ARTICLE IN PRESS

Available online at www.sciencedirect.com
SCIENCE DIRECT®

Pattern Recognition Letters xxx (2005) xxx–xxx

Pattern Recognition Letters

www.elsevier.com/locate/patrec

The interaction between classification and reject performance for distance-based reject-option classifiers

Thomas C.W. Landgrebe *, David M.J. Tax, Pavel Paclík, Robert P.W. Duin

Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands

Abstract

Consider the class of problems in which a *target* class is well-defined, and an *outlier* class is ill-defined. In these cases new *outlier* classes can appear, or the class-conditional distribution of the *outlier* class itself may be poorly sampled. A strategy to deal with this problem involves a two-stage classifier, in which one stage is designed to perform discrimination between known classes, and the other stage encloses known data to protect against changing conditions. The two stages are, however, interrelated, implying that optimising one may compromise the other. In this paper the relation between the two stages is studied within an ROC analysis framework. We show how the operating characteristics can be used for both model selection, and in aiding in the choice of the reject threshold. An analytic study on a controlled experiment is performed, followed by some experiments on real-world datasets with the distance-based reject-option classifier.

Keywords: Ill-defined classification problems; Unseen classes; Reject-option; Model selection; ROC analysis

1. Introduction

In pattern recognition, a typical assumption made during the design phase is that the various classes involved in a particular problem can be sampled reliably. However, in some problems, new classes or clusters may appear in the production phase that were not present during the design/training. In other problems, some classes may be sampled poorly, leading to inaccurate class models. Examples of applications that are affected by this are for instance:

- Diagnostic problems in which the objective of the classifier is to identify abnormal operation from normal operation (Dubuisson and Masson, 1993). It is often the case that a representative training set can be gathered for one of the classes, but due to the nature of the problem, the other class cannot be sampled in a representative manner. For example, in machine fault diagnosis (Ypma et al., 1999) a destructive test for all possible abnormal states may not be feasible or very expensive.
- Recognition systems that involve a rejection and classification stage, for example, road sign classification. Here a classifier needs not only to discriminate between examples of road sign classes, but must also reject non-sign class examples (Paclík, 2004). Gathering a representative set of non-signs may not be possible. Similarly face detection (Pham et al., 2002), where a classifier must deal with well-defined face classes, and an ill-defined non-face class, and handwritten digit recognition (Liu et al., 2002), where non-digit examples are a serious issue.

For simplicity we consider the problem as one in which there is a well-defined *target* class, and a poorly defined *outlier* class. The primary objective is to maintain a high

* Corresponding author. Address: P.O. Box 5031, 2600 GA Delft, Mekelweg 4, 2628CD Delft, The Netherlands. Tel.: +31 (0)15 27 88433; fax: +31 (0)15 27 81843.

E-mail addresses: t.c.w.landgrebe@ewi.tudelft.nl (T.C.W. Landgrebe), d.m.j.tax@ewi.tudelft.nl (D.M.J. Tax), p.paclik@ewi.tudelft.nl (P. Paclík), r.p.w.duin@ewi.tudelft.nl (R.P.W. Duin).

0167-8655/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2005.10.015

classification performance between known classes, and simultaneously to protect the classes of interest from new/unseen classes (or changes in expected conditions, reflected in the change of distribution of these classes). We refer to the latter performance measure as *rejection* performance. Classification performance is defined between a well-defined *target* class ω_t, and some partial knowledge existing for the *outlier* class ω_o. Rejection performance is defined between ω_t and a new (unseen) cluster/class from the *outlier* class ω_r that is not defined precisely in training.

Several strategies have been proposed. The first strategy to cope with this situation was proposed in (Dubuisson and Masson, 1993), called the *distance*-based reject-option. Here a reject-rule was proposed to reject distant objects (with respect to the *target* class) post-classification. This evaluation differs considerably from the second strategy, the *ambiguity* reject-option (defined in (Dubuisson and Masson, 1993)) as proposed in (Chow, 1970). In ambiguity reject, a threshold is included to reject objects occurring in the *overlap* region between two known classes. It is assumed that all classes have been sampled in a representative manner. This is in contrast to this study, in which it is assumed that classes are poorly sampled or not sampled at all.

Classifiers with the reject-rule differ from conventional classifiers in that two thresholds are used to specify the target area, namely a classification threshold θ, and a rejection threshold t_d (we define the target area to be the region in the feature space in which all examples are labelled *target*). A limitation of the distance-reject criterion is that the threshold itself has no direct relationship with the distribution of the known classes, as discussed in (Muzzolini et al., 1978). Thus a modified reject-rule was proposed in (Muzzolini et al., 1978), involving computing the probability of a new object belonging to any of the known classes, based on covariance estimates. The threshold can then be based on a degree of model-fit to the known classes.

In (Landgrebe et al., 2004) we presented a third reject strategy, involving *combinations* of one-class (Tax, 2001) and supervised classifiers. This scheme allowed different models to be specifically designed for the purposes of classification or rejection. It was argued that a model optimised for the sake of classification may differ from that optimised for rejection, and that combining both optimised models can improve the overall combined classification/rejection performance. Experiments showed that this strategy outperforms the other reject-rules in some situations. It was also observed that a relation between the classification and rejection performance exists, and that optimising either performance is at the detriment of the other.

Each of the strategies has a classification and rejection threshold. In both (Dubuisson and Masson, 1993; Muzzolini et al., 1978), it has been shown how the distance-reject-rule can be applied in practise, involving distance- or class-conditional probability-thresholding of new incoming objects. In the case of the ambiguity reject-option, the classifiers can be evaluated and optimised since it is assumed that all classes have been sampled, as shown in (Chow, 1970) for known costs, and applied to imprecise environments in (Ferri and Hernandez-Orallo, 2004; Tortorella, 2004) to name a few. However, in the case of the distance-based reject-option, a challenging problem posed is that the distribution of the unseen class is by definition absent, and thus standard cost-sensitive evaluations and optimisations become ill-defined, lacking a closed Bayesian formalism.

