The Combining Classifier: to Train or Not to Train?

Robert P.W. Duin
Pattern Recognition Group, Faculty of Applied Sciences
Delft University of Technology, The Netherlands
duin@ph.tn.tudelft.nl

Abstract

When more than a single classifier has been trained for the same recognition problem the question arises how this set of classifiers may be combined into a final decision rule. Several fixed combining rules are used that depend on the output values of the base classifiers only. They are almost always suboptimal.

Usually, however, training sets are available. They may be used to calibrate the base classifier outputs, as well as to build a trained combining classifier using these outputs as inputs. It depends on various circumstances whether this is useful, in particular whether the training set is used for the base classifiers as well and whether they are overtrained.

We present an intuitive discussing on the use of trained combiners, relating the question of the choice of the combining classifier to a similar choice in the area of dissimilarity based pattern recognition. Some simple examples will be used to illustrate the discussion.

1. Introduction

For almost any real world pattern recognition problem a series of approaches and procedures may be used to solve it. Various representations may be possible, like graphs, dissimilarities and features. In each of them the use of several object measurements may be considered. Once the representation has been established, many decision procedures are available, most of them having again a wide choice of possible training or estimation procedures.

At some stage in the design of a pattern recognition system one thereby has collected a set of possible classifiers, that may be based on entirely different object representations. Traditionally the best classification system is selected on the basis of an evaluation. Recently, the possibilities of combining sets of classifiers has been considered. There are many examples found in which such a combination of classifiers has a better performance than any of the base classifiers in the set [3,9,11]. How to construct such a combination of classifiers has become an important direction of research [14,15].

The difficulties that arise in combining a set of classifiers is directly evident if one considers the metaphor of a committee of experts. How has such a committee to arrive at a final decision? By voting? But that neglects their differences in skills and seems pointless if the constitution of the committee is not carefully set up. This may be solved by assigning areas of expertise and following the best expert for each new item of discussion. In addition to the decision, the experts may be asked to provide some confidence. But what to do with an expert that claims to have a great insight with respect to aspects of the problems that is not shared by anyone else? As a consequence he is dominating the decision at points that seem arbitrary for the others. Who will decide whether he is faking or really an expert?

The above problem can not be detected if we just establish the committee, give them a decision procedure using their own confidences and follow their decisions. For designing an optimal decision procedure we need to evaluate the committee. Such an evaluation would mean that we supply problems with known solution, study the expert advises and construct from that procedure the combined decision rule.

In terms of classifiers this is called training. It will be clear that training is needed unless the collection of experts fulfills certain conditions. We will start by summarizing in section 2 the ways a set of base classifiers can be found or generated. In section 3 conditions will be discussed for some fixed rules that can be applied. Some typical situations in which these conditions fail are discussed as well.

Several ways to train the combining classifier are discussed in section 4. Consequences for the use of the training set are treated in section 5. The problem of finding a combining classifier has strong similarities with building classifiers from dissimilarity based representations [22,23]. This is explained further in section 6.

The paper is concluded by a final discussion.

2. The base classifiers

The base classifiers may arise from the application or may deliberately be generated in order to construct an advanced classifier that performs better than any of the base

1051-4651/02 $17.00 (c) 2002 IEEE

classifiers. We will enumerate the various possibilities of the second type. When in solving applications a series of classifiers is encountered that may be combined then it is usually within one of these types.

Base classifiers should be different (as it makes no sense to combine identical classifiers), but they should also be comparable, i.e. their outputs should be represented such that a combining classifier can use the as inputs. We will return on the second demand later. A consistent set of different classifiers may be generated in the following ways:

1. *Different initializations*. If the training is initialization dependent, different initializations may result in different classifiers. This holds, for instance, for neural networks.
2. *Different parameter choices* like the number of neighbors in the k-NN rule, the size of the smoothing parameter in the Parzen classifier, the kernel in the support vector classifier, the amount of pruning in a decision tree, the size of a regularization parameter and the choice of the target value for neural networks.
3. *Different architectures* like the size of a neural network.
4. *Different classifiers* trained in the same feature space and by the same training set, like Fisher's Linear Discriminant, Bayes normal, a support vector classifier and a decision tree.
5. *Different training sets*, i.e. different samples from the same design set, with or without replacement. Well known examples are bagging [2] and boosting [6]. In the first, training sets are fully independent by bootstrapping, in the second, they differ systematically as a result of previous classification. Training sets may also be different if each of the classes is first split by a cluster analysis and then the classes are separated cluster by cluster. Another example of this group is the set of two-class discriminants that may be used to solve an m-class problem by discriminants between each of them and the m-1 others [31].
6. *Different feature sets*. In some applications objects may be represented in entirely different feature domains, e.g. in identification by speech and by images, or by a set of answers on a medical checklist to be combined with the results of medical tests. The generation of random subsets out of a large feature set appears also to be successful [27,28]

