Combining One-Class Classifiers

David M.J. Tax and Robert P.W. Duin

Pattern Recognition Group, Delft University of Technology
{davidt,bob}@ph.tn.tudelft.nl

Abstract. In the problem of one-class classification target objects should be distinguished from outlier objects. In this problem it is assumed that only information of the target class is available while nothing is known about the outlier class. Like standard two-class classifiers, one-class classifiers hardly ever fit the data distribution perfectly. Using only the best classifier and discarding the classifiers with poorer performance might waste valuable information. To improve performance the results of different classifiers (which may differ in complexity or training algorithm) can be combined. This can not only increase the performance but it can also increase the robustness of the classification. Because for one-class classifiers only information of one of the classes is present, combining one-class classifiers is more difficult. In this paper we investigate if and how one-class classifiers can be combined best in a handwritten digit recognition problem.

1 Introduction

The goal of the Data Description (or One-Class Classification [10]) is to distinguish between a set of target objects and all other possible objects (per definition considered outlier objects). It is mainly used to detect new objects that resemble a known set of objects. When a new object does not resemble the data, it is likely to be an outlier or a novelty. When it is accepted by the data description, it can be used with higher confidence in a subsequent classification.

Different methods have been developed to make a data description. In most cases the probability density of the target set is modeled [12]. This requires a large number of samples to overcome the curse of dimensionality [4]. Other techniques than estimating a probability density estimate exist. It is possible to use the distance ρ to model or just to estimate the boundary around the class without estimating a probability density. A neural network can be restricted to form a closed decision surface [10], various forms of vector quantization [3] are possible and recently a method based on the Support Vector Classifier, the Support Vector Data Description [13] was proposed.

As in the normal classification problems, one classifier hardly ever captures all characteristics of the data. Combining classifiers can therefore be considered. Commonly a combined decision is obtained by just averaging the estimated posterior probabilities. This simple algorithm already gives very good results [11]. This is somewhat surprising, especially considering the fact that averaging of

J. Kittler and F. Roli (Eds.): MCS 2001, LNCS 2096, pp. 299–308, 2001.

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 [1]).

One-class classifiers cannot provide posterior probabilities for target objects, because information on the outlier data is not available. When a uniform distribution over the feature space is assumed, posterior probability can be estimated when the target class probability is found. When a one-class classifier does not estimate a density, its output should be mapped to a probability before it can be combined with other classifiers. In this paper we investigate the influence of the feature sets (are they dependent or not) and the type of one-class classifiers for the best choice of the combination rule.

2 Theory

We assume that we have data objects $\mathbf{x}_i, i = 1...N$, which are represented in several feature spaces $\chi_k, k = 1...R$. Each object can be a target object, labeled ω_T, or an outlier object ω_O (although during the training of one-class classifiers we assume example outlier objects are not available). In each feature space different one-class classifiers are trained. In [7] and in [8] a theoretical framework for combining (estimated posterior probabilities from) normal classifiers is developed. For different types of combination rules derivations are obtained. When classifiers are applied on (almost) identical data representations $\chi_1 = \chi_2 = ... = \chi_R$, the classifiers estimate the same class posterior probability $p(\omega_j|\mathbf{x}^k)$, potentially suffering from the same noise in the data. To suppress the errors in these estimates and the overfitting by the individual classifiers, the classifier outputs may be averaged. This results in the mean combination rule:

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

where j indexes the target and outlier class.

On the other hand, when independent data representations χ_i are available, classifier outcomes should be multiplied to gain maximally from the independent representations. This results in the product combination rule:

$$f_j(\mathbf{x}^1, ..., \mathbf{x}^R) = \frac{\prod_{k=1}^{R} f_j^k(\mathbf{x}^k)}{\sum_{j'} \prod_{k=1}^{R} f_{j'}^k(\mathbf{x}^k)} \tag{2}$$

2.1 One-Class Classifiers

One-class classifiers are trained to accept data from the target class and to reject outlier data. We can distinguish two types of one-class classifiers. The first type are the density estimators, which just estimate the target class probability

density $p(\mathbf{x}|\omega_T)$. In this paper we use a normal density, a mixture of Gaussians and the Parzen density estimation.