In (Landgrebe et al., 2004), the ill-defined class problem was tackled by deriving strategies that can be used to study the way in which classification and rejection performance interact, based on the assumption that a new unseen class could occur anywhere in feature space. The rationale is that a minimal target area provides, in general, the most robust solution to an unseen class that could occur anywhere in feature space.[1] The methodology involved the artificial generation of the unseen class by assuming a uniformly distributed unseen class. Based on this methodology, it was observed that similar to the ambiguity-reject case, there is interaction between classification and rejection performance.

This paper is concerned with evaluating and optimising classifiers taking into account this interaction between classification and rejection. For this, receiver operating characteristic (ROC) curves will be used. ROC analysis (Metz, 1978), is a tool typically used in the evaluation of two-class classifiers in imprecise environments, plotting detection rate (true-positive rate) against the false positive rate. We extend this analysis to the unseen class problem by including an additional dimension that is related to the general robustness of the classifier to an unseen class. A similar 3-dimensional ROC analysis has been applied elsewhere, such as in (Ferri and Hernandez-Orallo, 2004; Mossman, 1999; Dreisetl et al., 2000), but in these cases this did not involve the ill-defined class problem. Our approach attempts to minimise the volume of the classes of interest in the feature space for robustness against unseen classes. It allows models to be compared (in a relative sense, since an absolute measure cannot be obtained) and provides insight into the choice of a reject threshold, that does not impact too much on classification performance.

In Section 2, an example is studied analytically to investigate the nature of the relation between classification and rejection rates, and the extended ROC analysis is presented. In Section 3, a criterion is proposed for the comparison of the extended ROC's. This criterion is applied to a synthetic 2-dimensional example with three different models. Finally, we discuss how to optimise an operating point (i.e. choose a classification and rejection threshold). Section 4 consists of a number of experiments to demonstrate the methodology in some realistic scenarios. Conclusions are given in Section 5.

[1] Rather than assuming that unseen classes can occur anywhere in feature space, it may be better to consider the nature of each problem, incorporating prior knowledge with respect to natural bounds in this space. To keep the discussion general, for now we assume a uniform, maximum entropy distribution.

2. The relation between classification and rejection performance

First we will develop our notation and illustrate the interaction between the classification and rejection performance by showing an example. In Fig. 1, a synthetic example is presented in which ω_t and ω_o are two Gaussian-distributed classes distributed across domain x. Additionally we assume that a class ω_r is uniformly distributed across x. The class-conditional densities for ω_t, ω_o and ω_r are denoted $p(x|\omega_t)$, $p(x|\omega_o)$, and $p(x|\omega_r)$, respectively, with priors $p(\omega_t)$, $p(\omega_o)$, and $p(\omega_r)$, which are assumed equal here. The unconditional density $p(x)$ can then be written as in Eq. (1). Note that in training we only have access to ω_t and ω_o, and in testing ω_r will also appear.

For the total probability distribution of x therefore holds:

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

For this 1-dimensional data, a classifier is defined which only consists of a single threshold, denoted θ. The position of θ determines the classification performance, and can be specified given a desired true-positive rate (TP_r) or false-positive rate (FP_r). As θ varies, so do the respective TP_r and FP_r, resulting in the ROC between ω_t and ω_o. In a typical discrimination problem (ignoring the reject threshold), we can define the true-positive rate (TP_r) and false positive rate (FP_r) in terms of θ as in Eq. (2).

$$\begin{aligned} TP_r(\theta) &= \int_{-\infty}^{\infty} p(\omega_t)p(x|\omega_t)I(x|\theta)\mathrm{d}x \\ FP_r(\theta) &= \int_{-\infty}^{\infty} p(\omega_o)p(x|\omega_o)I(x|\theta)\mathrm{d}x \end{aligned} \tag{2}$$

The indicator function $I(x|\theta)$ specifies the relevant domain, as defined in Eq. (3).

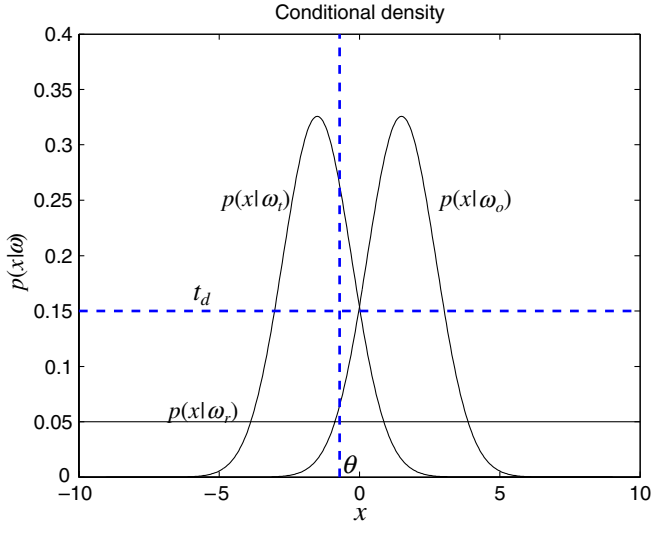

Fig. 1. A synthetic example, illustrating the class-conditional densities for ω_t, ω_o, and ω_r, with a distance-based reject-option classifier. The classification boundary is specified by θ, and the rejection boundary by t_d, $t_d = 0.15$.

$$I(\boldsymbol{x}|\theta) = \begin{cases} 1 & \text{if } p(\omega_t)p(\boldsymbol{x}|\omega_t) - p(\omega_o)p(\boldsymbol{x}|\omega_o) > \theta \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

In a typical discrimination problem, the evaluation criterion used is the mean classification error between ω_t and ω_o and the ability to reject new unseen classes ω_r is often not considered. In ill-defined classification problems, the performance with respect to both the known outliers ω_o and the unknown outliers ω_r is important (Landgrebe et al., 2004). The $FP_r(\theta)$ can therefore be decomposed into two parts:

$$\begin{aligned} FP_r^o(\theta) &= \int_{-\infty}^{\infty} p(x|\omega_o)I(x|\theta)\mathrm{d}x \\ FP_r^r(\theta) &= \int_{-\infty}^{\infty} p(x|\omega_r)I(x|\theta)\mathrm{d}x \end{aligned} \tag{4}$$

Standard classifiers will ignore FP_r^r and only focus on minimising FP_r^o. For these situations the distance-based rejection-option classifier (Dubuisson and Masson, 1993), or the combined sequential one-class and multi-class classifier (Landgrebe et al., 2004) should be used. A reject-option classifier is demonstrated in Fig. 1.