This list is roughly ordered such that lower in the list the combination of classifiers becomes more successful due to the fact that classifiers are increasingly different and still informative [5,32]. How different the resulting classifiers are and especially how this should be measured is an open, but heavily studied topic [20].

It should be realized that training sets for various classifiers may have different sizes and that these sets in particular may differ from a possible training set for the combining classifier as well as from an evaluation set. The objects in these sets need to have representations that can be applied to all constituting classifiers, otherwise the combination becomes impossible. Consequently, they may be significantly smaller than sets used for training the base classifiers.

The size of the set of base classifiers determines the input dimensionality for the combiner (the number of classes times the number of base classifiers). We will return later on how important this size is, in relation with the size of the training set for judging whether a trained combiner should be used.

3. Fixed combining rules

The fixed combining rules make use of the fact that the outputs of the base classifiers are not just numbers, but that they have a clear interpretation: class labels, distances, or confidences. The confidence is sometimes interpreted or generated by fuzzy class membership functions[1,17] sometimes by class posterior probabilities [4]. In the following discussion we will use the latter concept. The confidence $P_i(\mathbf{x})$ of object $\mathbf{x}$ with respect to class ω_i ($i = 1, ..., c$) is defined as

$$P_i(\mathbf{x}) = \mathrm{Prob}(\omega_i \mid \mathbf{x}) \tag{1}$$

In relation with classifier $\mathbf{C}_j(\mathbf{x})$, however, it depends just on the outcome $C_{ij}(\mathbf{x})$ of this classifier for class ω_i:

$$P_{ij}(\mathbf{x}) = \mathrm{Prob}(\omega_i \mid C_{ij}(\mathbf{x})) \tag{2}$$

$C_{ij}(\mathbf{x})$ is some numerical outcome of classifier j for class ω_i. It can be the distance to a prototype, the distance to a separating hyperplane, the outcome of an output unit of a neural network, etcetera. The probabilities in (1) and (2) are defined over all $\mathbf{x}$ in the universe of objects of interest. Classification is done by assigning object $\mathbf{x}$ to the class with the highest confidence. The probability that $\mathbf{x}$ is correctly classified by classifier j, called the local accuracy $\eta_j(\mathbf{x})$, is

$$\eta_j(\mathbf{x}) = \max_i\{P_i(\mathbf{x})\} \tag{3}$$

The expected accuracy η_j of classifier $\mathbf{C}_j(\mathbf{x})$ thereby is

$$\eta_j = E_{\mathbf{x}}[\max_i\{P_{ij}(\mathbf{x})\}] \tag{4}$$

If we assume that the classifier outputs $C_{ij}(\mathbf{x})$ are estimates of the confidences $P_{ij}(\mathbf{x})$ then $\eta_j(\mathbf{x})$ is estimated by

$$\hat{\eta}_j(\mathbf{x}) = \max_i\{C_{ij}(\mathbf{x})\} \tag{5}$$

The expected accuracy can be found by an evaluation set:

$$\hat{\eta}_j = \sum_k \max_i\{C_{ij}(\mathbf{x}_k)\} \tag{6}$$

It is possible to use for this estimate the training set. For a well trained classifier holds that $\hat{\eta}_j \approx \eta_j$, while for an overtrained classifier will hold that $\hat{\eta}_j$ » η_j.

Some well known simple fixed rules for combining the set of base classifiers $\{\mathbf{C}_j(\mathbf{x}), j=1, ..., n\}$, into a combining classifier $\mathbf{Q}(\mathbf{x}) = \{Q_i(\mathbf{x}), i=1, ..., c\}$ will now be summarized (see [19] for a theoretical comparison of some of them). Note that normalization may be needed.