The second type of methods fit a model to the data and compute the distance $\rho_T(\mathbf{x})$ to this model. Here we will use four simple models, the support vector data description [14], k-means clustering, k-center method [15] and an auto-encoder neural network [6]. Here a descriptive model is fitted to the data and the resemblance (or distance) to this model is used. In the SVDD a hypersphere is put around the data. By applying the kernel trick (analogous to the support vector classifier) the model becomes more flexible to follow the characteristics in the data. Instead of the target density the distance to the center of the hyper sphere is used. In the k-means and k-center method the data is clustered, and the distance to the nearest prototype is used. Finally in the auto-encoder network the network is trained to represent the input pattern at the output layer. The network contains one bottleneck layer to force it to learn a (nonlinear) subspace through the data. The reconstruction error of the object in the output layer is used as distance to the model.

2.2 Posterior Probabilities for One-Class Classifiers

To make an accept/reject decision in all the one-class methods, a threshold should be set on the estimated probability or distance. A principled way for setting this threshold is to supply the fraction of the target set f_T which should be accepted. This defines the threshold:

$$\theta_{f_T} : \int I(p(\mathbf{x}|\omega_T) \geq \theta_{f_T}) d\mathbf{x} = f_T \tag{3}$$

where $I()$ is the indicator function. In this paper it is assumed that for all methods the threshold is put such that f_T of the target data is accepted ($f_T = 0.9$).

When one-class classifiers are to be combined based on posterior probabilities, Bayes rule should be used to compute $p(\omega_T|\mathbf{x})$ from $p(\mathbf{x}|\omega_T)$:

$$p(\omega_T|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_T)p(\omega_T)}{p(\mathbf{x})} = \frac{p(\mathbf{x}|\omega_T)p(\omega_T)}{p(\mathbf{x}|\omega_T)p(\omega_T) + p(\mathbf{x}|\omega_O)p(\omega_O)} \tag{4}$$

Because the outlier distribution $p(\mathbf{x}|\omega_O)$ is unknown, and even the prior probabilities $p(\omega_T)$ and $p(\omega_O)$ are very hard to estimate, equation (4) cannot be used directly. The problem is solved when an outlier distribution is assumed. When $p(\mathbf{x}|\omega_O)$ is independent of $\mathbf{x}$, i.e. it is an uniform distribution in the area of the feature space that we are considering, $p(\mathbf{x}|\omega_T)$ can be used instead of $p(\omega_T|\mathbf{x})$.

Regardless of the fact if a one-class classifier estimates a density or a reconstruction error (distance), for all types the chance of accepting and rejecting a target object, $p(\text{acc } \mathbf{x}|\omega_T)$ and $p(\text{rej } \mathbf{x}|\omega_T)$, are available. Then $p(\omega_T|\mathbf{x})$ is approximated by just two values, f_T and $1-f_T$. The binary outputs of the one-class methods can be replaced by these probabilities. Using just the binary output (accept or reject) the different one-class methods can only be combined by majority voting.

When the more advanced combining rules are required (equations (1) or (2)) $p(\mathbf{x}|\omega_T)$ should be available and a distance or resemblance $\rho(\mathbf{x}|\omega_T)$ should be transformed to a resemblance. Therefore some heuristic mapping has to be applied. One possible transformation is:

$$\tilde{P}(\mathbf{x}|\omega_T) = \exp\left(-\rho(\mathbf{x}|\omega_T)/s\right) \tag{5}$$

(which models a Gaussian distribution around the model if $\rho(\mathbf{x}|\omega_T)$ is a squared distance). The scale parameter s can be fitted to the distribution of $\rho(\mathbf{x}|\omega_T)$. Furthermore it has the advantage that the probability is always bounded between 0 and 1.

2.3 OC Combining Rules

Given a set of R posterior probability estimates, the following set of combining rules can be defined:

First the *mean vote*, which combines the binary (0-1) output labels:

$$y_{mv}(\mathbf{x}) = \frac{1}{R}\sum_{k=1}^{R} I(P_k(\mathbf{x}|\omega_T) \geq \theta_k) \tag{6}$$

Here θ_k is the threshold for method k. When the heuristic method for computing a probability $P_k(\mathbf{x}|\omega_T)$ from a distance $\rho(\mathbf{x}|\omega_T)$ has to be used (equation (5)), the original threshold for the method should also be mapped. For a threshold of 0.5 this rule becomes a majority vote in a two class problem.

The second combining rule is the *mean weighted vote*, where the weighting by $f_{T,k}$ and $1 - f_{T,k}$ is introduced. Here $f_{T,k}$ is the fraction of the target class that is accepted by method k.

$$y_{mwv}(\mathbf{x}) = \frac{1}{R}\sum_{k=1}^{R} \left(f_{T,k} I(P_k(\mathbf{x}|\omega_T) \geq \theta_k) + (1 - f_{T,k}) I(P_k(\mathbf{x}|\omega_T) < \theta_k)\right) \tag{7}$$

This is a smoothed version of the previous version, but it gives identical results when a threshold of 0.5 is applied.

