KYBERNETIKA — VOLUME *34* (1998), NUMBER 4, PAGES 381 - 386

# HANDWRITTEN DIGIT RECOGNITION BY COMBINED CLASSIFIERS

M. VAN BREUKELEN, R. P. W. DUIN, D. M. J. TAX[1] AND J.E. DEN HARTOG

Classifiers can be combined to reduce classification errors. We did experiments on a data set consisting of different sets of features of handwritten digits. Different types of classifiers were trained on these feature sets. The performances of these classifiers and combination rules were tested. The best results were acquired with the mean, median and product combination rules. The product was best for combining linear classifiers, the median for $k$-NN classifiers. Training a classifier on all features did not result in less errors.

## 1. INTRODUCTION

In practical pattern recognition problems one often tries a number of classifiers and a number of feature sets in order to find the best combination. As soon as this combination is found the other classifiers and features are no longer used. Methods for combining classifiers to reduce the number of classification errors are described in recent literature. In this paper the usefulness of combining classifiers was tested on a real data set consisting of several sets of features of handwritten digits. The questions we would like to answer are: "When does combining classifiers result in a reduction of classification errors and why?" and "If we have a large set of features, how do we divide this set into subsets in order to get the best results?" Section 2 of this paper describes how classifiers can be combined, Section 3 how our classifiers estimate posterior probabilities and Section 4 describes our data. In Section 5 we describe the experiments we did and in Section 6 our conclusions.

## 2. COMBINING CLASSIFIERS

Classifiers can be combined by their outputs. The output of a classifier can be a label, ranking of labels or a continuous output such as an estimate for the posterior probability. As mentioned above, the goal of the combination is a reduction of classification errors. In literature several rules for combining classifiers have been described. In our experiments the mean, max(imum), min(imum), median, majority vote and product rule are used, see also [1, 6, 7]. The majority vote rule combines

[1] This work was partly supported by the Foundation for Applied Sciences (STW) and the Dutch Organization for Scientific Research (NWO). We also thank the PNEM for supplying the data.

the classifiers by the labels produced by the classifiers. The other rules combine the classifiers by the posterior probabilities estimated for the different classes. The maximum rule can be considered as a 'strongest support', the mean as a 'strongest average support' and the minimum as a 'least objection' rule. The product rule is more sensitive to 'objection' than to 'support' since variations in an output close to zero have more influence on the product than variations close to one. If the feature sets are independent the product rule is expected to perform best [6].

Another way of combining classifiers is training an 'output classifier' on the outputs of the classifiers. This way the outputs of the classifiers are treated as new features. In the experiments in this paper each combined classifier is of the same type and has its own set of features. Combining classifiers instead of combining features directly can be useful if adding features does not improve the performance of a classifier, for example due to dimensionality problems.

## 3. ESTIMATION OF POSTERIOR PROBABILITIES BY OUR CLASSIFIERS

Different types of classifiers result in different estimates for the posterior probabilities. This section discusses the estimation of posterior probabilities by classifiers. All the used classifiers originate from the Matlab toolbox PRTOOLS [2]. We used the following classifiers: Gaussian linear, Fisher linear, Karhunen–Loève linear and the $k$-NN rule. The Gaussian linear classifier estimates the posterior probabilities for the classes assuming Gaussian density distributions for the features.

Our Fisher linear classifier is based on a pseudo inverse if the covariance matrix is close to singular. To handle a multiple class problem linear discriminant functions are calculated between each class and the total of all other classes. The normalised sigmoids of the distances to the discriminant functions are taken as the estimates for the posterior probabilities. The distances are scaled according to the maximum likelihood principle, implemented using the logistic model. This model was chosen for its wide coverage of probability distributions.

The KL linear classifier uses a Karhunen–Loève expansion to project the features onto the first $n$ eigenvectors of the total learning set. The value of $n$ is such that the variance remains 90 % of the original variance. In this decorrelated subspace a Gaussian classifier is calculated and transformed back to the original space. Posterior probabilities are estimated by this Gaussian classifier.

