The Data Replication Method for the Classification with Reject Option

Ricardo Sousa [a,*], Jaime S. Cardoso [b]

[a] *Instituto de Telecomunicações, Faculdade de Ciências, Universidade do Porto*

[b] *INESC TEC (formely INESC Porto), Faculdade de Engenharia, Universidade do Porto*

Classification is one of the most important tasks of machine learning. Although the most well studied model is the two-class problem, in many scenarios there is the opportunity to label critical items for manual revision, instead of trying to automatically classify every item.

In this paper we tailor a paradigm initially proposed for the classification of ordinal data to address the classification problem with reject option. The technique reduces the problem of classifying with reject option to the standard two-class problem. The introduced method is then mapped into support vector machines and neural networks. Finally, the framework is extended to multiclass ordinal data with reject option. An experimental study with synthetic and real datasets verifies the usefulness of the proposed approach.

Keywords: Reject Option, Support Vector Machines, Neural Networks, Supervised Learning, Classification

1. Introduction

Decision support systems are becoming ubiquitous in many human activities, most notably in finance and medicine. Automatic models are being developed to imitate, as closely as possible, the usual human decision. Within this context, classification is one of the most representative predictive learning tasks. Classification predicts a categorical value for a specific data item. The most well studied scenario is when the class to be predicted can assume only two values—binary setting. The classifier is developed to partition the feature space in two regions, discriminating between the two classes.

In credit scoring modeling, models are developed to determine how likely applicants are to default with their repayments. Previous repayment history is used to determine whether a customer should be classified into a 'good' or a 'bad' category [29]. Prediction of insurance companies' insolvency has arisen as an important problem in the field of financial research due to the necessity of protecting the general public whilst minimizing the costs associated to this problem [29]. In medicine, the last decades have witnessed the development of advanced diagnostic systems as alternative, complementary or a first opinion in many applications [3]. These are just some applications that continue to challenge researchers in the deployment of fully automated decision support systems.

One of the problems with classifying complex items is that many items from distinct classes have similar structures in the feature space, resulting in a setting with overlapping classes. The automation of decisions in these regions leads invariably to many wrong predictions. On the other hand, and although items in the historical data are labeled *only* as 'good' or 'bad', the deployment of a decision support system in many environments has the opportunity to label critical items for manual revision, instead of trying to automatically classify each and every item. The system automates only those decisions which can be reliably predicted, labeling the critical ones for a human expert to analyze. Therefore, the development of classifiers with a third output class, the reject class, in-between the good and bad classes, is attractive.

In a preliminary study [27], it was proposed a new learning methodology, which is extended and explored in various directions in this paper. Here, we first detail the presentation of the method, in-

*Corresponding author: Ricardo Sousa
Instituto de Telecomunicações, Faculdade de Ciências, Universidade do Porto
Rua Campo Alegre 1021/1055, 4169-007 Porto, Portugal
E-mail: rsousa@dcc.fc.up.pt

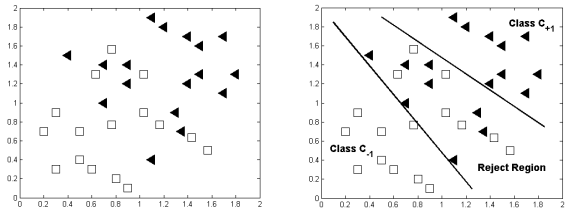
AI Communications
ISSN 0921-7126, IOS Press.

troducing the mapping to support vector machines (SVMs) and neural networks (NNs). Second, we generalize the framework from binary classification problems to multiclass ordinal data. Finally, the experimental work reported at the end of this communication is expanded, including a comparison over more datasets and with conventional and state of the art methods. A principled approach for learning critical regions on complex data is motivated and presented in Section 2, followed by a review of the most relevant works addressing the reject option problem. In Section 3 the fundamental concept of the reject option paradigm is revised and the proposed model of this paper is described in Section 4. An extension of standard procedures for the reject option problem in the ordinal context is presented in Section 5 and the implementation considerations of the methods discussed in this paper is presented in Section 6. Performance assessment is conducted in Section 7. Finally, conclusions are drawn in Section 8.