The third is the *product of the weighted votes*:

$$y_{pwv}(\mathbf{x}) = \frac{1}{Z}\prod_{k=1}^{R} f_{T,k} I(P_k(\mathbf{x}|\omega_T) \geq \theta_k) \tag{8}$$

with $Z = \prod_{k=1}^{R} f_{T,k} I(P_k(\mathbf{x}|\omega_T) \geq \theta_k) + \prod_{k=1}^{R} (1 - f_{T,k}) I(P_k(\mathbf{x}|\omega_T) < \theta_k)$

Finally the *mean of the estimated probabilities*:

$$y_{mp}(\mathbf{x}) = \frac{1}{R}\sum_{k=1}^{R} P_k(\mathbf{x}|\omega_T) \tag{9}$$

and the *product combination of the probabilities*:

$$y_{pp}(\mathbf{x}) = \frac{\prod_k P_k(\mathbf{x}|\omega_T)}{\prod_k P_k(\mathbf{x}|\omega_T) + \prod_k P_k(\mathbf{x}|\omega_O)} \tag{10}$$

Here we will use the approximation that the outliers are uniformly distributed $P_k(\mathbf{x}|\omega_O) = \theta_T$. All these combining rules will be compared in a real world one-class problem in the next section.

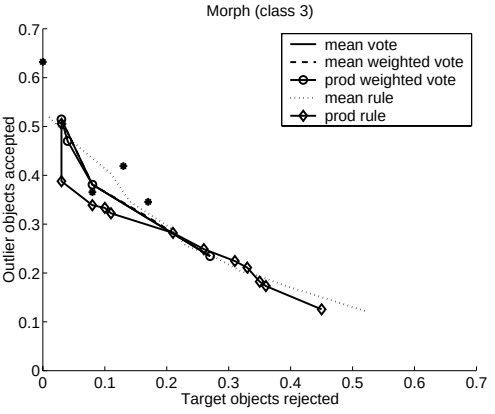

Fig. 1. ROC curves of the five combining rules. Individual classifiers are shown by stars.

2.4 Error

For the evaluation of one-class classifiers and the combination rules, we consider the Receiver-Operating Characteristic curve (ROC curve). It gives the target acceptance and outlier rejection rates for varying threshold values. Note that for estimating the outlier rejection rate, we need example outlier objects. An example is shown in figure 1. Here the results for four individual classifiers trained on one identical feature set and five combination rules are shown. Because each classifier is trained for a 10% target rejection rate, the method is optimized for just one point on the ROC curve (ideally on the vertical line with 10% target rejection rate). These points indicated by the thick dot. The 2-dimensional curves are the ROC curves of the combining rules. The product combination rule performs best here, because for the same fraction of target objects rejected, less outlier objects are accepted than by other methods.

To make comparisons between classifiers a 1-dimensional error is derived from this curve. This is called the Area Under the Curve (AUC) [2], and it measures the total error integrated over (in principle) all threshold values. Because we are mainly interested in situations where we accept large fractions of the target set,

we use threshold values with a target rejection rate from 0.05 to 0.5. Although each classifier is optimized to reject 10% of the target data, during the evaluation of the combination rules, this complete range over the ROC curve is considered.

2.5 Difference Mean and Product Rule

In the combination of normal classifiers it appears that often the more robust average combination rule is to be preferred. Here extreme posterior probability estimates are averaged out. In one-class classification only the target class is modeled and a low uniform distribution is assumed for the outlier class. This makes this classification problem asymmetric and extreme target class estimates are not cancelled by extreme outlier estimates.

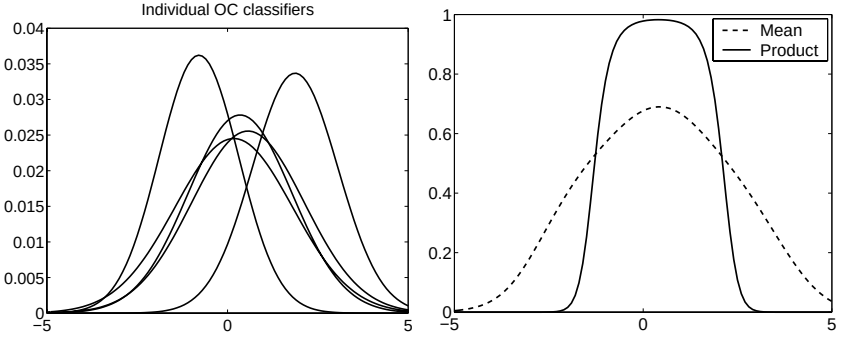

Fig. 2. (Left) Five target probability density estimates which should be combined. (Right) Combination of the five target probability density estimates

