A combining strategy for ill-defined problems

Thomas Landgrebe, David M.J. Tax, Pavel Paclík, Robert P.W. Duin, and Colin Andrew

Elect. Eng., Maths and Comp. Sc.,
Delft University of Technology, The Netherlands
{t.c.w.landgrebe, d.m.j.tax, p.paclik, r.p.w.duin}@ewi.tudelft.nl
colin.andrew@ieee.org

Abstract

In this paper we present a combining strategy to cope with the problem of classification in ill-defined domains. In these cases, even though a particular *target* class may be sampled in a representative manner, an *outlier* class may be poorly sampled, or new *outlier* classes may occur that have not been considered during training. This may have a considerable impact on classification performance. The objective of a classifier in this situation is to utilise all known information in discriminating, and to remain as robust as possible to changing conditions. A classification scheme is presented that deals with this problem, consisting of a sequential combination of a one-class and multi-class classifier. We show that it can outperform the traditional classifier with reject-option scheme, locally selecting/training models for the purpose of optimising the classification and rejection performance.

1. Introduction

Consider a problem in which a *target* class is to be discriminated with respect to an *outlier* class. In many applications, both classes are sampled as a set of measurements in order to construct a training set that represents the class. A classifier can then be designed, for example, by estimating the class conditional densities for both classes. Good estimates allow for an optimal tradeoff to be made between the classes. However in some applications some classes may not be well-defined. In this paper we assume that the *target* class is well represented, but the *outlier* class is not. This may be due to a variation in *outlier* class distribution, such as *sensor drift* [1], or new *outlier* classes may be present that were not represented during training. Examples of this phenomenon include:

- Diagnostic problems in which the objective of the classifier is to identify abnormal operation (*outlier* class) from normal operation (*target* class) [2]. It is often the case that a representative training set can be gathered for the *target* class, but due to the nature of the problem the *outlier* class cannot be sampled in a representative manner. For example in machine fault diagnosis [3] a destructive test for all possible abnormal states may not be feasible.
- Recognition systems often involve a detection and classification stage. An example is road sign classification, in which a classifier needs not only to discriminate between examples of road sign classes, but must also reject non-sign class examples [4]. Gathering a representative set of non-signs may not be possible. Similarly face detection [5], where a classifier must deal with well-defined face classes, and an ill-defined non-face class, as well as handwritten digit recognition [6], where non-digit examples are a serious issue.

The goal of a classifier in these cases is to obtain a high true positive rate and low false positive rate, with respect to the *target* class. Even though the *outlier* class is poorly defined, we would still like to make use of all knowledge that exists for the problem (to account for known class overlaps). Thus the objective is to obtain a high *classification performance*, and robustness to changes in the *outlier* class (referred to as *rejection performance*). Formalisation and consequences of this problem are given in Section 2.

Previous work in this area has typically been the *classifier with reject-option*, first proposed by Chow in [7], often called the *ambiguity reject-option.* In this reject-option, when the cost of misclassification is higher than the cost of rejection, the example in question should be rejected, based on thresholding of the posterior probability. This reject-option is applicable for handling ambiguity between classes (examples close to the *target* class), which is not of interest here. In this paper we are interested in rejecting examples occurring far away from the *target* class. Dubuisson and Masson proposed the *distance reject-option* in [2]. This rejection scheme was designed to cope with the condition in which new classes are present that are not represented during training, introducing a different type of *reject* class ω_r. New examples situated a particular distance (based on a reject threshold t_d) from known class centroids are rejected. A similar procedure can be applied to density-based classifiers, except here the class conditional density is thresholded. In this way a *closed* decision surface is obtained, providing protection against new unseen classes[1]. New classes will be rejected if they fall outside the class description. Thus to minimise the probability of accepting examples from class ω_r, assuming they are uniformly distributed in feature space, the volume of the description should be minimised. The reject-option is discussed further in Section 3.

The limitation of the reject-option approach is that a model chosen for good classification performance does not necessarily imply good rejection performance. The opposite is also true. Improved performance may result from a practitioner viewpoint if an adequate evaluation methodology is used. However as will be discussed later, since the same model is used for classification and rejection, we may have to sacrifice the performance of one for the other. In this paper we present a classification strategy that can in some cases alleviate this situation, consisting of a sequential combination of one-class and multi-class classifiers (called *SOCMC*). The proposed 2-stage scheme allows both rejection and classification performance to be explicitly modified by varying the respective models and representations. Thus a classifier model can be designed to obtain good performance

[1] This thresholding of a single class model is equivalent to one-class classification [8].

on known classes, and a separate classifier model to improve robustness with respect to unknown classes. The SOCMC is discussed in Section 4.