2. Problem Statement and Standard Solutions

Predictive modeling tries to find good models for predicting the values of one or more variables in a dataset from values of other variables. Our target can assume only two values, represented by 'good' and 'bad' classes. When in possession of a "complex" dataset, a simple separator is bound to misclassify some points. Two types of errors are possible, 'false positives' and 'false negatives'. The construction of a model can be conducted to optimize some adopted measure of business performance, be it profit, loss, volume of acquisitions, market share, etc, by giving appropriate weights to the two types of errors. When the weights of the two types of errors are heavily asymmetric, the boundary between the two classes will be pushed near values where the most costly error seldom happens.

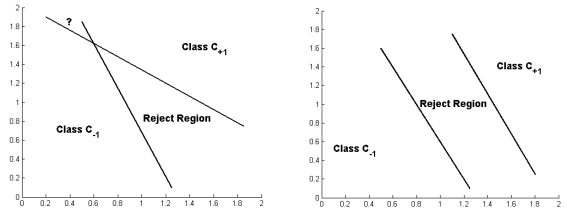
This fact suggests a simple procedure to construct a three-class output classifier: training a first binary classifier with a set of weights heavily penalizing the false negative errors, we expect that when this classifier predicts an item as negative, it will be truly negative. Likewise, training a second binary classifier with a set of weights heavily penalizing the false positive errors, we expect that when this classifier predicts an item as positive, it will be truly positive. When a item is predicted as positive by the first classifier and negative by the second, it will be labeled for review. This setting is illustrated in Fig. 1.



(a) Overlapping regions. (b) Typical separator lines, obtained with two independent binary classifiers.

Fig. 1. Illustrative setting with overlapping classes.

A problem arises when an item is predicted as negative by the first classifier and positive by the second classifier as in Fig. 2(a). That can happen because the two separator lines intersect each other, generating therefore regions with a *non-logical decision* (regions where individual classifiers are inconsistent, individually deciding for different classes). A convenient workaround is then to avoid this problematic state by imposing that the two boundaries of the classifiers do not intersect, Fig. 2(b).



(a) Intersecting separating lines. (b) Non-intersecting separating lines.

Fig. 2. Potential discriminative boundaries. The advantage of the approach depicted in Fig. 2(b) on an ordinal setting has already been stated in [8].

Before delving into the proposed method, it is worth discussing the simple solution of using a single classifier. If more than just discriminating between the two classes, the model to use yields the posterior probability for each target class, then two cutoffs can be defined on this value. All items with predicted probability of belonging to class $\mathcal{C}_{-1}$ less

than a low threshold are labeled as $\mathcal{C}_{+1}$, items with predicted probability of belonging to class $\mathcal{C}_{-1}$ higher than a high threshold are labeled as $\mathcal{C}_{-1}$, items with predicted probability of belonging to class $\mathcal{C}_{-1}$ in-between the low and high threshold are labeled for review. Two issues can be identified with this approach. First, we need to estimate the probability of each class, which is by itself a problem harder than the problem of discriminating classes. Second, the estimation of the two cutoffs is not straightforward nor can be easily fitted into standard frameworks.