In figure 2 five one-class classifiers are shown for an artificial 1-dimensional problem with data normally distributed round the origin (with unit variance). Due to some atypical training samples two of the classifiers are somewhat remote from the other three. In figure 2 the resulting estimates by the product and mean combination rules are shown. The mean combination covers a broad domain in feature space, while the product rule has restricted range. Especially in high dimensional spaces this extra area will cover a large volume and potentially a large number of outliers.

This effect is observable in figure 1. For target rejection rates less than 20% the product combination rule accepts less outlier objects than the mean combination, or other combination rules. This indicates that the covered volume is less than for the other combining rules.

3 Experiments

We will apply the combining rules to one-class classifiers trained on a handwritten digits dataset [5]. This dataset consists of six feature sets: profile features,

Table 1. Results of all the individual classifiers on class 3. Results are multiplied by 100 and averaged over 10 runs. The number between brackets indicates the standard deviation of the outcome.

	profile	Fourier	KL	morph	pixel	Zernike
Gauss	2.98 (1.33)	3.37 (0.88)	1.34 (0.77)	10.76 (0.80)	0.45 (0.34)	6.23 (1.32)
MoG	3.46 (2.17)	3.34 (0.75)	1.66 (1.50)	11.17 (0.82)	0.43 (0.33)	6.60 (1.27)
Parzen	2.26 (0.89)	2.50 (0.73)	0.52 (0.35)	11.03 (0.84)	0.28 (0.27)	3.97 (1.52)
svdd	7.84 (2.51)	3.75 (3.63)	5.13 (2.49)	17.60 (3.48)	1.53 (1.12)	13.07 (3.59)
kmeans	4.17 (1.84)	2.64 (2.04)	1.07 (0.47)	12.56 (1.48)	0.49 (0.23)	8.40 (4.39)
kcenter	4.16 (1.23)	3.78 (3.57)	1.63 (0.77)	17.17 (3.79)	0.74 (0.29)	7.71 (1.65)
autoenc	8.67 (3.93)	3.52 (2.93)	1.93 (1.00)	13.21 (0.80)	0.89 (0.57)	9.99 (2.15)

Fourier features, Karhunen-Loève features, some morphological features, pixel features and Zernike features extracted from the scanned handwritten digits. For the one-class combining problem one class (digit class 3) of handwritten digits is described by the data descriptions and distinguished from all other classes. One hundred training objects are drawn from the target class (no negative examples are used). For testing again 100 objects per class, now both target and outlier classes, are used. This gives thus a total of 100 target and 900 outlier objects. All feature sets are mapped by PCA to retain 90% of the variance in the data. After the PCA all features are scaled to zero mean and unit variance.

All one-class classifiers contain some magic parameters. In the normal distribution the covariance matrix is regularized by $\Sigma' = \Sigma + \lambda \mathbf{1}$ to make inversion of the matrix possible (where λ is taken as small as possible to make inversion possible, most often $\lambda = 1 \cdot 10^{-3}$). The number of clusters in the mixture of Gaussians, the k-means and k-center methods are 5, 10 and 10 respectively. The number of units in the bottleneck layer in the autoencoder network is 5 and the SVDD is trained to reject 10% of the target data. Finally, the width parameter in the Parzen density is optimized using maximum likelihood optimization [9].

In table 1 the AUC-errors of the individual methods are shown for the different feature sets. The first three methods are density estimators, the other four are distance based methods. Different classifiers give different performances, and in most cases the Parzen density estimator performs best. Only for the most difficult dataset, the Morphological dataset the normal distribution performs better (on average). The best individual classifier is the Parzen density estimator, while the easiest dataset to classify is the pixel dataset. Apparently the pixel training set is a representative sample from the true distribution and the number of training objects is sufficient to do a proper density estimation by a Parzen density estimation. Finally note that in some cases the variance is very large!

In table 2 the AUC errors are shown for target class 3 when different classifiers are combined on the same dataset. In the top part of the table the three density methods are combined, the normal density, the mixture of Gaussians and the Parzen density estimation. In these cases the output of the methods do not require any mapping to probabilities. The results show that the product

Table 2. Results of the combination of classifiers by the five combination rules on class 3. Numbers in bold indicate an improvement over the best individual classifier.