A number of experiments are performed to investigate the SOCMC approach in Section 5. All experiments benchmark SOCMC results with the distance-based reject option, as well as with traditional discriminant-based approaches. Experiments are performed on a number of real datasets, showing the applicability of the new approach. Finally, conclusions are given in Section 6.

2. Ill-defined problems

To formalise this problem, we assume that there is a well defined *target* class ω_t, and the *outlier* class is composed of two classes ω_o and ω_r, where the former consists of known class information, and the latter of unknown information (called the *reject* class). Note that in this setup, we classify examples considered to be either ω_o and ω_r as *outlier*. Examples of each class are composed of vectors of measurements $\mathbf{x}$ with dimensionality d. It is assumed that $\mathbf{x}$ is represented by a feature space χ (later we discuss classifiers that operate on the data in new feature spaces, consisting of various mappings of the original space χ). The unconditional density $p(\mathbf{x})$ can then be written as in Equation 1.

$$p(\mathbf{x}) = p(\omega_t)p(\mathbf{x}|\omega_t) + p(\omega_o)p(\mathbf{x}|\omega_o) + p(\omega_r)p(\mathbf{x}|\omega_r) \quad (1)$$

To evaluate classifiers in this situations, two performance measures are of interest:

1. The classification performance (performance between known classes/data), denoted perf(ω_t, ω_o).
2. The rejection performance (performance between the ω_t and ω_r), denoted perf(ω_t, ω_r).

Ideally both perf(ω_t, ω_o) and perf(ω_t, ω_r) should be high. Note that estimation of perf(ω_t, ω_r) is not straightforward, since this class is by definition absent during training. In the experimental Section 5 a methodology is given to provide some estimate of this. In Figure 1 an example of this problem is shown, demonstrating the weakness of general discrimination approaches with respect to this problem. Here a synthetic dataset has been constructed in two dimensions. In the left image, the training set consisting of ω_t and ω_o is shown, upon which a Bayes quadratic classifier is shown. In the right image the testing situation is shown, in which a new class ω_r is present. The classifier is clearly not robust to these changes in conditions.

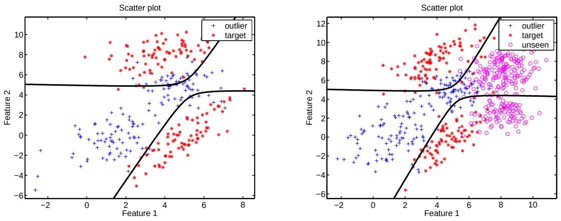

Figure 1: Illustrating a discrimination classifier applied to the problem in which a new unseen class is present in testing. The left plot shows the classifier decision boundary for the training data, and the right plot for the testing data, in which a new class ω_r is present. A Bayes quadratic classifier is used.

Two classification approaches are utilised in this paper. The first are multi-class classifiers (sometimes referred to as MCC's/discriminators). In this paper, we deal specifically with two-class discriminant classifiers, denoted D_{MCC}. A classifier trained on ω_t and ω_o can be defined as in Equation 2, with $\hat{p}(\omega|\mathbf{x})$ representing an estimate of the posterior probability of class ω. These classifiers result in an open decision boundary, since it is assumed that ω_t and ω_o are well represented.

$$D_{MCC} : \begin{cases} \textit{target} & \text{if } \hat{p}(\omega_t|\mathbf{x}) > \hat{p}(\omega_o|\mathbf{x}) \\ \textit{outlier} & \text{otherwise} \end{cases} \quad (2)$$

The second classification approach used is one-class classification (sometimes referred to as OCC's), denoted D_{OCC} [8]. These classifiers are trained on only a single class, resulting in a closed description of the class density or domain. No assumptions about other classes are made, and thus these classifiers do not make a trade-off between overlapping classes. The decision boundary is however constrained/closed, i.e. all objects situated outside the class description are rejected as outliers, providing protection against new, unseen classes. The OCC description/model is trained, with some allowance made for outliers in the training set by adjusting a decision threshold θ. The D_{OCC} can be written as in Equation 3, classifying all objects as either *target* or *outlier*.