The design of classifiers with reject option can be systematized in three different approaches:

- the design of two, *independent*, classifiers. A first classifier is trained to output $\mathcal{C}_{-1}$ only when the probability of $\mathcal{C}_{-1}$ is high and a second classifier trained to output $\mathcal{C}_{+1}$ only when the probability of $\mathcal{C}_{+1}$ is high. In other words, train a first classifier with a set of weights heavily penalizing the false negative errors in order to obtain truly negative predictions; and, train a second classifier with a set of weights heavily penalizing the false positive errors in order to obtain truly positive predictions. The simplicity of this strategy has the weakness of producing intersecting boundaries, leading to regions with a *non-logical decision* as aforementioned.
- the design of a single, standard binary classifier. This approach already provides non-intersecting boundaries. If the classifier provides some approximation to the a posteriori class probabilities, then a pattern is rejected if the maximum of the two posterior probabilities is lower than a given threshold. If the classifier does not provide probabilistic outputs, then a rejection threshold targeted to the particular classifier is used. For example, the rejection techniques proposed with support vector machines consist in rejecting patterns those distance from the optimal separating hyperplane is lower than a predefined threshold. The rejection region is determined *after* the training of the classifier, by defining appropriate threshold values on the output of the classifier [9,17,18].
- the design of a single classifier with embedded reject option. This approach has consisted in the design of algorithms specifically adapted for the reject option problem. Although the option has the advantage of determining the reject region during the training phase of the classifier, it requires the implementation of very specific algorithms, usually appropriate for a single class of classifiers, like support vector machines [16,5].

The method to be proposed belongs to the type of classifiers with embedded reject option. The main advantage of the methods in this category is simultaneously their main limitation: since the cost matrix is embedded during the design, they are optimal (in some sense) to that 'business' criterion. By integrating the business performance in the model construction we expect to attain an 'optimal' classifier, tuned for the business criterion. However, a change in the business rules implies that the model needs to be re-designed. Nonetheless, that may be easily accomplished. Typically the cost matrix evolves slowly. So, instead of retraining, a simpler update is usually sufficient. For instance, for a neural network, that may require to run the training process a few iterations, starting from the previously optimized network, instead of starting randomly. Since the costs are similar, the convergence should be very fast.

In the next subsection we overview the current state of the art related to the reject option problem.

2.1. State of the Art

In one of the first works to analyze the tradeoffs between erring and rejecting, Chow in [9] derived a general error and reject tradeoff relation for the Bayes optimum recognition system. This derivation assumed a complete knowledge of the a priori probability distribution of the classes and the posterior probabilities which, in real problems, are usually unknown. Fumera [17,18] shows that Chow's rule does not perform well if a significant error in probability estimation is present, proposing the use of multiple reject thresholds related to the data classes.

The incorporation of reject option opens new fields of applications for a learning method. For instance, application to Multiple Instance Learning (MIL) for image categorization as presented in [33], the improvement of reliability in banknote neuro-classifier [1] through the use of PCAs and a Learning Vector Quantization (LVQ), among others.

The introduction of the reject option in a classifier also demands the introduction of new evaluation measures. In [13] new measures are developed to find a relation between the reduction of the number of misclassified instances and the reduction of the number of unclassified instances. Despite the results obtained and presented, they claim that their measures can not be statistically interpreted and henceforth no formal interpretation can be taken [13]. Following this idea, in [12] the concept of delegating classifiers in a systematic way is developed. These type of methods follow the concept of divide-to-conquer [19,13,12], where a more generic classifier abstains on a part of the examples and delegates them to a second, more specific, classifier. However, such approaches could potentially delegate only a small number of instances to the second classifier which will lead to overfitting [12].

Based on the ROC curve principle, as in [13], a cost-sensitive reject rule for SVM classifiers is introduced in [30]. Other strategies are taken in [31,26] where a reject rule based on the ROC curve is specially designed for binary classifiers.

In [22] the authors explored the idea of combining one-class learning models with supervised learning. They further evaluated their strategy concerning the incorporation of a reject option on classification tasks through ROC analysis [21]. The measures explored in [21] aid in choosing and optimizing a classifier that reduces the risk of misclassifying an unseen class (outlier). Another system to identify outliers, in contrast with those proposed in [22,21], is presented in [28]. The authors propose a heuristic which combines any type of one-class models for solving multi-class classification problem with outlier rejection. This is achieved through the use of two models: density and distance based class models. In this scheme, PCA is used to avoid the dimensionality problem. Instead of rejecting outlier instances, in [23] it is suggested a new rejection scheme. Their technique encompasses the rejection of instances from one class determined as outlier and the assignment of instances to the remaining classes.