1. *the product rule*:

$$Q_i(\mathbf{x}) \sim \prod_j C_{ij}(\mathbf{x}) \qquad (7)$$

This rule is good if the individual classifiers are independent, i.e. that the outcomes of $C_{ij}(\mathbf{x})$ for random **x** are independent for fixed *i* (class) and variable *j* (classifier). This is hardly ever the case. An example may be found by two classifiers computed for different feature spaces that are entirely unrelated, e.g. based on face images and voices assuming that within a class the feature distributions in the two spaces are independent. See [13, 30]. The rule assumes noise free and reliable confidence estimates. It fails if these estimates may be accidentally zero or very small.

2. *the sum rule*:

$$Q_i(\mathbf{x}) \sim \sum_j C_{ij}(\mathbf{x}) \qquad (8)$$

This is equivalent to the product rule for small deviations in the classifier outcomes (still assuming independent classifiers).

An entirely different set-up for which this rule may work well is a collection of similar base classifiers with independent noise behavior. In this case the errors in the confidences are averaged out by the summation. An example is a set of classifiers based on the same model (e.g. Bayes normal) in the same feature space, but trained by independently drawn training sets. This is the case in bagging [2,26]. Also in case a large set of similar classifiers is generated based on different, randomly selected feature sets, the sum-rule may be useful in reducing the noise in large sets of so-called weak classifiers [10,27,28].

3. *the maximum rule*:

$$Q_i(\mathbf{x}) \sim \max_j\{C_{ij}(\mathbf{x})\} \qquad (9)$$

At first glance this seems reasonable: select the classifier that is most confident of itself. This immediately fails, however, if some classifiers are more overtrained than others. In that case they may be overconfident and thereby dominating the outcome, without having a better performance. This can be corrected by a following training procedure.

The maximum rule, however, also fails for simple classifiers that are not sensitive for nuances that more complicated, and thereby better, classifiers are able to detect. The first ones dominate the maximum rule, deteriorating the classification accuracy. It appears to be hard to find examples in which the global maximum rule (9) is intuitively a good choice. Our main application is the combination of a set of two-class discriminants for solving multi-class problems [31].

4. *the minimum rule*:

$$Q_i(\mathbf{x}) \sim \min_j\{C_{ij}(\mathbf{x})\} \qquad (10)$$

This is not as strange as it seems, as it finally will select the outcome of the classifier that has the least objection against a certain class. But, like for the maximum rule, a good example of a situation in which this rule is really adequate is hard to find.

5. *the median rule*:

$$Q_i(\mathbf{x}) \sim \mathrm{median}_j\{C_{ij}(\mathbf{x})\} \qquad (11)$$

This rule is similar to the sum rule (8) but may yield more robust results.

There are several rules for crisp classifier outputs, based on the generated class labels only, like the (weighted) majority vote and the naive Bayes combiner [18]. We will not discuss them in this paper.

There are many examples in which fixed rules appear to be useful, i.e. that the combined classifier is better than each of the base classifier individually. In particular this holds if the feature spaces are different, but also combinations of classifiers trained in the same feature space by the same training set may show improved results [5,12].

From the above summary it can be concluded that only under very strict conditions a fixed rule is really the best combination. They will certainly be sub-optimal if the base classifiers generate unreliable confidences (e.g. caused by a small training set or by overtraining). But also if the available set of objects is sufficiently large to avoid this an improved result may be found by carefully training the combining classifier. This will be discussed further in the next section.

4. Trained combiners

Instead of using one of the fixed combining rules a training set can be used to adapt the combining classifier to the classification problem. A few possibilities will be discussed. In the following section we will go into the issue of the choice and the size of the training set.

4.1 Calibration of base classifier outputs

The base classifiers may be trained independently. There are several reasons why their outputs are not automatically optimally scaled with respect to each other: they deal with different feature spaces, they are based on different models, they needed different numbers of training epochs, etcetera. It is important that their outputs are properly scaled, especially if they are used for confidence estimates and a fixed combiner is used. A simple normalization of weights, even if all base classifiers are linear and in the same feature space, is not sufficient. It is important for combining that the confidence estimates are such that a classifier outcome of $y = C_{ij}(\mathbf{x})$ implies that a fraction y of all objects x with the same classifier outcome belong to class ω_i. So, if $C_{ij}(\mathbf{x})$ is based on some discriminant $S_{ij}(\mathbf{x})$ then

$$C_{ij}(\mathbf{x}) = f(S_{ij}(\mathbf{x})) \tag{12}$$

with f(•) such that

$$C_{ij}(\mathbf{x}) \approx \mathrm{Prob}(\mathbf{x} \in \omega_i | S_{ij}(\mathbf{x})) \tag{13}$$

In general, f(•) has to map distances to a discriminant to probabilities, so $f: \Re \rightarrow [0,1]$. An example is the logistic function $f(z) = 1/(1+\exp(-\alpha z))$. The free parameter α has to be optimized on the training set such that (13) holds.