$$D_{OCC} : \begin{cases} \textit{target} & \text{if } \hat{p}(\mathbf{x}|\omega_t) > \theta \\ \textit{outlier} & \text{otherwise} \end{cases} \quad (3)$$

3. The classifier with reject-option

As previously mentioned, the original reject option (*ambiguity* reject) [7] rejects objects that are considered to be ambiguous, based on a threshold t_d. For an incoming test object, the classifier assigns a class label. The relevant posterior of the assigned class is examined and compared to t_d. Examples are either assigned to an $ACCEPT$ region $\Re_{accept}$ or $REJECT$ region $\Re_{reject}$, as shown in Equation 4.

$$\begin{aligned} \Re_{accept} &= \{\mathbf{x}| \max_i p(\omega_i|\mathbf{x}) \geq t_d\}, \; i \in \{t, o\} \\ \Re_{reject} &= \{\mathbf{x}| \max_i p(\omega_i|\mathbf{x}) < t_d\}, \; i \in \{t, o\} \end{aligned} \quad (4)$$

With the *distance reject option*, the conditional density of the class of interest is thresholded, resulting in a *closed* decision boundary[2], providing protection against unseen classes. Again a two-stage procedure is undertaken. In the first stage an example is assigned to a particular class $\omega_i, i = t, o$, referring to *target* and *outlier*, using Bayes rule. In the second step, if the example has been assigned to the *target* class, the conditional probability $p(\mathbf{x}|\omega_t)$ is thresholded via a reject threshold t_d. Examples exceeding this threshold are rejected. Examples are either assigned to an $ACCEPT$ or $REJECT$ region, $\Re_{accept}$ and $\Re_{reject}$ as shown in Equation 5.

$$\begin{aligned} \Re_{accept} &= \{\mathbf{x}|p(\mathbf{x}|\omega_i) \geq t_d\}, \; i \in \{t, o\} \\ \Re_{reject} &= \{\mathbf{x}|p(\mathbf{x}|\omega_i) < t_d\}, \; i \in \{t, o\} \end{aligned} \quad (5)$$

The *distance* reject option is illustrated on a simple example in Figure 2. The left plot shows a model based on a linear classifier, and the right image a mixture-of-Gaussians classifier with 15 mixtures. It is clear that a closed boundary results, and the

[2] For classifiers that are not density-based such as k-Nearest Neighbour, Dubuisson and Masson proposed to reject based on the mean distance to the k nearest neighbours. In this case a meaningful threshold should be chosen based on the scale of the distances.

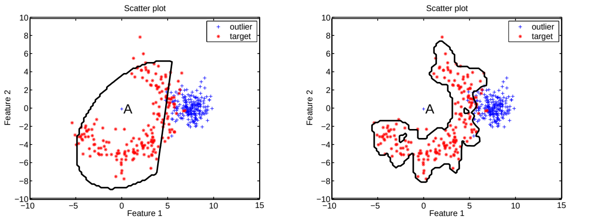

Figure 2: Illustrating the *distance* reject option classifier in a two-class 2D example, showing a linear classifier model in the left plot which results in good classification performance, but poor rejection performance. The right image depicts a more complex mixture-of-Gaussians classifier, resulting in good rejection, but poor classification. The decision boundary indicates the threshold used for class assignment. Poor models have purposefully been chosen for the sake of illustration to simulate realistic conditions.

trade-off between known classes is accounted for. We discuss two situations that could lead to sub-optimal performance.

In the first situation we discuss the practitioner. If the practitioner designs a classifier based on knowledge of the ω_t and ω_o classes only, a situation such as that depicted in the left figure may result. Here the classifier obtains near optimal classification performance, but since the model is not chosen explicitly to fit the *target* class distribution, sub-optimal rejection results. Thus we propose to evaluate classifiers in these situations based on both classification and rejection performance. This may lead to choosing more appropriate models. For example some classifiers focus on discrimination only, discarding domain information (e.g. support-vector classifier). A better choice would be to choose models modeling the distribution (e.g. mixture-of-Gaussians density estimation).