The $k$-NN classifier calculates the squared distance from the sample to be classified to the $k$th nearest sample of each class. The reciprocals of these divided by their sum are used to estimate the posterior probabilities.

## 4. THE DATA SET

We did experiments on a data set consisting of different feature sets. The acquisition and the feature extraction of the data set is described in this section.

The digits used in our experiments were extracted from nine original maps from a Dutch public utility. The maps were scanned in 8 bits grey value at a density of 400 dpi, scanned, sharpened [6] and thresholded. The digits were then automatically

extracted [4] and deskewed using the orientation of the arrow. They were normalised to fit into a 30 by 48 pixel region to prevent scaling artifacts and labeled manually.

From a set of 2000 digits with 200 samples for each of the ten classes four types of feature sets have been extracted: Zernike moments, Karhunen–Loève features, Fourier descriptors and image vectors. The Zernike set consists of 47 rotation invariant Zernike moments [5] and 6 morphological features like the number of endpoints of the skeleton. The Karhunen–Loève Transform is a linear transform that corresponds to the projection of images onto the eigenvectors of a covariance matrix, see also [3]. This set consists of 64 features. The Fourier set consists of 76 two-dimensional shape descriptors. The features of the 'pixel' feature set were obtained by dividing the image of 30x48 pixels into 240 tiles of 2x3 pixels and counting the number of object pixels in each tile. In Figure 1 these 'Pixel' features are visualised for some samples. The Zernike and Fourier feature sets are rotation invariant and can't distinguish the samples of class '6' from the ones of class '9' and vice versa. More details about the feature sets can be found in [1].

**Fig. 1.** Visualization of some samples of the 'pixel' feature set.

## 5. EXPERIMENTS

In the first experiments a Gaussian classifier was trained on each feature set. These classifiers were combined using combination rules. The performances of the classifiers and combination rules were tested on the learning and remaining testing samples. The classification errors against the number of learning samples per class are plotted in Figure 2. On the left the results of the individual classifiers are plotted and on the right those of the combination rules.

The Gaussian classifiers on the Fourier and Zernike set make the most errors, about 20 % and 15 % respectively on the test set. This is due to the rotation invariance of these feature sets. The classifiers on the Pixel and the Karhunen–Loève set perform equally well for a learning set of 100 samples per class. Their test errors at this point are about 5 %. All combination rules, except for majority voting, reduced the error. The product rule performed best, its error was somewhat less than 2 % for 100 learning samples per class.

After these promising results we tried the same with other classifiers instead of the Gaussian classifier. The same sets of learning samples were used each time preserving 100 samples/class for testing. Table 1 shows the average errors over 8 experiments. The first four rows show the results of the classifiers on the feature sets. The fifth row shows the results on the Zernike features after normalisation on their variances. The results of the combination rules are shown in the next six rows. The combination rules used the first three plus either the fourth or the fifth classifier of the same column. The last row shows results of a trained output classifier, in this case the Fisher linear classifier.