4.2 Global selection and weighting of base classifiers

The base classifiers may differ in performance as well as in the amount of overtraining. Both can be measured by an evaluation set. The results may be used for the selection or weighting of classifiers like in the weighted majority rule traditionally used in boosting [6,28]. Confidence estimates of overtrained classifiers can be improved by rescaling their outputs as discussed in section 4.1.

4.3 Global selection of ensembles

Instead of selecting base classifiers, also a selection may be made out of a set combining systems, each of them being based on possibly another subset of base classifiers and/or another combining rule. To make such a selection by an evaluation set is in fact a kind of training. This type of selection differs from the traditional selection of the best single classifier as the number of possible combinations is much larger. For a discussion see [29].

4.4 Local selection of base classifiers

Base classifiers may differ in performance over the set of objects. For different objects different classifiers may perform well. If their local confidence estimates $C_{ij}(\mathbf{x})$ are sufficiently reliable they might be used directly, resulting in the maximum rule as discussed in section 3. Unless the classifier is based on density estimates in the entire feature space, this method is not local. A linear discriminant, for instance, produces the same outcomes on all points in the feature space that have the same distance to the discriminating hyperplane. The corresponding confidence estimate may be good for some objects and bad for others. Here a training set may be used to estimate locally the confidence (performance) for each classifier.

The are several schemes proposed for the local selection of classifiers, e.g. see [7,8,18,33]. The basic idea, however, is similar: use the training set to partition the feature space in regions and find for each region the best base classifier. The combining classifier has first to find the region of the object **x** to be classified. Next, the classification is done by the base classifier assigned to that region.

Selection of base classifiers appears to work surprisingly well, even if just a small set of objects is used to define the regions. The large comparison experiment reported in [20] shows the best results for a combiner based on a selection procedure originally proposed in [33].

4.5 The general combining classifier

The outputs of the base classifiers can be used as the input features of a general classifier used for combining, e.g. the Parzen classifier, a neural network or Fisher's linear discriminant. An example of such a classifier used for combining is the Decision Template proposed in [16]. It is in fact the nearest mean method applied on the confidence outputs of the base classifiers.

If a Bayes consistent classifier is used like the Parzen classifier, then for large training sets the classifier is optimal and thereby the combiner is. It may be expected, however, that other, more specialized classifiers are possible that perform better (i.e. approach the asymptotic Bayes performance faster). In particular it may be expected that classifiers like some fuzzy sets based decision rules [1,17] can show this behavior as they make use of the fact that we are dealing with confidences and not with arbitrary numerical features.

5. Some remarks on training sets

In the design of a combined classifier system the problem of how to use the total available set of objects, the design set, is more complicated than in the design of a single classifier, see also [25,29]. On several places there is the need to train and to evaluate classifiers and sets of classifiers. On the basis of these evaluations again decisions have to be made, e.g. with respect to the set of base classifiers and to the selection of the combining rule.

The re-use of the training set used for the design of the base classifiers on the combination level has to be discouraged if the base classifiers are (almost) overtrained. Corrections are sometimes possible [24], but may be better avoided. As the combination of weak classifiers can be very successful, the combination of weakly trained base classifiers may be a good option as it allows the re-use of the same training data.

We see the following possible strategies (in all cases a part of the design set has to be reserved exclusively for the final evaluation):

1. Use just a single training set. Train the base classifiers carefully, such that overtraining is really avoided and

confidence estimates are reliable. The use of fixed combining rules may be effective now.

2. Use just a single training set. Train the base classifiers weakly. The same training set may now be used for training a combining classifier.
3. Separate the available training sets into two parts. Use one part for the base classifiers and one part for the combining classifier. The base classifiers can be trained as good as possible, without more precaution against overtraining than usual. If some overtraining occurs, this is corrected by training the output classifier on an independent training set.