It can be seen that the class-conditional density $p(x|\omega_t)$ is thresholded such that any object x assigned to ω_t will only be accepted if $p(x|\omega_t) > t_d$. Thus for the distance-based reject-option classifier, TP_r (Eq. (2)), FP_r^o and FP_r^r (Eq. (4)), can be written for the general multivariate case as:

$$\begin{aligned} TP_r(\theta, t_d) &= \int_{-\infty}^{\infty} p(\omega_t)p(\boldsymbol{x}|\omega_t)I(\boldsymbol{x}|t_d, \theta)\mathrm{d}\boldsymbol{x} \\ FP_r^o(\theta, t_d) &= \int_{-\infty}^{\infty} p(\omega_o)p(\boldsymbol{x}|\omega_o)I(\boldsymbol{x}|t_d, \theta)\mathrm{d}\boldsymbol{x} \\ FP_r^r(\theta, t_d) &= \int_{-\infty}^{\infty} p(\omega_r)p(\boldsymbol{x}|\omega_r)I(\boldsymbol{x}|t_d, \theta)\mathrm{d}\boldsymbol{x} \end{aligned} \tag{5}$$

where $I(\boldsymbol{x}|t_d, \theta)$ is the indicator function:

$$I(\boldsymbol{x}|t_d, \theta) = \begin{cases} 1 & \text{if } p(\boldsymbol{x}|\omega_t) > t_d \text{ and} \\ & p(\omega_t)p(\boldsymbol{x}|\omega_t) - p(\omega_o)p(\boldsymbol{x}|\omega_o) > \theta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Eq. (5) can be used to study the complete operating characteristic of a classifier. Note that in a real situation, it is unlikely that the class-conditional densities are known. A typical classifier evaluation in these situations involves computing the ROC curve on an independent test set.

We define the complete operating characteristic by all combinations of θ and t_d. The operating characteristic of the example in Fig. 1 is illustrated in Fig. 2. This is similar to standard ROC analysis, in which the TP_r is traded off against the FP_r. Here the FP_r is decomposed into FP_r^o and FP_r^r, resulting in a 3D ROC plot with a 2D ROC *surface*.

In Fig. 2, the operating characteristics are shown for a number of rejection thresholds (t_d), and across all classification thresholds (θ). The left-column plots depict the class-conditional density distributions, with the respective t_d shown in relation to the actual ω_r distribution. In the

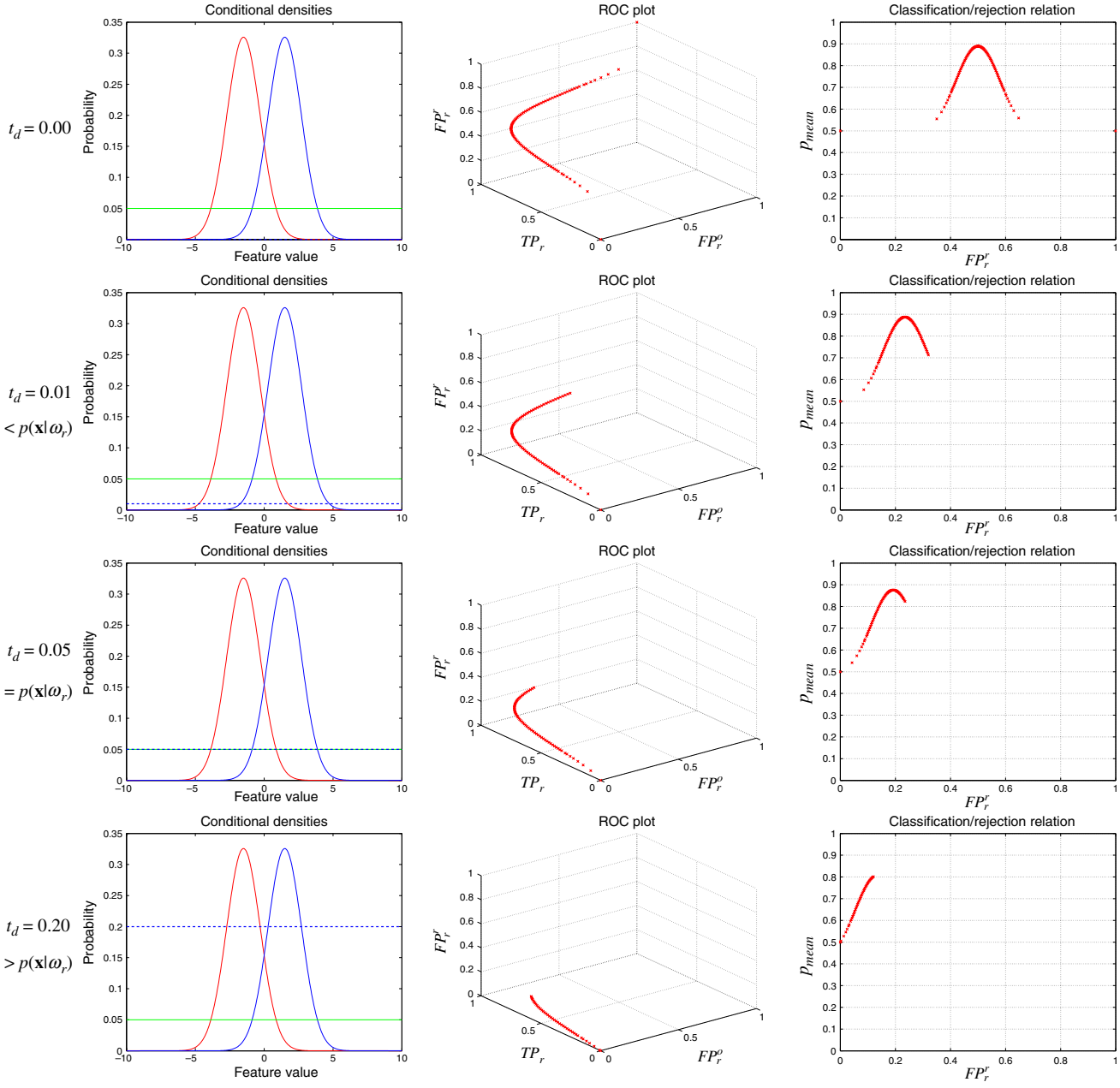

Fig. 2. Fixed t_d, and varying θ.

