2. Results

We trained five linear Gaussian classifiers on each separate data set and combined the outputs. All classifiers are trained using different number of training patterns (at most 100 objects per class, in total 1000 objects for 10 classes) and are tested on a separate test set consisting of 4292 objects in total. The results are shown in Fig. 6. We see that in the Fourier feature set and in the Zernike feature set, classification errors are quite high. This is due to the rotation invariance of these features. The pixel set and Karhunen–Loève set both perform well. The posteriori probabilities seem to be estimated well, by using combination rules the classification performance dramatically improves. The product rule is consequently better than the mean combination rule. The best classification performance achieved is about 2.3% error.

In Fig. 7(a) the worst classifier, i.e. the classifier on the Fourier data set is removed. Again the product combination rule outperforms the other methods and achieves an error of 2.8%. Although the Fourier classifier had average classification error of about 25%, it still contributed to obtain a better classification.

In Fig. 7(b) the best classifier, the classifier trained on the Karhunen–Loève data set, is removed. Surprisingly by removing this data set the product combination rule still works very well. The mean combination rule deteriorates the performance of the classifier on the Karhunen–Loève data set. The best performance is now about 2.5% error.

These examples show that the classifiers on the independent data sets classify a large fraction of the objects correctly. This gives a stable behaviour when one classifier is removed. It also shows that the independent views of the different classifiers can contribute significantly to correct the output for some of the more difficult objects.

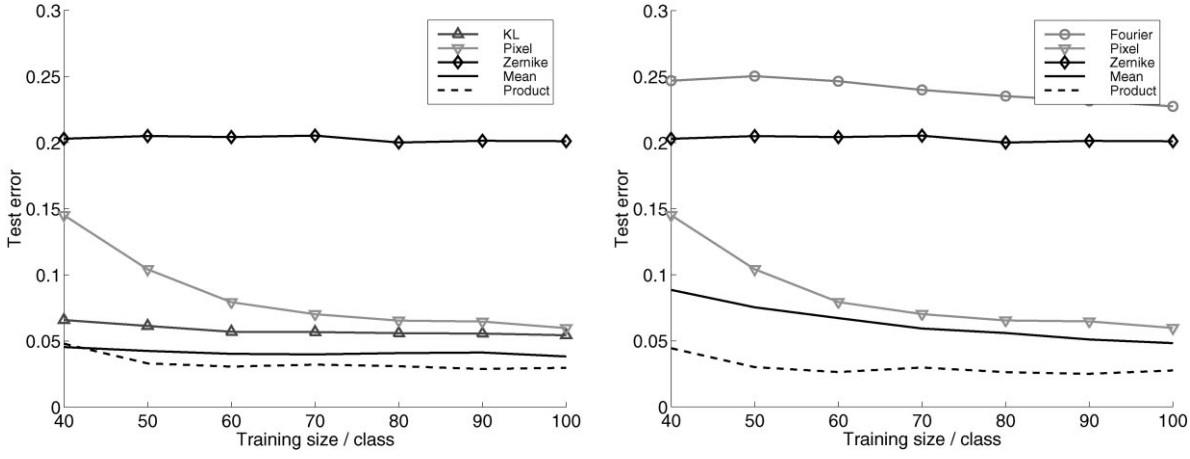

Fig. 7. Performance of the combining rules and the individual Gaussian linear classifiers on three data sets. In the left figure on data sets Karhunen–Loève, Pixel and Zernike, in the right figure on data sets Fourier, Pixel and Zernike.

Especially the product combination rule gains from these independent classifications. The mean combination rule on the other hand averages these contributions, which can lead to some performance loss. This is especially visible in Fig. 7(b). We can conclude that by using all available information, thus using all available data sets with a product combination rule, the optimal classification performance is achieved.

This experiment was repeated using another classifier, the Fisher linear discriminant (see Fig. 8). We see that here the estimations of the posteriori probabilities are worse than in the case of the Gaussian linear classifiers. The combination rules are still useful but the performance improvement is much lower. Especially in cases where one or two classifiers perform very badly, here with the training set sizes smaller than 40, outliers deteriorate the outcome of the combination rules. Both combination rules achieve about the same classification performance with an error of about 3.4%.