In the second situation we assume the practitioner is aware of an adequate evaluation methodology. In this case the practitioner will focus on obtaining the best-possible rejection/classification performance. In real problems, typical limitations are that the training set size is limited, the input dimensionality high, and computation time limited. In these situations, choosing a model that results in high classification performance (i.e. focus on known overlapping regions) may be at the expense of a worse performance in terms of rejection performance e.g. the class conditional density may be well estimated in the overlapping region, but poor in other areas. This is depicted in the left plot of Figure 2 where a new *outlier* example marked *A* on the plot will be incorrectly classified as *target*. Similarly, the classification performance may be compromised for the case in which a model is chosen for good rejection performance (right plot). In Section 4 a classification scheme is presented in which different models can be selected/trained explicitly for classification and rejection respectively. We argue that in some cases it is better to choose a local model suitable to perform the classification, and another for rejection. This flexibility is lacking in the reject-option case.

4. Sequential combining of a one-class and multi-class classifier

We present a classification scheme here consisting of the sequential combining of one-class and multi-class classifiers (SOCMC). The rationale is that the class model and representation used in the first stage (denoted D_{OCC}) can be explicitly chosen for the purpose of rejection i.e. between ω_t and ω_r. In a similar way the second stage classifier D_{MCC} can be chosen locally in the area of known overlap to obtain good classification performance between known classes, i.e. between ω_t and ω_o. The SOCMC classifier is depicted in the block diagram in Figure 3. In the first stage, the one-class classifier D_{OCC} attempts to detect all *target* examples from $p(x)$, given a test set. A one-class classifier [8] is appropriate for this stage since it protects against unseen classes ω_r, and capitalises on the knowledge that the *target* class is well defined by the training set[3]. At this stage it is not important if examples of the class ω_o are incorrectly accepted, since we rely on this discrimination in the second stage. Thus it is assumed that the output of D_{OCC}, denoted $\tilde{\mathbf{x}}$ will consist only of examples of class ω_t and ω_o, with all ω_r having been rejected (as well as ω_o examples that do not overlap with the *target* OCC description.).

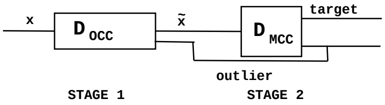

Figure 3: Block diagram of the *SOCMC* classifier. The first stage classifier D_{OCC} consists of an OCC, trained on the well defined *target* class. The second stage classifier D_{MCC} is a multi-class discriminant trained on examples considered to be *target* by the first stage.

Note that both the representation and class description model can be selected/trained to improve the rejection performance perf(ω_t, ω_r). The D_{OCC} represents the input data $\mathbf{x}$, derived from the feature space χ, by a new representation χ_{OCC} (D_{OCC} consists of both a representation and classification stage). The classifier can thus be written as $D_{OCC}(\mathbf{x}_{OCC})$, defined as in Equation 3 for class ω_t. The output $\tilde{\mathbf{x}}$ is then shown in Equation 6.

$$\tilde{\mathbf{x}} = \{\mathbf{x} | D_{OCC}(\mathbf{x}_{OCC}) = \textit{target}\} \qquad (6)$$

The output $\tilde{\mathbf{x}}$ is then applied to the second stage classifier D_{MCC}. Note that D_{OCC} is used to select objects for the second stage. We still have the opportunity to optimise the representation and model selection used in the second stage. Thus $\tilde{\mathbf{x}}$ is used by D_{MCC} in the original representation χ. The D_{MCC} classifier is trained on the data $\tilde{\mathbf{x}}$, which is assumed to be a mixture of data from ω_t and ω_o only, which are represented by the training set. A discriminator is thus trained, with the objective of obtaining an optimal trade-off in terms of class overlap. As with the D_{OCC}, the representation and classification model can be chosen, but in this case for the purpose of optimising the classification performance perf(ω_t, ω_o). A model is trained focused on the local region, specified by a training dataset $\tilde{(\mathbf{x})}_{tr}$. The input data $\tilde{\mathbf{x}}$ that is represented by χ is now mapped to a new representation space χ_{MCC}, resulting in the classifier $D_{MCC}(\mathbf{x}_{\tilde{MCC}})$, defined as in Equation 2 between classes ω_t and ω_o. The final *SOCMC* classifier, denoted D_{SOCMC} is de-

[3] New classes will be rejected if they fall outside the class description. Thus to minimise the probability of accepting examples from class ω_r, assuming they are uniformly distributed, the volume of the description should be minimised.

fined in Equation 7.