centre column plots, the ROC curve is presented for all TP_r, FP_r^o, and FP_r^r. By projecting this on the (TP_r, FP_r^o)-plane, the traditional ROC curve is retrieved. The plots in the right column simplify the 3-dimensional surface, plotting the classification performance in terms of mean classification performance for each θ and t_d, against FP_r^r. The mean classification performance is defined as

$$p_{\text{mean}}(\theta, t_d) = 1 - \frac{(1 - \text{TP}_r(\theta, t_d)) + \text{FP}_r^o(\theta, t_d)}{2} \tag{7}$$

It can be observed that as t_d is increased, the FP_r^r progressively decreases: the amount of unseen data that is classified as *target* decreases, and consequently the rejection performance increases. In the top row ($t_d = 0$) there is no rejection protection, the classification performance is maximal but the rejection performance is very poor (high FP_r^r). As soon as some t_d is enforced, the FP_r^r decreases radically, indicating a lower probability of accepting a randomly distributed example from ω_r. Even a very loose boundary in the tails of the ω_t distribution significantly decreases the volume of the decision space. As t_d is increased, the rejection performance increases at some sacrifice of classification performance. This effect is most apparent when t_d is quite high, due to the fact that more *target* examples from the tails of the distribution are excluded.

This synthetic example makes it evident that classification and rejection are interrelated. In Section 3, these extended ROC plots are used to derive a performance

criteria to evaluate different models in these situations, and provide insight into threshold selection.

3. Model selection and optimisation

Now a model selection criterion is formalised that makes use of the full operating characteristics, extending ROC analysis to this problem domain. This will be developed and demonstrated by a synthetic example using three different classifier models.

3.1. Model selection

To select the optimal model, an evaluation criterion for the 3D ROC should be defined. In (Mossman, 1999; Dreisetl et al., 2000; Ferri and Hernandez-Orallo, 2004) the 3-dimensional *volume* under the 2-dimensional surface is related to the overall performance. This is an extension of the AUC (area under the ROC curve) (Bradley, 1997). In a similar manner, we derive an evaluation of the 3-dimensional ROC in our analysis, providing a measure of the classification–rejection performance for the classifier. The ROC in this case plots the $\mathrm{TP_r}$ achieved against $\mathrm{FP_r^o}$ and $\mathrm{FP_r^m}$ (see the example in Fig. 2). Plotting $1-\mathrm{TP_r}$ against $\mathrm{FP_r^o}$ and $\mathrm{FP_r^m}$, it is evident that the volume under this ROC surface should be minimal in the ideal case, implying generally that the classifier achieves low classification error rates, and has high rejection performance (low volume of *target* decision space). Formalising this performance criterion, we derive the VUC (volume under the ROC):

$$\mathrm{VUC}(\theta,t_\mathrm{d}) = 1 - \int\int (1-\mathrm{TP_r}(\theta,t_\mathrm{d}))\mathrm{dFP_r^o}(\theta,t_\mathrm{d})\mathrm{dFP_r^r}(\theta,t_\mathrm{d}) \tag{8}$$

The volume itself is subtracted from 1 to form a performance measure (high scores are favourable). In some cases it may be sensible to also integrate over a restricted range of $\mathrm{FP_r^o}(\theta,t_\mathrm{d})$ (by restricting the range of θ). This effectively analyses a patch of the ROC surface. This is because an AUC integrated over all classification decision thresholds is not always ideal. This occurs in the case in which the ROC surfaces of two different models intersect. In this case model selection is operating-point dependent (Adams and Hand, 1999), and thus the ROC should only be analysed for a range of interest.

3.2. Evaluation on artificial data

In Fig. 3, a scatter-plot of the data is shown. The left plot shows the data available in training (ω_t and ω_o only), and the right plot shows an additional reject-class uniformly distributed, assumed to occur during testing. The dataset consists of 1600 ω_t examples, 800 ω_o examples, and 2400 ω_r examples. The experimental procedure involves using 80% of the data for training, and 20% for testing, with ω_r excluded from training. This is repeated 10 times, following a randomised hold-out procedure. The right plot also depicts the decision boundary of three different classifiers trained on the data, namely a Bayes linear classifier (LDC), a Bayes quadratic classifier (QDC), and a mixture of Gaussians classifier (MOGC), with three clusters per class. The decision boundary is plotted for a single fixed classification and rejection threshold. From the decision boundaries, it is clear that the MOGC fits the data well, as opposed to the LDC and QDC models. These weaker models are typical in real-high dimensional problems where both data and computation time are limited, and thus a model choice may not be obvious.

In Fig. 4 the operating characteristics are shown for the three classifiers. The top row shows the full operating characteristics, and the bottom row depicts the mean classification performance (see Eq. (7)) versus the $\mathrm{FP_r^r}$. These clearly show that the classification–rejection characteristic varies considerably across the different models. The MOGC is able to achieve a higher mean classification performance