Combining 3 density methods						
	profile	Fourier	KL	morph	pixel	Zernike
mv	5.61 (1.29)	11.12 (11.93)	6.84 (13.44)	15.59 (0.99)	23.61 (22.56)	8.03 (1.22)
mwv	5.61 (1.29)	11.12 (11.93)	6.84 (13.44)	15.59 (0.99)	23.61 (22.56)	8.03 (1.22)
pwv	5.61 (1.29)	11.12 (11.93)	6.84 (13.44)	15.59 (0.99)	23.61 (22.56)	8.03 (1.22)
mp	3.00 (1.33)	3.37 (0.89)	1.35 (0.76)	10.85 (0.84)	0.45 (0.34)	6.23 (1.32)
pp	2.72 (1.25)	2.60 (0.62)	0.89 (0.54)	10.92 (0.81)	0.30 (0.30)	4.84 (1.61)
Combining distance methods						
	profile	Fourier	KL	morph	pixel	Zernike
mv	4.23 (1.19)	3.78 (2.71)	1.53 (0.55)	13.51 (0.99)	6.16 (13.68)	**7.03 (2.05)**
mwv	4.33 (1.30)	3.78 (2.70)	1.52 (0.55)	13.45 (1.06)	6.18 (13.68)	**7.16 (2.36)**
pwv	**4.14 (1.19)**	3.81 (2.73)	1.48 (0.53)	13.54 (1.03)	6.15 (13.69)	**6.93 (1.99)**
mp	5.71 (1.55)	2.67 (2.13)	1.42 (1.21)	12.86 (1.63)	**0.48 (0.27)**	7.81 (2.92)
pp	**3.63 (0.81)**	**2.62 (2.07)**	1.14 (0.59)	**11.96 (1.06)**	**0.48 (0.27)**	**6.71 (2.31)**
Combining all methods						
	profile	Fourier	KL	morph	pixel	Zernike
mv1	3.42 (1.18)	5.83 (1.09)	1.19 (0.26)	12.33 (0.65)	1.48 (0.68)	5.96 (1.65)
mwv	3.42 (1.16)	5.84 (1.09)	1.31 (0.58)	12.30 (0.67)	1.47 (0.68)	5.96 (1.62)
pwv	3.44 (1.15)	5.83 (1.10)	1.22 (0.30)	12.34 (0.65)	1.48 (0.68)	6.15 (1.96)
mp	3.23 (0.99)	4.57 (1.83)	1.23 (0.78)	12.29 (1.74)	0.75 (0.56)	7.41 (2.80)
pp	2.55 (0.55)	3.35 (0.73)	0.86 (0.42)	12.12 (1.87)	0.64 (0.71)	4.79 (0.96)

combination rule is a very good combining rule. When the three density methods would estimate approximately the same probability, the mean combination would give a more robust estimate. The fact that the density models vary much, combined with the effect that the mean combination rule tends to increase the estimated target class volume (see section 2.5), causes somewhat worse results than the product combination rule.

In none of the cases the combination rules achieve an improvement over the best individual performances of the one-class classifiers. But in most cases the product combination rule comes close. Only in one case the mean combination rule improves the product combination rule. Furthermore, the first three combination rules are often significantly worse than the last two, indicating that approximating the probabilities by one value is insufficient. Differences between these three rules are very small. They have an averaging behavior and often do not approach the best individual performance.

In the middle part of the table the combining results for the combination of distance methods is shown. Here a mapping to probabilities is performed (by equation (5)). Still most often no improvement over the best individual classifier can be observed, only for the product combination rule reliable improvements can be observed. The individual performances on the Zernike dataset are very poor, and almost all combination rules (except for the mean combination rule) can improve these. The good performance of the product combination rule is also

somewhat surprising, because the classifiers are trained on identical data, while the mapping from distance to probability might introduce extra noise. Because of the large diversity of the methods however, the errors became uncorrelated and extreme estimates are suppressed by the product combination rule.

Finally the last part of table 2 shows the results of combining both the density and distance based methods. Here the best performance never beats the best individual performance (most often the Parzen density estimator). Again it can be observed that the product combination rule performs the best. In most cases adding the distance methods improves the first three combining rules, but deteriorates the last two.

Table 3. ROC errors obtained by combining the same classifiers trained on the six different feature sets.