$$D_{SOCMC}(\mathbf{x}|D_{OCC}, D_{MCC}) = \begin{cases} \textit{outlier} \text{ if } D_{OCC}(\mathbf{x}) = \textit{outlier} \\ D_{MCC}(\tilde{\mathbf{x}}_{MCC}) \text{ otherwise} \end{cases} \quad (7)$$

We illustrate the operation of the *SOCMC* classifier in the same situation as in Section 3, in Figure 4. We noticed that the classifier D1 in the left plot of Figure 2 resulted in high classification performance, and low rejection performance. The opposite was true for the classifier D2 in the right plot. In the SOCMC classifier, we select/train specific local models for the purposes of classification and rejection respectively, illustrating that the SOCMC can in some cases improve performance. In this example the model used for D1 is chosen for the D_{MCC} stage (i.e. a linear classifier), and the D2 model is used for the D_{OCC} stage (Mixture-of-Gaussians with 6 mixtures). This classifier results in a good classification and rejection performance. A number of training considerations need to be made for the *SOCMC* classifier:

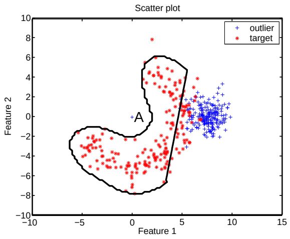

Figure 4: Illustrating the *SOCMC* classifier. A linear model has been chosen locally in the area of overlap for good classification performance, and a Gaussian-mixture model with 15 mixtures is used for rejection, showing that example *A* is correctly classified.

- Training set size for D_{MCC}: If a training set $\mathbf{x}_{tr}$ is used, only a subset $\tilde{\mathbf{x}}_{tr}$ will be available for the D_{OCC}. If $\mathbf{x}_{tr}$ is small, or $\tilde{\mathbf{x}}_{tr} \ll \mathbf{x}_{tr}$, there may not be sufficient samples to train D_{MCC}. This may limit the complexity of the model/representation used. Alternatively the entire $\mathbf{x}_{tr}$ could be used to train the D_{MCC}.
- Training technique: The *SOCMC* classifier is analogous to the trained combiner used in classifier combining, as discussed in [9]. If the same training set $\mathbf{x}_{tr}$ is used to train both D_{OCC} and D_{MCC}, the D_{SOCMC} could overfit to the noise in the training set. An alternative training strategy could be to split $\mathbf{x}_{tr}$ into two independent training sets $\mathbf{x}_{tr1}$ and $\mathbf{x}_{tr2}$, with the first used to train D_{OCC}, and the second used to train D_{MCC}. This may generalise better, but may actually be worse than the former strategy when the training set size is small.

5. Experiments

5.1. Evaluation

Since the exact nature of the *outlier* conditional distribution cannot be predicted in advance, estimating perf(ω_t, ω_r) is not straight forward. We propose an evaluation method to provide some confidence as to the robustness of the classifier, and to compare classifiers. The evaluation assumes a uniform *outlier* distribution. This test allows a classifier to be evaluated assuming that *outlier* examples can occur anywhere in feature space around the *target* class. It provides a measure for how well the classifier protects the *target* class (in the respective feature space) from changing conditions. However for real high dimensional problems, the number of artificial examples to be generated may be computationally prohibitive, so two methods of artificial data generation are used in real experiments to attempt to overcome this problem:

1. In the first method, called perf$_{a1}(\omega_t, \omega_r)$, a number of *outlier* examples are artificially generated uniformly in a sphere around a subspace of the *target* class [10]. Here examples are generated within a PCA (Principal Component Analysis) subspace. The original data is scaled to unit variance, and the artificial data is then generated within this space with a radius of 1.1 of the covariance of the *target* class. These can be mapped into the original space by an inverse of the PCA mapping.
2. Similar to the previous analysis, except data is generated in the original representation, following a Gaussian distribution. Here examples are generated around the *target* class, using an enlarged covariance matrix of the *target* class. The covariance matrix is enlarged by a fraction of 1.5 (this is simply a multiplication of the covariance matrix to spread the new generated examples further). The test is called perf$_{a2}(\omega_t, \omega_r)$.