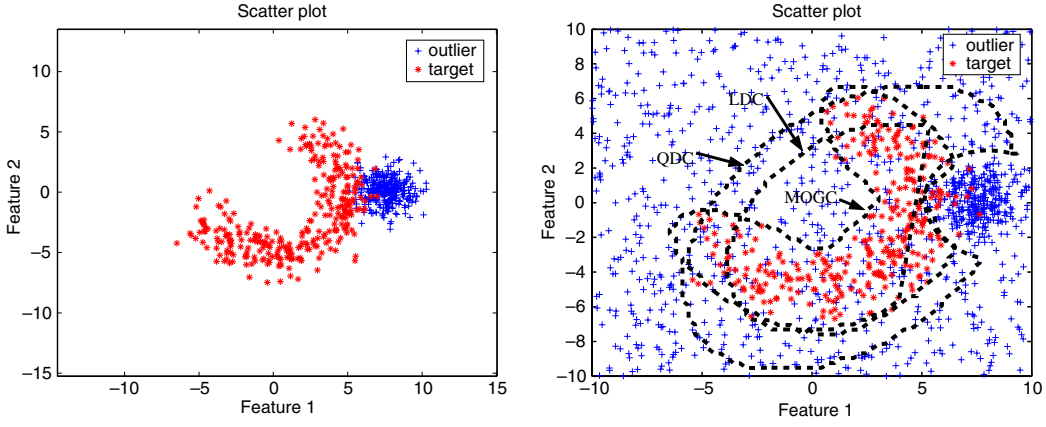

Fig. 3. Scatter-plot for the synthetic example. The left plot shows the data available in training, and the right plot shows a testing scenario, in which a new unseen class exists that should be rejected. The right plot also shows the classifier decision boundaries for three classifiers, at a set operating point. The classifiers are labelled LDC, QDC and MOGC, respectively.

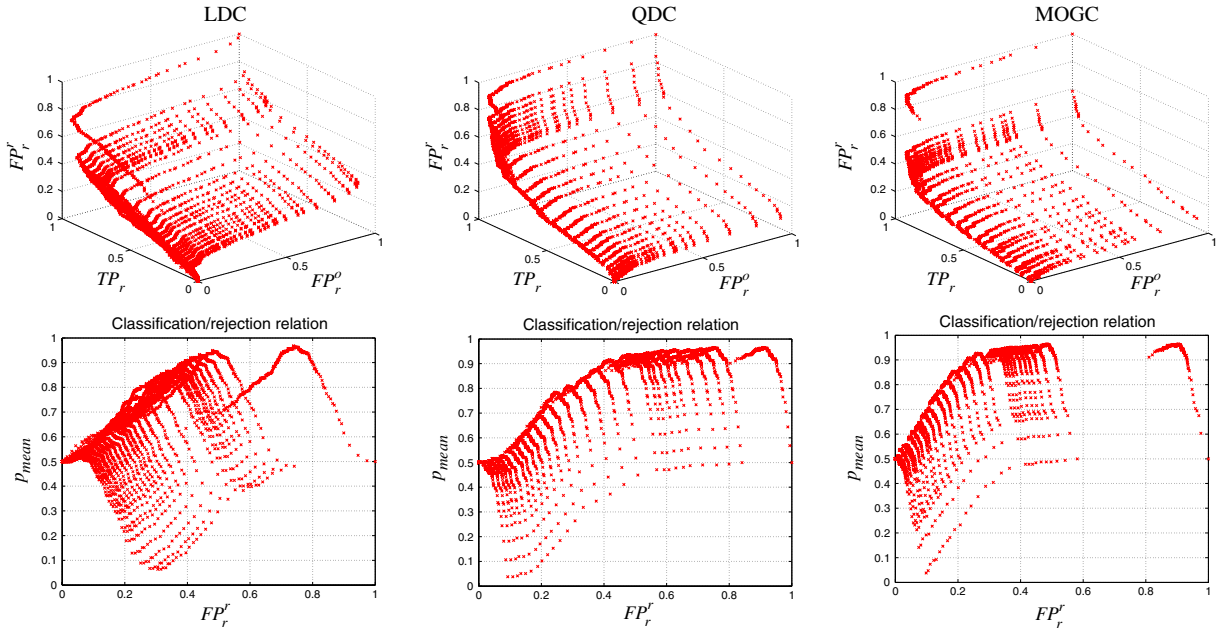

Fig. 4. Operating characteristics for the three classifiers. The left column consists of LDC results, followed by QDC results in the centre column, and MOGC results in the right column. The top row depicts the full ROC, i.e. all possible operating points. The bottom row shows the mean classification performance versus FP_r^r, clearly showing the interplay between the two measures.

for a lower FP_r^r than the LDC and QDC for most regions. For example, for MOGC, it can be seen that the optimal p_{mean} is over 90.00% for a FP_r^r of around 25.00% (based on the assumed ω_r distribution). The LDC, however, only achieves a p_{mean} of around 80.00% for the same degree of rejection performance. Even though the ω_r distribution is unknown, it is apparent in this case that the MOGC classifier can achieve higher rejection robustness than the LDC for the given classification performance. A lower FP_r^r is indicative of a reduced *target* decision space, and the model in question is thus less likely to accept a randomly distributed example from ω_r.

In the lower plots, the top-right characteristics/curves correspond to the case in which $t_d = 0.0$, with curves corresponding to increasing t_d shown to the left of this. A general observation that can be made is that for low values of t_d, the classification performance decreases rapidly for increasing t_d, and the FP_r^r decreases (improving rejection performance). This was expected, showing once again that the classification and rejection performances should be traded off, but also that for large t_d, both classification and rejection performances decrease rapidly. It is also clear that only some (typically low) values of t_d make practical sense, since a large value leads to a very poor recovery of ω_t examples.

In Table 1, the VUC is computed for the synthetic problem for each fold. This performance measure indicates that the MOGC classifier is superior, significantly better than the QDC and LDC model. This is an expected result, since the decision boundary fits the class distribution well, providing both a high classification performance, and a lower decision space volume for high rejection performance. The QDC model is superior to the LDC model. Thus the VUC measure proved to be a useful performance criterion, sensitive to both classification and rejection capabilities of the classifier. Note that in this example, computing the error rate only (on known data) results in competitive performance between QDC and MOGC, however the new VUC criterion and the evaluation methodology showed that MOGC is in fact the better choice since it is better at rejection. Table 1 also shows the results of two other experiments. The ϵ_{rej} measure shows the performance of the three reject-option classifiers for a chosen set of thresholds ($\theta =$ minimum error point, $t_d = 0.05$) based on 10 independent sets of data (drawn from the same distribution, with 1600 ω_t examples, 800 ω_o examples, and 3200 uniform ω_r examples), in which the classifiers were trained on the original set. This performance measure is a simple error rate measure that averages the ω_t and other errors:

$$\epsilon_{rej} = \tfrac{1}{3}((1 - TP_r) + FP_r^o + FP_r^r) \quad (9)$$

Table 1
Comparison of the three classifiers in the case study

	VUC	ϵ_{rej}	ϵ_{norej}
LDC	0.846 ± 0.005	0.197 ± 0.009	0.353 ± 0.007
QDC	0.907 ± 0.003	0.116 ± 0.006	0.420 ± 0.004
MOGC	0.928 ± 0.005	0.087 ± 0.004	0.404 ± 0.003

VUC is computed for each classifier (high scores are favourable), serving as the model selection criterion. The score ϵ_{rej} is the classification error obtained on the independent tests with fixed thresholds for the classifiers with reject-option, and ϵ_{norej} is the error on the same models without reject protection.