In Fig. 9 experiments with quadratic classifiers are shown. The quadratic classifier is regularized a bit (by adding 0.2 times the identity matrix to the covariance matrix) to make inversion of the covariance matrix possible. When we use a sufficient number of training patterns, more than 40 patterns per class, the individual performances on the Pixel and Karhunen–Loève data set improve with respect to the linear Gaussian classifiers, while performance on the Fourier data set deteriorates. Combining gives a serious improvement, which is an indication of improved probability estimates. Lowest classification error for both mean and product combination rule becomes 1.8%.

In Fig. 10 combinations of multi-layer perceptron classifiers is shown. For each data set a 8-hidden unit network is trained and the network outputs are combined.

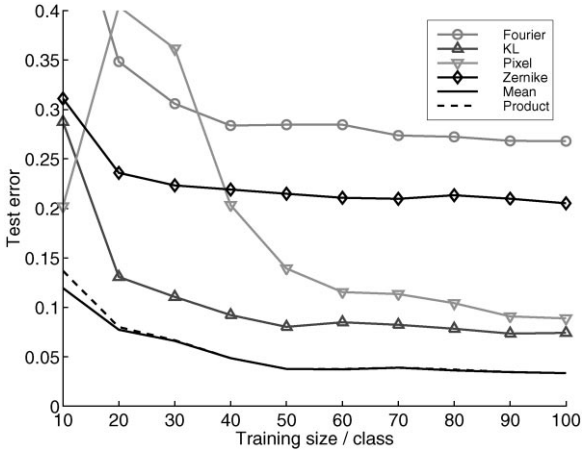

Fig. 8. Performance of the individual Fisher linear classifiers and the combining rules.

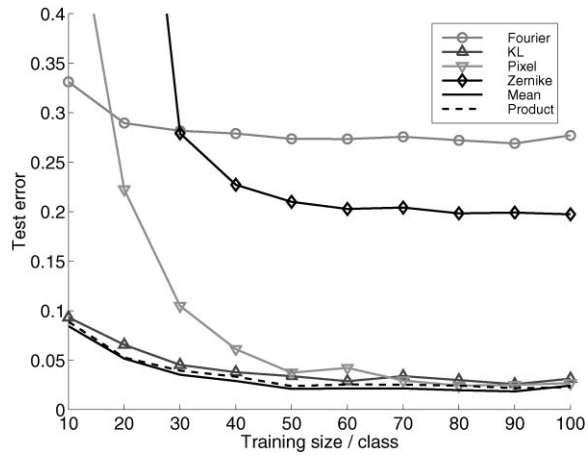

Fig. 9. Performance of the individual quadratic classifiers and the combining rules.

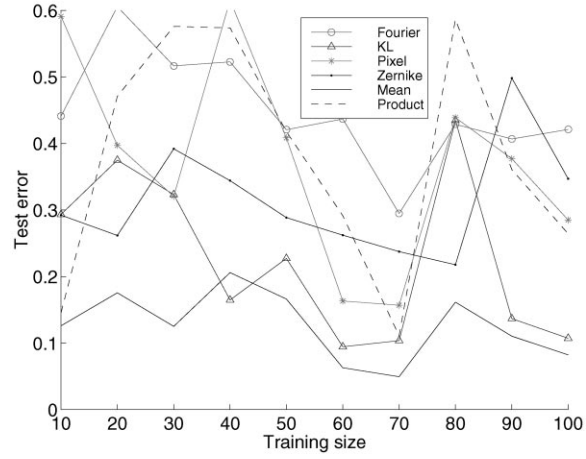

Fig. 10. Performance of the individual multi-layer perceptrons (eight hidden units) and the combining rules. Standard deviations on the MLP's are about 5%, only on the Pixel data this is about 20%.

This is done 5 times. The network outputs are very noisy, especially network on Pixel data can have extremely bad performance. Standard deviations on the graphs are 20% for the Pixel data and 5% for the other data sets. This results in a bad performance for the product combination rule. Only the more noise robust mean combination can improve upon the individual classifiers with a best performance of 5.0% error.

In Fig. 11 the performances on multi-layer perceptrons with 20 hidden units is shown (averaged over 10 runs). Here severe overtraining occurs and the product combination rule breaks down. Only the mean combination rule is robust enough to give reasonable results.