The perf(ω_t, ω_o) measure relates to the known classes ω_t and ω_o. This performance is approximated using standard techniques. For all experiments a 20-fold cross-validation procedure is carried out, and the primary performance measure used is the AUC (Area under the Receiver-Operator Curve). The variance of the estimates is depicted in terms of the standard deviation. To summarise, the following performance measures are computed for each experiment:

- perf(ω_t, ω_o), estimated using cross-validation with 20-folds, computing the respective AUC.
- perf$_{a1}(\omega_t, \omega_r)$, estimated using 20-fold cross-validation procedure. In testing, for each fold an independent *target* portion of $\mathbf{x}$ is used, together with the generated artificial *outlier* data that was not used for training. Again the AUC is computed.
- perf$_{a2}(\omega_t, \omega_r)$, estimated as per perf$_{a1}(\omega_t, \omega_r)$.

5.2. Dataset description

A number of real-world datasets are used in the experimentation. These datasets have been selected based on their relevance to this problem. The following datasets are used:

1. *Face-amsterdam (Face)*: This dataset consists of a face class ω_t, and non-face class ω_o, and is described in [5], and downloaded at [11]. Each face is stored as a 20 × 20 image. Only the first 1000 faces from the face database, and the first 1000 non-faces from the non-face test database are used. This dataset is used because it can be argued that finding a representative set of non-face examples may be infeasible.
2. *Mfeat-Fou Digit4 (Mfeat)*: This is a dataset consisting of examples of ten handwritten digits, which can be found in [12]. In this dataset, Fourier components have been extracted from the original images, resulting in a 76-dimensional representation of each digit. 200 examples of each digit are available. In these experiments, digit *4* is used as the *target* class, and all the others as *outlier*.

3. *Geophysical (Geo)*: A multi-modal dataset, in which a *target* and *outlier* class are represented by spectra. In this problem, new *outlier* classes may appear during testing. 3982 *target* examples exist, and 3675 *outlier* examples.

5.3. Results

The results for a number of experiments on the real-world datasets are now presented. The objective of the experiments is to assess the SOCMC classifiers on the real-world problems to ascertain whether they do in fact outperform conventional discriminant-based classifiers. This paper also shows that the SOCMC classifiers can result in higher performance than the distance-based reject-option classifiers. In each experiment, SOCMC results are shown benchmarked against discriminant and reject-option classifiers. For a fair comparison, the same model and representation used for the discriminant classifier is used in the reject-option classifier, and also in the multi-class stage D_{MCC} of the SOCMC. A number of different D_{OCC} models are then chosen to attempt to improve the rejection performance perf(ω_t, ω_r), with only the best results shown for brevity (there are examples where SOCMC classifiers do not work – some optimisation is required is to select appropriate models). The reject threshold for the reject-option and one-class classifiers is fixed to reject 5.00% of *target* examples for the given training set. As a starting point for the comparison, it is important to note that the SOCMC classifier results in a similar performance to the reject-option classifier when the same model (i.e. same representation and data model) is used for both the D_{OCC} and D_{MCC} results. Small differences in results are attributed to the fact that only a subset of $\mathbf{x}$ is used to train the D_{MCC}. These results are not included due to space constraints.

In Table 1 details of each experiment are shown. The first column indicates the dataset used, and the second column the model used for the discriminant classifier **M**, the reject-option classifier **R** and the D_{MCC} stage of the SOCMC classifier **S**. The last column shows the representation and classifier used for the D_{OCC} of the SOCMC. For each classifier, three performance results are shown (in terms of mean AUC[4] over 20-folds with standard deviations shown). These consist of the perf(ω_t, ω_o), perf$_{a1}(\omega_t, \omega_o)$ and perf$_{a2}(\omega_t, \omega_r)$ measures, denoted clf, $rj1$, and $rj2$ respectively. Ideally, all three performances should approach 1.00.

First we discuss the *face* results in Figure 5. In the first experiment *face A*, it can be seen that the discriminant classifier **MA** has a rejection performance ($rj1$ and $rj2$) that is much lower than the classification performance clf. This is attributed to the fact that the *target* decision space is unconstrained, providing little protection against changing conditions. The reject-option classifier **RA** then shows a marked improvement in rejection performance in terms of test $rj1$, with a small decrease in clf. This sacrifice of classification performance for improved rejection performance alludes to a tradeoff between these two measures. The poor performance on $rj2$ was unexpected at first, but on closer inspection of the model used (QDC), which assumes unimodality, and the fact that data generated in the $rj2$ test is also distributed in a uniform manner only in the region of the *target* class, may provide an adequate explanation. These results only show marginal (but significant at times) improvements of the SOCMC classifier over the reject-option. It is suspected that this dataset is largely unimodal, and close to Gaussian-distributed (and the outliers in $rj2$ are generated in a similar fashion). In the first experiment, the **SA** performances in terms of clf and $rj1$ are slightly better than **RA**. In the second experiment *face B*, **SB** results in a much higher rejection performance than **RB**, but with some loss in classification performance. Again we observe a trade-off between classification and rejection performance. The third experiment once again shows small improvements over the reject-option with respect to **SC**.