These tests confirm the results of the VUC model selection i.e. the MOGC is the most appropriate choice. In order to demonstrate the advantage of using a reject-option, this same test is repeated on the three same models without reject-option, resulting in ϵ_{norej}. In each case it is clear that performance is significantly worse compared to the classifiers with reject-option. Interestingly, the MOGC classifier without reject-option fared worse than the QDC model since it optimised by surrounding the ω_o class, resulting in a very large decision space i.e. poor rejection performance.

3.3. Choosing an operating point

Based on the VUC measure, the most appropriate model (on average) was selected. In the example, the MOGC classifier was found to be superior, and this was demonstrated further by computing error rates on independent data at a specific operating point. Subsequent to the model selection, the next step is to choose classification and rejection thresholds best suited to the problem. In the standard cost-sensitive approach (Provost and Fawcett, 2001), this would involve minimising the overall system loss L, given costs and priors. Assume c_t is the cost of misclassifying a ω_t example, c_o the cost of misclassifying a ω_o example, and c_r the cost of misclassifying an ω_r example (ignoring to which class an error is assigned). The loss can then be computed as:

$$L = \theta p(\omega_t) c_t (1 - \mathrm{TP}_r) + (1 - \theta) p(\omega_o) c_o \mathrm{FP}_r^o + t_d p(\omega_r) \mathrm{FP}_r^r \tag{10}$$

Minimising L involves computing L for all combinations of θ and t_d until the optimal thresholds are found (or geometrically, intersecting an iso-performance *surface* on the ROC surface). However in this problem, since the distribution of ω_r is unknown, the concept of optimality becomes undefined (true FP_r^r values cannot be obtained). The synthetic experiments have, however, shown how to analyse the impact of a varying t_d on classification performance, and decision space volume. Thus a practical step that can be taken in aiding optimisation is to attempt to choose a t_d such that the ω_t decision space is minimised, without sacrificing too much classification performance. Given this premise, we propose that the classification threshold should be optimised first. This is because we have sampled these classes properly, allowing for a cost-sensitive design. Once θ has been chosen, a t_d should be selected that encloses ω_t. The operating characteristics can also be helpful in inspecting the sensitivity over a range of t_d, where a less sensitive choice is to be preferred.

3.4. Summary

In summary, the following steps are involved in generating the full operating characteristics, given a model D, a *target* class ω_t, and an *outlier* class ω_o:

- Assume a distribution for ω_r, for example uniformly distributed around ω_t, and generate data accordingly.
- Train D using ω_t and ω_o.
- Define a range of classification and rejection thresholds θ and t_d. For each threshold, compute the respective TP_r, FP_r^o, and FP_r^r, using independent sets of ω_t and ω_o, and the ω_r data.

The performance criterion VUC can then be used to perform model selection, and the thresholds can be chosen using the operating characteristics.

4. Experiments

In this section a number of real-world examples are conducted, demonstrating practical application of the proposed ROC analysis methodology. Model selection criteria are compared for a number of competing models, and the performance of a classifier with reject-option is compared to the same model, without reject-option. In each case, an independent test set is applied, in which the ω_r class is unseen in training, simulating the effect that an unseen class may have on each classifier. These validation tests are performed to demonstrate the applicability of the VUC measure in these ill-defined classification problems.

The following datasets have been used for these experiments, in which the objective is to detect *target* examples as well as possible, without accepting too many examples from ω_o or ω_r:

(1) *Phoneme*: This dataset is sourced from the ELENA project (ELENA, 2004), in which the task is to distinguish between oral and nasal sounds, based on five coefficients (representing harmonics) of cochlear spectra. In this problem, the "nasal" class is chosen as ω_t. A k-means clustering is performed on the "oral" class, requesting three clusters. The first two clusters are regarded as ω_o (used in training), and the third cluster is treated as ω_r (not used in training).

(2) *Mfeat*: This is a dataset consisting of examples of 10 handwritten digits, originating from Dutch utility maps.[2] In this dataset, Fourier components have been extracted from the original images, resulting in a 76-dimensional representation of each digit. Two hundred examples of each digit are available. In these experiments, digits 2 and 6 are used as the *target* class, digits 4, 5, 6 and 8 as ω_o, and digits 1, 2 and 9 as ω_r (not used in training).

(3) *Satellite*: This dataset consists of 6435 multi-spectral values of a satellite image (Murphy and Aha, 1992), with 36 dimensions (four spectral bands in a nine pixel neighbourhood). Six classes have been identified

[2] Available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/.

to characterise the topography, labelling the dataset accordingly. In the experiments, the fifth class is considered ω_t, classes 1, 4, and 6 are considered known ω_o classes, and classes 2 and 3 are unseen in training (ω_r).