A fourth possible strategy, often used in practice, is discouraged by us. That is the use of just a single training set combined with training of base classifiers without more precaution against overtraining than usual. In this case, both, the use of fixed as well as trained combiners is not well possible anymore. Fixed combiners will not work because the confidence estimations are due to overtraining. The training of a combining classifier will fail if performed by the same training set as its representation in the output space of the base classifiers is not representative for new objects.

In the selection of a strategy the size of the available training set is important. A large set of base classifiers will result into a high dimensional feature space for the combining classifier. This requires a large training set.

6. Dissimilarity based pattern recognition

The use of untrained classifiers like the fixed combiners is not uncommon in pattern recognition. To some respect the nearest neighbor rule is also untrained. In this case the available training set is just used as a reference, but there is no classifier optimised on it. New objects are directly classified according to the most similar training object. In fact this rule is comparable to the maximum combiner.

It appears to be possible to construct a trained classifier on the distance matrix that represents the training set. In the nearest neighbor rule this distance matrix is not used. Classification is done on the distances to new objects only. In dissimilarity based pattern recognition [22,23] classifiers are trained on the distance matrix of the training set. Instead of the nearest neighbor distance some linear or nonlinear combination of all distances is optimized, similar to the trained combiner as being a function of confidences.

It is argued here that trained combiners are asymptotically better than fixed combiners. This may also be true for dissimilarity based classifiers. The initial experiments show that this holds often for objects directly represented by dissimiliraities. The relation between dissimilarity based pattern recognition and combining classifiers may be deepened further, as the distance to a labeled training object is in fact very similar to the outcome of a classifier.

7. Discussion

We presented a discussion on the use of trained combiners. This is by far not a new issue. The book by Nilsson [21] treated this topic already in 1965. Confidences, however, were hardly used at that time.

Confidences make the issue of combining classifiers manageable as it allows for continuous feature spaces. Still much more has to be done. In the fixed combining rules confidences are treated according to their interpretation, but these rules are sub-optimal. In the trained combining rules that interpretation is usually neglected. They may be asymptotically optimal, but might do better if a better use could be made of the properties of confidences.

We emphasized that a proper training of base classifiers is important, certainly if one likes to get most out of the data. In relation with classifier combining, proper training implies avoiding overtraining entirely as the performance of the base classifiers is not of primary importance, Instead, reliable outcomes transformable to unbiased confidence estimates is the main issue.

A next step in research may be the retraining of base classifiers (or even the redesign of the whole set) after training and evaluating the combining classifier. By this, the design of a combined classifier system becomes an iterative procedure. For the construction of advanced and complicated pattern recognition systems this may be finally unavoidable.

Acknowledgement

The author gratefully acknowledges his colleagues Pavel Paclik, Elżbieta Pękalska, Dick de Ridder, Marina Skurichina, David Tax and Piotr Jusczcak for stimulating discussions and support.

8. References

[1] J.C. Bezdek, S.K. Pal, *Fuzzy models for Pattern Recognition*, IEEE Press, Piscataway, 1992.

[2] L. Breiman, Bagging predictors, *Machine Learning*, vol. 24, pp. 123-140,1996.

[3] K. Chen, L. Wang and H.S. Chi, Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification, *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 11, no. 3, 417-445.

[4] R.P.W. Duin and D.M.J. Tax, Classifier conditional posterior probabilities, in: A. Amin, D. Dori, P. Pudil, H. Freeman (eds.), *Advances in Pattern Recognition*, Lecture Notes in Computer Science, vol. 1451, Springer, Berlin, 1998, 611-619.

[5] R.P.W. Duin and D.M.J. Tax, Experiments with Classifier Combining Rules, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems* (Proc. First International Workshop, MCS 2000, Cagliari, Italy, June 2000), Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 2000, 16-29.

[6] Y. Freund and R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences*, vol. 55, no. 1, 1999, 119-139.

[7] G. Giacinto and F. Roli, Dynamic classifier selection based on multiple classifier behaviour, *Pattern Recognition*, vol. 34, no. 9, 2001, 1879-1881.

[8] G. Giacinto and F. Roli, An approach to the automatic design of multiple classifier systems, *Pattern Recognition Letters*, vol. 22, no. 1, 2001, 25-33.

[9] T.K. Ho, J.J. Hull, and S.N. Srihari, Decision combination in multiple classifier systems, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, 1994, 66-75.

[10] T.K. Ho, The random subspace method for constructing decision forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 8, 1998, 832-844.