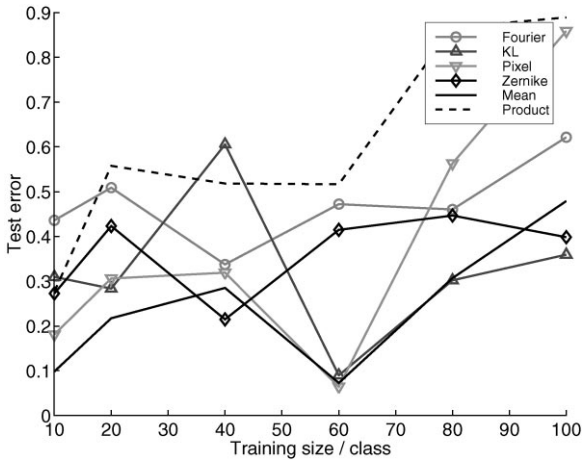

Fig. 11. Performance of the individual multi-layer perceptrons (20 hidden units) and the combining rules. Standard deviations are about 20%.

To check the results of Section 4.1 the combination rules are also applied to a two-class problem, the distinction between the classes 6 and 9. In Fig. 12 the performance of the Gaussian linear classifiers and the combination rules is shown for this two-class problem. In the left figure we see that the classifier on the Karhunen–Loève data set performs very well, with a minimum error of 1.0%. On the other hand, the performance on the Zernike data set is as bad as 50%. This is caused by the fact that the Zernike moments are rotational invariant and no differences between 6 and 9 can be found. Also the Pixel and Fourier data sets show bad classification performances. For these last three classifiers poor posteriori probabilities are expected and thus the expansion in Eq. (8) will not hold.

This is confirmed by the performances of the combination rules, both rules behave completely different. The mean combination rule gives reasonable results over the complete range of train sizes, but is far worse than the best classifier. The product combination rule encounters the problem that the posteriori probabilities are estimated very badly and for both classes a probability of zero is obtained. When for all classes a probability of zero is given, the final output probabilities cannot be normalized and the product combination rule does not give an outcome. Only for a large number of training objects per class, 90 and 100 objects per class, the product rule obtains classifications. This is shown in the right figure, which is an enlarged version of the lower part of the left figure.

Fig. 13 shows the results when Fisher linear discriminants are used for the same two-class problem, the 6 and the 9. Again the Zernike data set does not provide useful information but the classifiers on the Karhunen–Loève and the Pixel data set work well. For smaller training sets the probability estimates are not very accurate and the combination rules (especially the product combination rule) do not improve classification. For larger training set sizes both combination rules converge to a classification error of 0.5%, which is the same as the individual classifiers on the Pixel and Karhunen–Loève data sets. This confirms that with sufficient accuracy of the posteriori probability estimates in a two-class problem, both combination rules obtain the same classification.

6. Conclusions

The main goal of this paper was to investigate the relative merits of simple averaging over classifier outputs and multiplying the outputs. Although taking the aver-

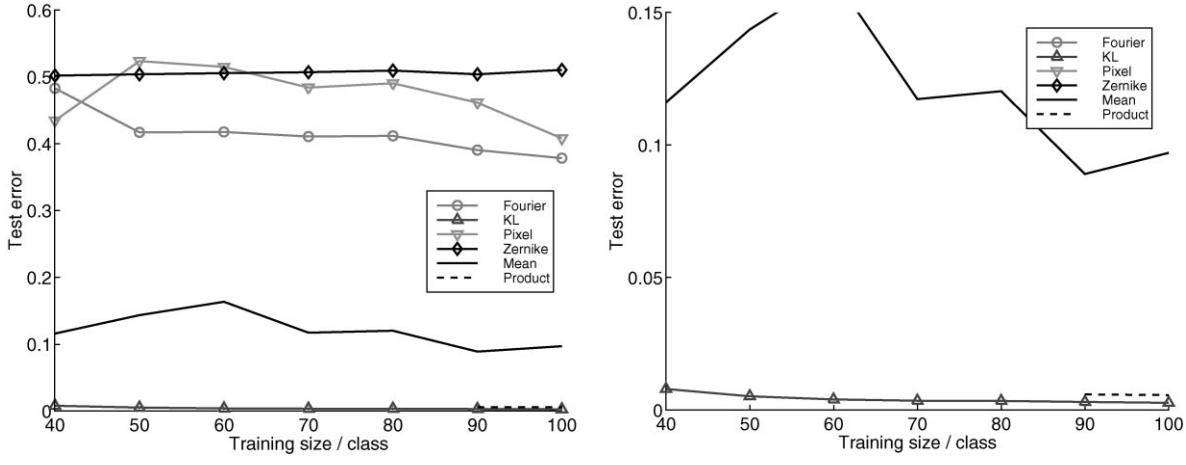

Fig. 12. Performance of the individual Gaussian linear classifiers and the combining rules on the classes 6 and 9.