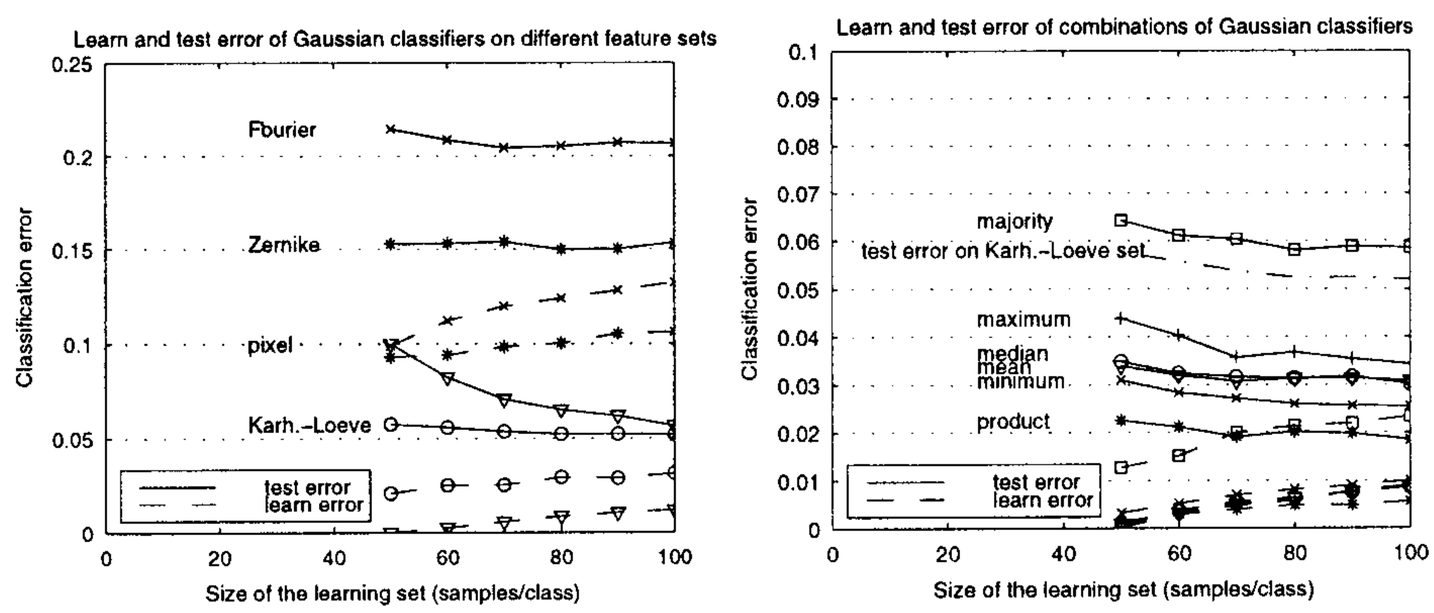


**Fig. 2.** The average errors on the learning and testing set against the number of learning samples/class for Gaussian linear classifiers (left) and combinations (right).

**Table 1.** Classification errors in % for different classifiers on the feature sets and combinations of these classifiers. Each row represents a classifier or combination, each column the type of classifier that was used.

| | 1-ncc | 1-ncc | 2-ncc | 2-ncc | 5-ncc | 5-ncc | K-L | K-L | Gaussian | Fisher |
|---|---|---|---|---|---|---|---|---|---|---|
| Fourier | 18.0 | 18.0 | 17.1 | 17.1 | 17.2 | 17.2 | 20.3 | 20.3 | 20.7 | 22.4 |
| K-L | 3.4 | 3.4 | 3.2 | 3.2 | 4.3 | 4.3 | 5.1 | 5.1 | 5.2 | 7.8 |
| pixel | 3.1 | 3.1 | 2.8 | 2.8 | 3.7 | 3.7 | 4.2 | 4.2 | 5.7 | 8.8 |
| Zernike | 23.4 | | 23.7 | | 27.0 | | 90.0 | | 15.4 | 15.9 |
| scld Zern. | | 18.2 | | 17.0 | | 17.1 | | 18.7 | | |
| maximum | 18.3 | 11.5 | 20.2 | 8.7 | 25.5 | 8.1 | 4.0 | 3.9 | 3.4 | 5.5 |
| minimum | 8.2 | 7.3 | 3.7 | 2.5 | 4.8 | 3.0 | 3.6 | 2.7 | 2.6 | 5.7 |
| mean | 8.3 | 6.6 | 3.8 | 1.7 | 4.9 | 2.1 | 3.7 | 3.3 | 3.1 | 4.6 |
| majority | 6.0 | 5.8 | 5.2 | 5.1 | 5.4 | 5.5 | 4.4 | 6.0 | 5.9 | 7.5 |
| product | 7.5 | 6.6 | 3.0 | 1.7 | 4.4 | 2.2 | 3.0 | 2.0 | 1.9 | 4.4 |
| median | 2.7 | 2.5 | 1.8 | 1.8 | 2.2 | 2.4 | 3.7 | 3.3 | 3.0 | 4.6 |
| Fisher | 8.3 | 6.6 | 4.8 | 4.7 | 3.3 | 3.5 | 3.6 | 3.5 | 5.1 | 7.1 |