[11] Y.S. Huang and C.Y. Suen, Method of combining multiple experts for the recognition of unconstrained handwritten numerals, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, 1995, 90-94.

[12] A.K. Jain, R.P.W. Duin, and J. Mao, Statistical Pattern Recognition: A Review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, 2000, 4-37.

[13] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, 1998, 226-239.

[14] J. Kittler, F. Roli, *Multiple Classifier Systems* (Proc. First Int. Workshop MCS 2000, Cagliari, Italy), Lecture Notes in Computer Science, vol. 1857, Springer Verlag, Berlin, 2000.

[15] J. Kittler, F. Roli, *Multiple Classifier Systems*, (Proc. Second Int. Workshop MCS 2001, Cambridge, UK), Lecture Notes in Computer Science, vol. 2096, Springer Verlag, Berlin, 2001.

[16] L.I. Kuncheva, J.C. Bezdek, and R.P.W. Duin, Decision Templates for Multiple Classifier Fusion: An Experimental Comparison, *Pattern Recognition*, vol. 34, no. 2, 2001, 299-314.

[17] L.I. Kuncheva, *Fuzzy classifier design*, Studies in Fuzziness and Soft Computing, Springer Verlag, Berlin, 2000.

[18] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: An experiment, *IEEE Transactions On Systems Man And Cybernetics, Part B-cybernetics*, vol. 32, no. 2, 2002, 146-156.

[19] L.I. Kuncheva, A theoretical study on six classifier fusion strategies, *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 24, no. 2, 2002, 281-286.

[20] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles (submitted).

[21] N.J. Nilsson, *Learning machines*, McGraw-Hill, New York, 1965.

[22] E. Pękalska and R.P.W. Duin, Automatic pattern recognition by similarity representations - a novel approach, *Electronic Letters*, vol. 37, no. 3, 2001, 159-160.

[23] E. Pękalska and R.P.W. Duin, Dissimilarity representations allow for building good classifiers, *Pattern Recognition Letters*, vol. 23, no. 8, 2002, 943-956.

[24] S. Raudys and A. Janeliunas, Reduction a Boasting Bias of Linear Experts, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems* (Proc. Third International Workshop, MCS 2002, Cagliari, Italy), Lecture Notes in Computer Science, Springer, Berlin, 2002.

[25] F. Roli, S. Raudys, G.L. Marcialis, An experimental comparison of fixed and trained fusion rules for crisp classifiers outputs, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems* (Proc. Third International Workshop, MCS 2002, Cagliari, Italy), Lecture Notes in Computer Science, Springer, Berlin, 2002.

[26] M. Skurichina and R.P.W. Duin, Bagging for linear classifiers, *Pattern Recognition*, vol. 31, no. 7, 1998, 909-930.

[27] M. Skurichina, *Stabilizing weak classifiers*, Ph.D. thesis Delft University of Technology, Delft, 2001, October 15, 1-208.

[28] M. Skurichina and R.P.W. Duin, Bagging, Boosting and the Random Subspace Method for Linear Classifiers, *Pattern Analysis and Applications*, 2002, in press.

[29] A.J.C. Sharkey, N.E. Sharkey, U. Gerecke, G.O. Chandroth, The "Test and Select" Approach to Ensemble Combination, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems* (Proc. First International Workshop, MCS 2000, Cagliari, Italy), Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 2000, 30-44.

[30] D.M.J. Tax, M. van Breukelen, R.P.W. Duin, and J. Kittler, Combining multiple classifiers by averaging or by multiplying?, *Pattern Recognition*, vol. 33, no. 9, 2000, 1475-1485.

[31] D.J.M. Tax and R.P.W. Duin, Using two-class classifiers for multi-class classification, *Proc. ICPR2002*, Quebec City, Canada, August 2002.

[32] W. Wang, P. Jones, D. Partridge, Diversity between Neural Networks and Decision Trees for Building Multiple Classifier Systems, in: J. Kittler, F. Roli (eds.), *Multiple Classifier Systems* (Proc. First International Workshop, MCS 2000, Cagliari, Italy), Lecture Notes in Computer Science, vol. 1857, Springer, Berlin, 2000, 240-249.

[33] K. Woods, W.P. Kegelmeyer, and K. Bowyer, Combination of multiple classifiers using local accuracy estimates, *IEEE Trans. on Pattern Anal. and Machine Intelligence*, vol. 19, no. 4, 1997, 405-410.