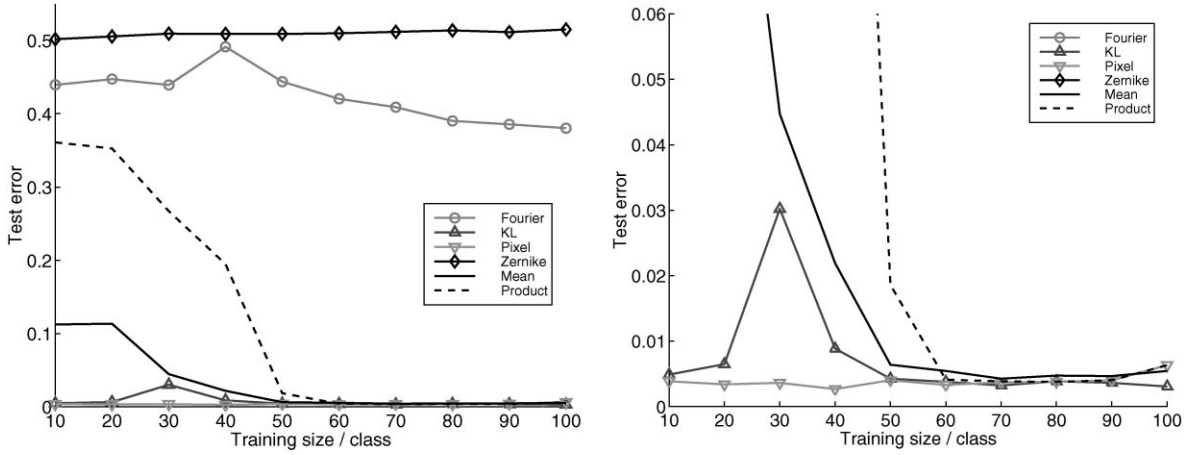

Fig. 13. Performance of the individual Fisher linear classifiers and the combining rules on the classes 6 and 9.

age is easy to perform and often results in good classification performance, this rule is not based on a solid Bayesian foundation. Under the assumption of independent feature spaces and using the Bayes rule would result in a product combination rule.

We showed that in case of a two-class problem in which posterior probabilities are well estimated (without a large number of extreme posterior probability estimations, one and zero), the mean combination rule and the product combination rule perform the same classification. Also when the rejection of objects with low classification confidence is allowed, mean and product combination rules do not differ significantly. Only in the case of larger estimation errors the product combination rule deteriorates with respect to the mean combination rule.

When the classification problem involves more than two classes, differences between the combination rules start to appear. Combining classifiers which are trained in independent feature spaces results in improved performance for the product rule, while in completely dependent feature spaces the performance is the same. When the rejection option is allowed, this holds for moderate rejection rates.

We can conclude that averaging-estimated posterior probabilities is to be preferred in the case when posterior probabilities are not well estimated. Only in the case of problems involving multiple classes with good estimates of posterior class probabilities the product combination rule outperforms the mean combination rule.

Acknowledgements

This work was partly 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).

References

[1] A. Sharkey, N. Sharkey, How to improve the reliability of artificial neural networks, Technical Report CS-95-11, Department of Computer Science, University of Sheffield, 1995.

[2] L. Xu, A. Kryzak, C.V. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Systems, Man Cybernet. 22 (3) (1992) 418–435.

[3] R. Battiti, A.M. Colla, Democracy in neural nets: voting schemes for classification, Neural Networks 7 (4) (1994) 691–707.

[4] K. Tumer, J. Ghosh, Order statistics combiners for neural classifiers, in: Proceedings of the World Congress on Neural Networks, INNS Press, Washington DC, 1995, pp. I:31–34.

[5] R. Jacobs, Method for combining experts' probability assessments, Neural Comput. 7 (5) (1995) 867–888.

[6] G. Rogova, Combining the results of several neural network classifiers, Neural Networks 7 (5) (1994) 777–781.