A combination is useful if it results in a smaller error than that of the best individual classifier used for it. The ten best results of Table 1 were all results of combination rules, namely of the median, mean and the product rule. These mean results vary from 1.7–2.4%, their standard deviations are in the range of 0.09–0.15%. The median rule was useful in all cases, but not always the best. It works better for the $k$-NN than for the linear classifiers. The product rule was the best combination for all combinations of linear classifiers. The mean and the product were more sensitive to worse performing classifiers, as can be concluded from the difference in results with the scaled and the unscaled Zernike features. Due to the fact that the $k$-NN classifier is a local classifier, the discriminant functions

are much more irregular. This can result in more outliers in the estimates of the posterior probabilities. This and the sensitivity of the product rule could explain why it performs better for linear classifiers. It also explains why the results for the mean and median are almost the same for the combinations of linear classifiers. The minimum and the maximum rule are sensitive too. The maximum rule performed bad, the minimum rule was only good for the Gaussian and the KL classifiers. Majority voting is not sensitive but was only for the Fisher classifiers useful. The trained output was useful occasionally but wasn't as good as the best combination rules. We think that better results may be obtained but more experiments will have to be done.

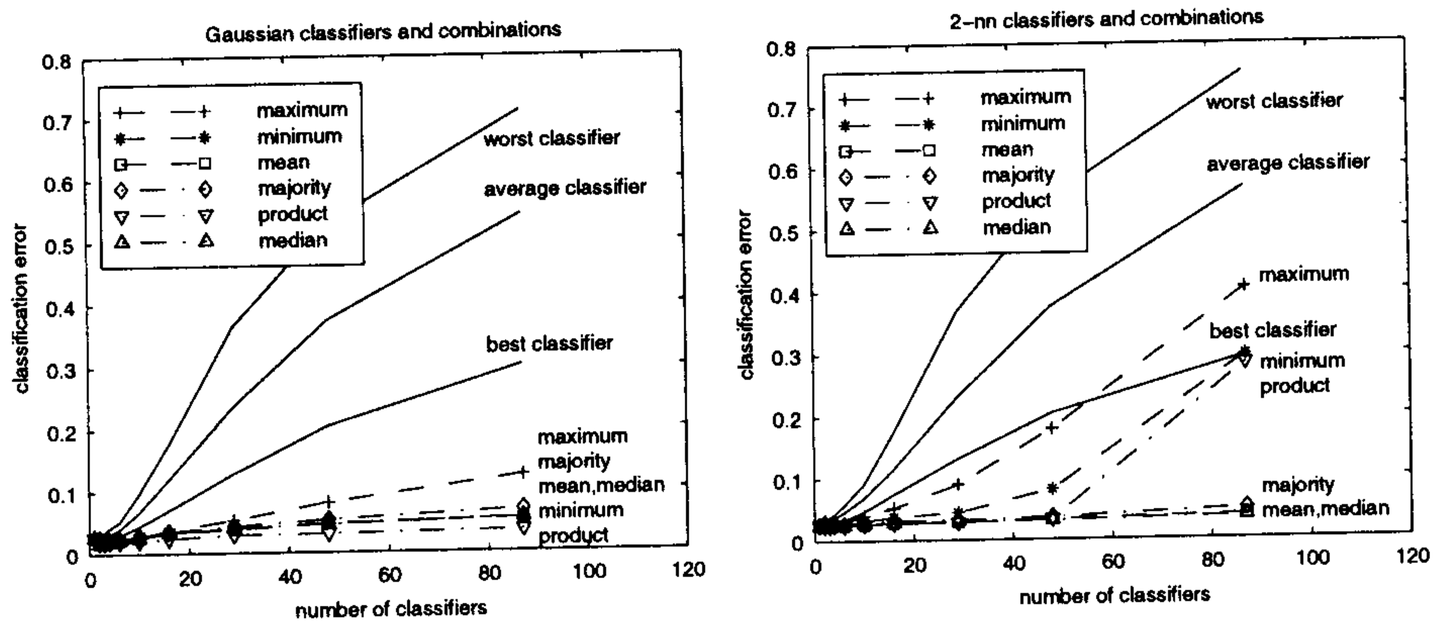


**Fig. 3.** The results of distributing the features over small numbers of classifiers for the Gaussian linear (left) and the 2-NN classifier (right).