Dataset	Base algorithm	SOCMC D_{OCC} model
Face A	PCA 0.99 QDC	PCA 0.99 Gauss
Face B	Fisher-map QDC	PCA 0.99 MoG-8
Face C	PCA 0.99 LDC	PCA 0.99 Gauss
Mfeat A	Nearest-mean	Gauss
Geo A	PCA 0.9 QDC	PCA 0.9 MoG-5
Geo B	PCA 0.999 QDC	PCA 0.999 MoG-5
Geo C	PCA 0.9 MoG-5/class	PCA 0.999 MoG-5

Table 1: Description of experiments. The first column shows the dataset used. In the second column the algorithm used for the discriminant classifier **M**, the reject-option classifier **R** and the D_{MCC} stage of the SOCMC classifier **S** is given. The last column shows the representation and classifier used for the D_{OCC} of the SOCMC. PCA is a principal component analysis mapping, followed by the percentage of retained variance. Gauss is a Gaussian model. MoG-N is a Mixture-of-Gaussians model with N mixtures. LDC and QDC are Bayes linear and quadratic classifiers respectively.

In the left-most plot of Figure 6, the results of the *mfeat-fou digit4* experiments are shown. Here a nearest-mean classifier has been used, resulting in a 92.44% AUC classification performance for **MA**. The rejection performances are however around 50.00%. The reject-option classifier **RA** is not significantly better than **MA** at rejection. In this case a large number the outliers generated were accepted by a clearly sub-optimal rejection model, even though the classification performance is high. However the SOCMC classifier performs much better here. Even though a nearest-mean classifier is used for classification, the Gaussian model is much better at rejection. Low performances on $rj2$ suggest again that the *target* data is unimodal, with most generated *outlier* examples falling within the domain of the *target* class.

In the three right-most plots in Figure 6, the results of the *geophysical* experiments are shown, showing considerable improvements achieved by the SOCMC scheme. In *Geo A* it can be seen that both *RA* and *SA* improve in terms of $rj1$ performance. However the SOCMC is much better at $rj2$ performance. In this case, the D_{OCC} model used was a Mixture-of-Gaussians, that could model the apparent multi-modality of the *target* class, and thus provide better protection against the *outlier* examples generated in $rj2$. The reject-option rejection model was constrained to the unimodal QDC. In the second experiment *Geo B*, a good example of the SOCMC approach is shown (see **RB** and **SB**), with a clear performance improvement over the reject option. The third experiment shows that the SOCMC and reject option classifiers result in a similar performance, with a slightly better $rj2$ performance achieved by the SOCMC. We conclude that a strong classification model (fitting the data well) will result in optimal classification and reject performance. It was observed that a discriminator can indeed obtain high classification performance, but a model chosen for good classification performance can be at the expense of rejec-

[4] where an ideal performance in a separable problem would result in an AUC score of 1.

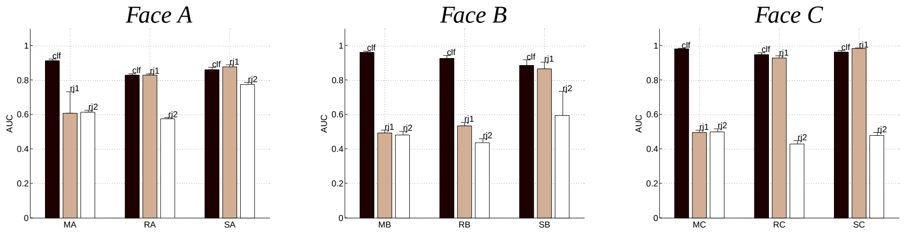

Figure 5: Summarised results of the *face-amsterdam* experiments.