The following procedure is used in each experiment (similar to the case study presented in Section 3):

- An independent test set (ω_t examples only) is extracted from the original data $\boldsymbol{x}$ (containing examples from ω_t and ω_o), resulting in $\boldsymbol{x}_{val1}$, with the remainder called $\boldsymbol{x}_{tr}$. In the experiments, 50% of the *target* examples are extracted.
- Various clusters/subclasses of the ω_o class in $\boldsymbol{x}$ are extracted, resulting in $\boldsymbol{x}_{val2}$, with $\boldsymbol{x}'_{tr}$ remaining. It is important to note that $\boldsymbol{x}_{val2}$ now contains data that will not be used to train the classifier, but will be applied only in the validation test. Since these extracted classes may have a very different distribution to that of $\boldsymbol{x}'_{tr}$, a classifier with reject-option is expected to result in better performance.
- $\boldsymbol{x}_{val1}$ and $\boldsymbol{x}_{val2}$ are combined into a single validation set, $\boldsymbol{x}_{val}$.
- The VUC for each model is estimated following a 10-fold randomised hold-out procedure, utilising the $\boldsymbol{x}'_{tr}$ dataset, and an assumed uniform distribution of ω_r. Since in these ill-defined problems, the ω_r class is absent (by definition), we assume that examples of this class can occur randomly in feature space (a worst-case scenario). Thus an additional class is generated artificially such that these new data surround the ω_t class uniformly. For efficiency reasons in high-dimensional problems, the data are generated in a hyper-sphere rather than a hyper-cube (Tax and Duin, 2001). Additionally the data are generated in a subspace of the *target* class, within a PCA (principal component analysis) subspace, retaining 99.9% of variance. This effectively results in the generation of new objects (the new examples can be reprojected into the original space using the inverse of the PCA mapping). The original data are scaled to unit variance, and the artificial data are then generated within this space with a radius of 1.1 of the covariance of ω_t.
- Following the VUC estimation, the classifier is trained using the full $\boldsymbol{x}'_{tr}$ data. The classification threshold is optimised to the equal error point, and an appropriate reject threshold is chosen for each dataset according to the operating characteristics (obtained in the previous step). The same data are then used to train a second classifier, using the same model, but without reject-option. The validation set $\boldsymbol{x}_{val}$ is then applied to each classifier, and the respective error rates of each classifier are computed, denoted ϵ_{rej} for the reject-option classifier, and ϵ_{norej} for the classifier without reject-option. The mean percentage difference in classification error rates between a classifier with and without reject-option is then computed ($100\text{mean}(\epsilon_{rej} - \epsilon_{norej})$). It is expected that the reject-option classifier should result in improved performance if the independent $\boldsymbol{x}_{val2}$ data distribution varies considerably from the trained distribution.

Thus the objectives of the experiments are to validate the usage of the performance criterion in these realistic scenarios, and show cases in which a good model choice, and reasonable choice of thresholds results in a performance improvement. Note that real data are only used in the validation step, resulting in a realistic set of experiments.

In Table 2, the experimental results are presented. Three groups of results are shown, corresponding to the *Phoneme*, *Mfeat* and *Satellite* experiments, respectively. The first column describes the classifier used, as well as the respective feature extraction procedure, and the rejection threshold used for the validation test. The second column gives the VUC results for each model (with standard deviation shown over 10 folds), in which high scores are favourable. In the third column, the reject-option classifier error rate is compared to the same classifier without reject-option. The error rate percentage of the classifier without reject-option is shown subtracted from the error rate percentage of the classifier with reject-option. A positive result here indicates the percentage improvement (and a negative value indicates the reject-option classifier has performed worse). Note that the test set here consists of both an independent test set from ω_t, and an unseen class/cluster, and thus this measure gives an overall impression of the classification–rejection performance improvement.

Table 2
Summary of experimental results on the *Phoneme*, *Mfeat*, and *Satellite* datasets, comparing models for the VUC performance criterion (high scores are favourable), and comparing the mean absolute percentage difference in classification of error rates between a classifier with and without reject-option on independent test sets

Experiment	VUC	Percentage gain
Phoneme		
QDC $t_d = 0.01$	0.742 ± 0.011	21.67 ± 0.40
PCA-3D QDC $t_d = 0.01$	0.553 ± 0.009	-0.19 ± 0.05
MOGC 1 4 $t_d = 0.01$	0.667 ± 0.036	19.80 ± 5.60
MOGC 1 3 $t_d = 0.01$	0.678 ± 0.022	19.26 ± 6.21
MOGC 1 2 $t_d = 0.01$	0.708 ± 0.021	22.10 ± 4.03
Mfeat		
Fisher NMC $t_d = 0.70$	0.341 ± 0.027	-1.81 ± 0.72
Fisher LDC $t_d = 0.05$	0.503 ± 0.034	28.49 ± 1.20
Fisher QDC $t_d = 0.05$	0.504 ± 0.032	29.22 ± 1.01
Fisher MOGC 2 3 $t_d = 0.05$	0.504 ± 0.027	29.95 ± 0.73
Satellite		
Fisher NMC $t_d = 0.70$	0.374 ± 0.0188	-1.58 ± 0.25
Fisher LDC $t_d = 0.10$	0.612 ± 0.009	10.27 ± 0.26
PCA-4D LDC $t_d = 0.10$	0.489 ± 0.015	12.10 ± 0.34

The standard deviations are also shown over the 10-fold experiments. PCA is a principal component analysis representation, followed by the number of retained components, and Fisher is a Fisher-projection to 1-dimension. NMC is a nearest mean classifier, LDC is a Bayes linear classifier, QDC is a Bayes quadratic classifier, and MOGC is a mixture of Gaussians classifier followed by the numbers of mixtures used per class.

In the *Phoneme* results it can be seen that four of the five models result in a large performance improvement in the independent tests. The PCA-3D QDC model does not result in improved performance, which is also supported by a lower VUC. The QDC model performs much better, indicating that dimensionality reduction is not appropriate here (the original data only have five features). This argument is also strengthened by the higher VUC measure. In the three mixture-of-Gaussian tests, it seems apparent that a lower number of mixtures results in better performance. The *Mfeat* experiments consider four different models following a Fisher dimensionality reduction. In the case of the NMC (nearest mean classifier), it should be mentioned that small values of t_d results in no rejection at all, and only very large values result in rejection protection. However, at this point, the impact on classification performance is severe, and thus the model is unsuitable for this task. This is also suggested by a lower VUC score. The three other models result in comparable performance. In the *Satellite* experiments, the NMC results in a similar situation to that seen in the *Mfeat* case. Both the Fisher LDC, and the PCA-4D LDC models result in a similar performance in the validation test. However it can be seen that the Fisher LDC has a significantly larger VUC. This discrepancy may be due to the fact that the VUC averages performance over all possible operating points, and is thus not locally sensitive. In this case it may make more sense to integrate over a smaller range (given that some prior knowledge about the problem exists). In these two cases, a more local VUC was performed, denoted VUC_2, in which the integration was applied over the full range of θ, and over a restricted t_d range, $0.0 < t_d \leqslant 0.2$. These experiments resulted in $VUC_2(\text{Fisher, LDC}) = 0.591 \pm 0.0111$, and $VUC_2(\text{PCA4D, LDC}) = 0.554 \pm 0.012$. It is clear that the performance measures are now more similar, which is an expected result.[3]