In final experiments we studied combining all features in a single feature set. We were interested whether there would be an optimal number of classifiers if the features were to be divided in subsets. For this purpose we first normalised all features and combined them in a single set. These features were then randomly distributed over a number of classifiers such that each classifier used the same number of features. The classifiers were again combined. We varied the number of classifiers 1 to 87. So in the one extreme a single classifier used all 433 features and in the other side 86 classifiers used 5 features and one classifier used 3 features. The results are shown in Figure 3 for the Gaussian and 2-NN classifier. For the Gaussian classifier some peaking is visible. The result of a single classifier is worse (2.8%) than the average result of two (2.6%) classifiers. The results of combinations of a few classifiers were somewhat worse than that of combinations of classifiers on the original feature sets. Since these results are averages over only three experiments this difference is not significant. The best result was an error of about 2.6% and was acquired by combining all features in a single set. Combining two or three classifiers with the product, mean and median did result in small, but not significant, improvements. The best result is worse than the, about 1.7%, error of the mean,

median and product rule on the original feature sets with only the Zernike features normalised. Combining many Gaussian classifiers resulted in small errors. This is because for the Gaussian classifier overlapping classes led to posterior probabilities close to priori probabilities. Consequently, bad classifiers have less influence on results of combination rules than good ones. The product rule is the best for all numbers of Gaussian classifiers. For the 2-NN rule, bad classifiers still have a large influence due to its local character. This is why all rules perform very bad for the combination of 87 classifiers.

## 6. CONCLUSIONS

The best results in our experiments were acquired using the mean, median and product combination rules. The median is a robust rule and worked especially good on the $k$-NN classifiers. The mean rule was never much better than the median rule and is less robust. The product rule was especially good for combining linear classifiers. Combining all features in a single feature set never resulted in fewer classification errors. We did only experiments with the Fisher classifier as output classifier. The results of the used output classifier were not very good, but we think better results can be obtained. More research will be needed to verify this.

(Received December 18, 1997.)

## REFERENCES

[1] M. van Breukelen, R.P.W. Duin, D.M.J. Tax and J.E. den Hartog: Combining classifiers for the recognition of handwritten digits. In: Proceedings of the 1st IAPR TC1 Workshop on Statistical Techniques in Pattern Recognition, Prague 1997, pp. 13–18.

[2] R.P.W. Duin: PRTOOLS, A Matlab Toolbox for Pattern Recognition. 1995.

[3] M.D. Garris et al: NIST Form–Based Handprint Recognition System. Internal Report National Institute of Standards and Technology NISTIR 5469, 1994.

[4] J.E. den Hartog, T.K. ten Kate, and J.J. Gerbrands: Knowledge–based interpretation of utility maps. In: Computer Vision Graphics and Image Processing: Image Understanding, 1996, pp. 105–117.

[5] A. Khotanzad and Y.H. Hong: Rotation invariant pattern recognition using Zernike moments, In: Int. Conf. on Pattern Recognition, Rome 1998, pp. 326–328.

[6] J. Kittler, M. Hatef and R.P.W. Duin: Combining classifiers. In: Proceedings of ICPR'96, pp. 897–901.

[7] K. Tumer and J. Ghosh: Theoretical Foundations of Linear and Order Statistics Combiners for Neural Pattern Classifiers. TR-95-02-98, The Computer and Vision Research Center, The University of Texas at Austin, 1995.

*M. van Breukelen, R. P. W. Duin and D. M. J. Tax, Faculty of Applied Sciences, Delft University of Technology, P.O.Box 5046, 2600 GA Delft. The Netherlands.*
*e-mails: martijnb, bob, davidt @ph.tn.tudelft.nl*

*J. E. den Hartog, TNO Institute of Applied Physics, Delft. The Netherlands.*
*e-mail: hartog@tpd.tno.nl*