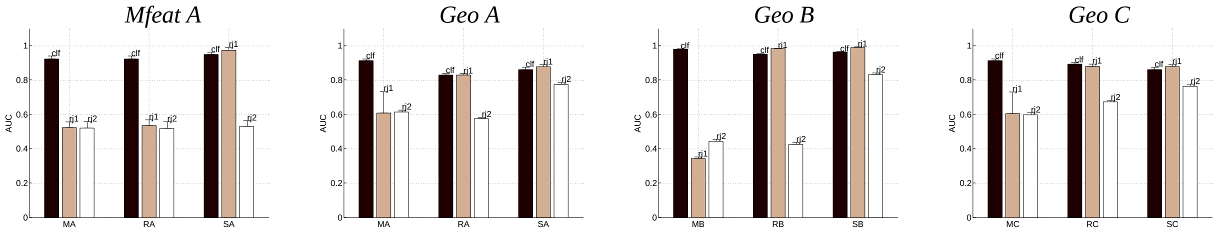

Figure 6: Summarised results of the *mfeat-fou digit4* and *geophysical* experiments.

tion performance. The SOCMC results showed that this classifier can improve upon the reject-option, with separate models trained locally for the purposes of classification and rejection respectively.

6. Conclusions

In this paper classification strategies for ill-defined problems was discussed. It was assumed that a well defined *target* class is to be discriminated from an ill-defined *outlier* class. First the implications on performance with respect to standard discrimination approaches was discussed, showing that a closed/constrained decision space around the *target* class is necessary for robustness to changing conditions. The state-of-the art classifier suited to this task is the distance-based reject option. It was pointed out that a practitioner should make use of an adequate evaluation methodology in selecting a classifier, considering both classification and rejection performance. A new classification strategy was proposed for these types of problems, involving the sequential combination of one-class and multi-class classifiers. These classifiers allow a model to be explicitly selected/trained in local regions of known overlap to emphasise either classification or rejection performance. Experimentation on a number of real-world datasets showed that in some cases the SOCMC classifier does indeed outperform the distance-based reject-option approach. An observation made during experimentation is that an inherent trade-off occurs between classification and rejection. Optimising this will be a focus of future research.

7. Acknowledgments

This research is/was supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.

8. References

[1] K. Copsey and A. Webb, “Classifier design for population and sensor drift,” *Joint IAPR Workshops on Syntactical and Structural Pattern Recognition, and Statistical Pattern Recognition*, pp. 744–752, August 2004.

[2] B. Dubuisson and M. Masson, “A statistical decision rule with incomplete knowledge about classes,” *Pattern Recognition*, vol. 26, no. 1, pp. 155–165, 1993.

[3] A. Ypma, D.M.J. Tax, and R.P.W. Duin, “Robust machine fault detection with independent component analysis and support vector data description,” *Proceedings of the 1999 IEEE Workshop on Neural Networks for Signal Processing, Madison*, 1999.

[4] P. Paclík, “Building road sign classifiers,” *PhD thesis, CTU Prague, Czech Republic*, 2004, To appear.

[5] T. V. Pham, M. Worring, and A. W. M. Smeulders, “Face detection by aggregated bayesian network classifiers,” *Pattern Recognition Letters*, vol. 23, no. 4, pp. 451–461, February 2002.

[6] C.L. Liu, H. Sako, and H. Fujisawa, “Performance evaluation of pattern classifiers for handwritten character recognition,” *International Journal on Document Analysis and Recognition*, pp. 191–204, 2002.

[7] C.K. Chow, “On optimum error and reject tradeoff,” *IEEE Transactions on Information Theory*, vol. It-16, no. 1, pp. 41–46, 1970.

[8] D.M.J. Tax, “One-class classification,” *PhD thesis, TU Delft, The Netherlands*, June 2001.

[9] R.P.W. Duin, “The combining classifier: To train or not to train?,” *ICPR16, Proceedings 16th International Conference on Pattern Recognition (Quebec City, Canada), IEEE Computer Society Press, Los Alamitos*, vol. 2, pp. 765–770, August 2002.

[10] D.M.J. Tax and R.P.W. Duin, “Uniform object generation for optimizing one-class classifiers,” *Journal for Machine Learning Research*, pp. 155–173, 2001.

[11] T. V. Pham, M. Worring, and A. W. M. Smeulders, “Face database,” http://carol.wins.uva.nl/ vietp/publication/list.html.

[12] “Mfeat,” *ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/*.