[7] S. Hashem, Optimal linear combinations of neural networks, Neural Networks (1994).

[8] M. Tanigushi, V. Tresp, Averaging regularized estimators, Neural Comput. 9 (1997) 1163–1178.

[9] J.A. Benediktsson, P.H. Swain, Consensus theoretic classification methods, IEEE Trans. Systems, Man and Cybernet. 22 (4) (1992) 688–704.

[10] J. Kittler, M. Hatef, R.P.W. Duin, Combining classifiers, Proceedings of ICPR'96, (1996) 897–901.

[11] J. Kittler, A. Hojjatoleslami, T. Windeatt, Weighting factors in multiple expert fusion, in: A.F. Clark, (Ed.), Proceedings of the eighth British Machine Vision Conference, University of Essex Printing Service, 1997, pp. 41–50.

[12] D.M.J. Tax, R.P.W. Duin, M. van Breukelen, Comparison between product and mean classifier combination rules, in: P. Pudil, J. Novovicova, J. Grim, (Eds.), First International Workshop on Statistical Techniques in Pattern Recognition, Institute of Information Theory and Automation, June 1997, pp. 165–170.

[13] D.W. Ruck, S.K. Rogers, M. Kabrisky, M.E. Oxley, B.W. Suter, The multilayer perceptron as an approximation to a Bayes optimal discrimination function, IEEE Trans. Neural Networks 1 (4) (1990) 296–298.
[14] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Walton Street, Oxford OX2 6DP, 1995.
[15] S. Raudys, R.P.W. Duin, Expected classification error of the fisher linear classifier with pseudo-inverse covariance matrix, Pattern Recognition Lett. 19 (5-6) (1998) 385–392.
[16] S. Geman, E. Bienenstock, R. Doursat, Neural networks and the bias/variance dilemma, Neural Comput. 4 (1992) 1–58.
[17] M. van Breukelen, R.P.W. Duin, D.M.J. Tax, Combining classifiers for the recognition of handwritten digits, in: P. Pudil, J. Novovicova, J. Grim, (Eds.), First International Workshop on Statistical Techniques in Pattern Recognition, Institute of Information Theory and Automation, June 1997, pp. 13–18.

About the Author—DAVID M.J. TAX received the M.Sc. degree in Physics from the University of Nijmegen, The Netherlands in 1996. Currently he works in the Pattern Recognition group in the Delft University of Technology. His research interest include pattern recognition with a focus on neural networks and support vector machines.

About the Author—MARTIJN VAN BREUKELEN received the M.Sc. degree in Applied Physics from the Delft University of Technology in the Netherlands in 1998. Currently he works at the Institute of Applied Physics of the Netherlands Organization for Applied Scientific Research (TNO). His professional interests include design and development of decision support systems and machine intelligence.

About the Author—ROBERT P.W. DUIN studied Applied Physics at Delft University of Technology in the Netherlands. In 1978 he received the Ph.D. degree for a thesis on the accuracy of statistical pattern recognizers. In his research he included various aspects of the automatic interpretation of measurements, learning systems and classifiers. Between 1980 and 1990 he developed and studied hardware architectures and software configurations for interactive image analysis. At present he is an associate professor of the Faculty of Applied Sciences of Delft University of Technology. His main research interest is in the design and evaluation of learning algorithms for pattern recognition applications. This includes in particular neural network classifiers, support vector classifiers and classifier combining strategies.

About the Author—J. KITTER graduated from the University of Cambridge in Electrical Engineering in 1971 where he also obtained his Ph.D. in Pattern Recognition in 1974 and the Sc.D. degree in 1991. He joined the Department of Electronic and Electrical Engineering of Surrey University in 1986 where he was a Professor, in charge of the Centre for Vision, Speech and Signal Processing. He has worked on various theoretical aspects of Pattern Recognition and on many applications including automatic inspection, ECG diagnosis, remote sensing, robotics, speech recognition, and document processing. His current research interests include Pattern Recognition, Image Processing and Computer Vision. He has co-authored a book with the title "Pattern Recognition: a statistical approach" published by Prentice-Hall. He has published more than 300 papers. He is a member of the Editorial Boards of Pattern Recognition Journal, Image and Vision Computing, Pattern Recognition Letters, Pattern Recognition and Artificial Intelligence, and Machine Vision and Applications.