Other approaches can be taken. If the probability density functions of classes are known, pattern recognition is a problem of statistical hypothesis testing [15]. Keeping in mind the minimization of the empirical risk principle, in [6] it is proposed a kernel learning method. This technique consists in a likelihood ratio based classifier where a Parzen window estimator is used to estimate the probability densities. In [5], the authors follow the statistical hypothesis testing rationale a little further through the use of the Neyman-Pearson (NP) criterion. NP does not introduce any new decision theory since it relies on the likelihood test as Bayes theory [15]. However, this criterion has a more natural way to specify a constraint on the false alarm (type I error) probability than to assign costs to the different kinds of errors. Based on this, a reject option method based on the Neyman-Pearson criterion is presented as an extension of the Chow's rule.

Although several learning methods exist addressing the reject option, only a few tackle the assessment of the sensibility. Devarakota et al. [10] present a generic approach where, through the quantification of uncertainty of a decision made by a statistical learning scheme, the method computes a confidence interval which can afterward be used on several learning techniques.

Despite the myriad of techniques that handle the incorporation of a reject option in their approaches, many of them do not fully account the pioneer work of [9]. Also, the principle issue usually used in pattern recognition, which is the minimization of the empirical risk, is feebly explored on the reject option case. Moreover, a major difficulty with these approaches is that the resulting formulations are no longer standard optimization procedures and cannot be solved efficiently, lacking some appealing features like convexity and sparsity. In this line, [2,32] consider a convex surrogate of the generalized loss function to efficiently solve the resulting problem under SVMs and of the convex loss functions. As an extension of this, in [20] it is proposed a double hinge function and a probabilistic viewpoint of the SVM fitting. Without changing the loss function, in [16] it is proposed a modified support vector machines (SVMs). In [27] a new embedded reject option learning scheme is presented and in [24] it is applied to the diagnostic of pathology on the vertebral column.

In this work we detail a solution that: a) uses standard binary classifiers; b) produces non-intersecting boundaries; c) determines the reject region during the training phase. The proposed solution is based on the extension of a technique developed for ordinal data.

3. The Optimum Reject Rule

The pioneer work of [9] about the optimum reject rule provided the foundations towards the reject option problem development, as already mentioned in the previous section. For completeness, we start by reviewing this work.

On statistical decision theory, one decides $\mathcal{C}_j$ for a given pattern $\mathbf{x}$ if,

$$\pi_j p(\mathbf{x}|\mathcal{C}_j) \geq \pi_i p(\mathbf{x}|\mathcal{C}_i) \qquad \forall i = 1, \ldots, K$$

where π_i is the a priori probability of class $\mathcal{C}_i$, and K the number of classes [15,11].

In what concerns to rejection, a decision is holdup if the maximum a posteriori probability is less than a given threshold, $1 - t$. In other words:

$$\max_i(\pi_i p(\mathbf{x}|\mathcal{C}_i)) < (1-t)\sum_{i=1}^{K}(\pi_i p(\mathbf{x}|\mathcal{C}_i)),$$

with $0 \leq t \leq 1$. Assuming uniform cost function within each class of decisions, i.e., no distinction is made among errors, among the rejects and among the correct recognition, the rejection threshold is related with the costs by

$$t = \frac{w_r - w_c}{w_e - w_c}$$

where w_r, w_e and w_c, are the costs for rejecting, error and correct classification, respectively [9,16].

Defining the probability of error, or error rate, as $E(t)$ and the probability of reject or reject rate as $R(t)$ and assuming $w_e = 1$ and $w_c = 0$, one obtains [9]

$$risk(t) = E(t) + tR(t) \tag{1}$$

We will return to this matter later in this paper.