On the whole, the experiments show that the derived VUC measure is useful in identifying more appropriate models. An important observation that can be made is that a reject-option classifier does not always result in adequate/beneficial protection against unseen classes. The ROC analysis approach presented here helps to identify these cases. Another point that should be raised is that an adequate rejection threshold varies according to the problem and model. The implication is that a reject-threshold setting that does not consider the operating characteristics may have little or a detrimental effect on performance. This is an important consideration that is highlighted and dealt with in this paper.

5. Conclusion

Classifiers designed to protect a well-defined *target* class from ill-defined conditions, such as new unseen classes, are defined by two decision thresholds, namely a classification and rejection threshold. The classification threshold is designed to provide an optimal trade-off between known classes, and the rejection threshold protects the *target* class against changes in conditions e.g. new unseen classes.

In this paper, we discussed the fact that classification and rejection performances are not independent, but that there is an interplay between them. The consequence of the interplay is that independently optimising classification performance may be at the expense of rejection performance, and the opposite also holds. Even though this interaction is expected, the fact that the unseen class is absent makes it difficult to devise a model selection and optimisation strategy that results in a classifier with both good classification and rejection performance. This paper tackled this problem by measuring how well the classifier protects the *target* class from a uniformly distributed ill-defined class, effectively resulting in a measure proportional to the volume occupied by the *target* class decision space. This measure aids in choosing and optimising a classifier that reduces the risk of misclassifying an unseen class (without too much loss of classification performance) since we can now inspect both the classification performance, and volume of the decision space.

The investigation of this problem involved the extension of classical 2-dimensional ROC analysis by including the errors associated with the unseen class as an additional dimension of the ROC. This results in a 3-dimensional ROC surface, allowing the classification–rejection dynamics to be investigated. This was demonstrated via a simple analytic example, and subsequently used to devise a performance measure involving integrating the volume of the ROC plot, resulting in the volume under the ROC (VUC), which is analogous to the area under the ROC measure. Experiments were performed which showed the effectiveness of this measure in selecting the most appropriate model for the problem. Real experiments validated the measure by including a test involving real unseen classes/clusters, in which there was a consistency between good VUC scores and classifier performance with respect to the unseen data. The experiments made it clear that careful attention should be paid in the choice of the reject threshold, showing how the proposed ROC analysis can lead to a solution involving minimal impact on classification performance, but large impact on reducing the risk of accepting unseen class examples.

Acknowledgements

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. A special mention is given to the anonymous reviewers who helped clarify some aspects of this work.

[3] Note that since a limited range is used, this performance measure is not bound between 0 and 1, but is instead bound by the volume over which the integration is performed in the unit cube.

References

Adams, N., Hand, D., 1999. Comparing classifiers when misallocation costs are uncertain. Pattern Recognition 32 (7), 1139–1147.

Bradley, A., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 30 (7), 1145–1159.

Chow, C., 1970. On optimum error and reject tradeoff. IEEE Trans. Inform. Theory It-16 (1), 41–46.

Dreisetl, S., Ohno-Machado, S., Binder, M., 2000. Comparing trichotomous tests by three-way ROC analysis. Med. Dec. Making 20 (3), 323–331.

Dubuisson, B., Masson, M., 1993. A statistical decision rule with incomplete knowledge about classes. Pattern Recognition 26 (1), 155–165.

ELENA, 2004. European ESPRIT 5516 project, phoneme dataset. Available from: <ftp://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases>.

Ferri, C., Hernandez-Orallo, J., 2004. Cautious classifiers. In: First Workshop on ROC Analysis in AI (ROCAI-2004), August, pp. 27–36.

Landgrebe, T., Tax, D., Paclík, P., Duin, R., Andrew, C., 2004. A combining strategy for ill-defined problems. In: Fifteenth Ann. Sympos. of the Pattern Recognition Association of South Africa, November, pp. 57–62.

Liu, C., Sako, H., Fujisawa, H., 2002. Performance evaluation of pattern classifiers for handwritten character recognition. Internat. J. Document Anal. Recognition, 191–204.

Metz, C., 1978. Basic principles of ROC analysis. Sem. Nucl. Med. 3 (4).

Mossman, D., 1999. Three-way roc's. Med. Dec. Making 19, 78–89.

Murphy, P., Aha, D., 1992. UCI repository of machine learning databases. University of California, Department of Information and Computer Science. Available from: <ftp://ftp.ics.uci.edu/pub/machine-learning-databases>.

Muzzolini, R., Yang, Y., Pierson, R., 1978. Classifier design with incomplete knowledge. Pattern Recognition 31 (4), 345–369.

Paclík, P., 2004. Building Road Sign Classifiers. PhD thesis, CTU Prague, Czech Republic, December.

Pham, T.V., Worring, M., Smeulders, A.W.M., 2002. Face detection by aggregated Bayesian network classifiers. Pattern Recognition Lett. 23 (4), 451–461.

Provost, F., Fawcett, T., 2001. Robust classification for imprecise environments. Machine Learn. 42, 203–231.

Tax, D., 2001. One-class Classification. PhD thesis, Delft University of Technology, The Netherlands, June.

Tax, D., Duin, R., 2001. Uniform object generation for optimizing one-class classifiers. J. Machine Learn. Res., 155–173.

Tortorella, F., 2004. Reducing the classification cost of support vector classifiers through an ROC-based reject rule. Pattern Anal. Appl. 7, 128–143.

Ypma, A., Tax, D., Duin, R., 1999. Robust machine fault detection with independent component analysis and support vector data description. In: Proc. 1999 IEEE Workshop on Neural Networks for Signal Processing, Madison, pp. 67–76.