	Gauss	MoG	Parzen	SVDD	kmeans	kcenters	autoenc
mv	1.7 (2.5)	0.87 (1.3)	**0.12 (0.07)**	7.5 (2.0)	2.8 (3.7)	**0.38 (0.33)**	**0.13 (0.05)**
mwv	1.72 (2.5)	0.8 (1.5)	**0.12 (0.07)**	7.5 (2.0)	3.1 (3.6)	**0.37 (0.33)**	**0.12 (0.05)**
pwv	1.84 (2.5)	0.9 (1.2)	**0.12 (0.07)**	7.5 (2.0)	5.4 (4.3)	**0.36 (0.32)**	**0.12 (0.05)**
mp	1.37 (2.2)	12.0 (1.3)	11.38 (0.82)	2.06 (1.9)	7.2 (4.5)	2.30 (1.12)	**0.43 (0.35)**
pp	**0.41 (0.7)**	**0.2 (0.1)**	**0.07 (0.05)**	2.1 (1.8)	3.1 (4.0)	1.77 (1.45)	**0.42 (0.34)**

Finally in table 3 the results of combining classifiers on different feature sets are shown. Clearly combining different feature sets is more effective than combining different classifiers. Only in some cases the performance is worse than the best individual classifier. For the density methods it is the mean combination rule, while for the three last methods (kmeans, kcenters and the autoencoder network) both the mean and product combination rule perform worse than the first three rules. Here the results on the different feature sets vary very much. It appears that the majority vote and the weighted versions are robust enough to use that.

4 Conclusions

In this paper we investigated the use of combining one-class classifiers. The best individual one-class classifiers in this problem appears to be the Parzen density estimator on the pixel dataset. Improving the results of the Parzen estimator appears to be hard, because the training sample in this dataset appears to be a representative sample from the "true" distribution. As can be expected, combining classifiers trained in different feature spaces is the most useful. Here the different feature sets contain much independent information which often results in good classification results. In most situations the product combination rule gives the best results. Approximating the probability by just two values does often harm the combination rules, so it is useful to use the complete density, or distance to the model. The mean combination rule suffers from the fact that the area covered by the target set tends to be overestimated, thus more outlier objects are accepted than is necessary.

Acknowledgments. This work was partly supported by the Foundation for Applied Sciences (STW) and the Dutch Organization for Scientific Research (NWO).

References

1. J.A. Benediktsson and P.H. Swain. Consensus theoretic classification methods. *IEEE Transactions on Systems, Man and Cybernetics*, 22(4):688–704, July/August 1992.
2. A.P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.
3. G.A. Carpenter, S. Grossberg, and D.B. Rosen. ART 2-A: an adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, 4(4):493–504, 1991.
4. R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
5. R.P.W. Duin. UCI dataset, multiple features database. Available from ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/, 1999.
6. N. Japkowicz. *Concept-Learning in the absence of counter-examples: an autoassociation-based approach to classification*. PhD thesis, New Brunswick Rutgers, The State University of New Jersey, 1999.
7. J. Kittler, R.P.W. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):226–239, 1998.
8. J. Kittler, A. Hojjatoleslami, and T. Windeatt. Weighting factors in multiple expert fusion. In Clark A.F., editor, *Proceedings of the 8th British Machine Vision Conference 1997*, pages 41–50. University of Essex Printing Service, 1997.
9. M.A. Kraaijveld and R.P.W. Duin. A criterion for the smoothing parameter for parzen-estimators of probability density functions. Technical report, Delft University of Technology, September 1991.
10. M.R. Moya, M.W. Koch, and L.D. Hostetler. One-class classifier networks for target recognition applications. In *Proceedings world congress on neural networks*, pages 797–801, Portland, OR, 1993. International Neural Network Society, INNS.
11. M. Tanigushi and V. Tresp. Averaging regularized estimators. *Neural Computation*, 9:1163–1178, 1997.
12. L. Tarassenko, P. Hayton, and M. Brady. Novelty detection for the identification of masses in mammograms. In *Proc. of the Fourth International IEE Conference on Artificial Neural Networks*, volume 409, pages 442–447, 1995.
13. D.M.J. Tax and R.P.W Duin. Data domain description using support vectors. In M. Verleysen, editor, *Proceedings of the European Symposium on Artificial Neural Networks 1999*, pages 251–256. D.Facto, Brussel, April 1999.
14. D.M.J. Tax and R.P.W Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, December 1999.
15. A. Ypma and R.P.W. Duin. Support objects for domain approximation. In *ICANN'98*, Skovde (Sweden), September 1998.