4. An Ordinal Data Approach for Detecting Reject Regions

Having motivated the development of classifiers with a third output class, the reject class, inbetween the good and bad classes, this particular class structure will be explored using concepts from ordinal data classification. In statistical pattern recognition, it is usually assumed that a training set of labeled patterns is available where each pair $\{\mathbf{x}_i, y_i\} \in \Re^d \times \mathcal{Y}$ has been generated independently from an unknown distribution. The goal is to induce a classifier, i.e., a function from patterns to labels $f : \Re^d \to \mathcal{Y}$. On the ordinal case the output space exhibits a natural order, for instance, Excellent$\succ$Good$\succ$Fair$\succ$Poor, and formally defined as $\mathcal{Y} = \{y_1, \cdots, y_K\}$, where $y_1 \succ \cdots \succ y_K$ and $\succ$ is a linear order relation in $\mathcal{Y}$.

4.1. The Data Replication Method for Ordinal Data

The data replication method for ordinal data can be framed under the single binary classifier (SBC) reduction, an approach for solving multiclass problems via binary classification relying on a single, standard binary classifier.

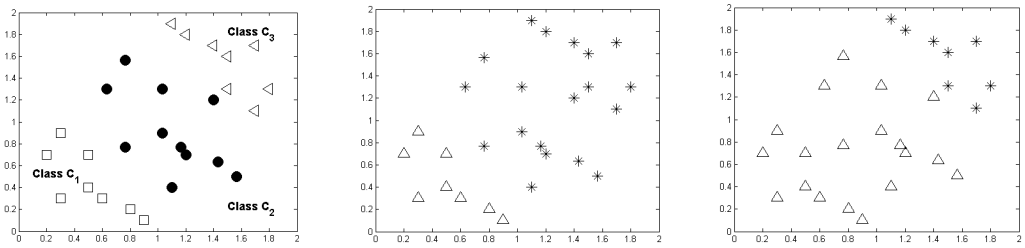
To introduce the data replication method, assume that examples in a classification problem come from one of K ordered classes, labeled from $\mathcal{C}_1$ to $\mathcal{C}_K$, corresponding to their natural order. Consider the training set $\{\mathbf{x}_i^{(k)}\}$, where $k = 1, \ldots, K$ denotes the class number, $i = 1, \ldots, \ell_k$ is the index within each class, and $\mathbf{x}_i^{(k)} \in \mathbb{R}^d$, with p the dimension of the feature space. Let $\ell = \sum_{k=1}^{K} \ell_k$ be the total number of training examples.

Let us consider a very simplified toy example with just three classes, as depicted in Fig. 3(a). Here, the task is to find two parallel hyperplanes, the first one discriminating class $\mathcal{C}_1$ against classes $\{\mathcal{C}_2, \mathcal{C}_3\}$ and the second hyperplane discriminating classes $\{\mathcal{C}_1, \mathcal{C}_2\}$ against class $\mathcal{C}_3$. These hyperplanes will correspond to the solution of two binary classification problems but with the additional constraint of parallelism—see Fig. 3. The data replication method suggests solving both problems simultaneously in an augmented feature space [8].

In the toy example, using a transformation from the $\mathbb{R}^2$ initial feature-space to a $\mathbb{R}^3$ feature space, replicate each original point, according to the rule (see Fig. 4(a)):

$$\mathbf{x} \in \mathbb{R}^2 \begin{array}{l} \nearrow [\begin{smallmatrix}\mathbf{x}\\ h\end{smallmatrix}] \in \mathbb{R}^3 \\ \searrow [\begin{smallmatrix}\mathbf{x}\\ 0\end{smallmatrix}] \in \mathbb{R}^3 \end{array}, \text{ where } h = \text{const} \in \mathbb{R}^+$$

Observe that any two points created from the same original point differ only in the extension feature. Define now a binary training set in the new (higher

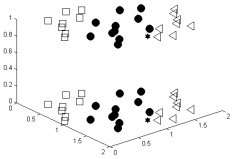


(a) Original dataset in $\mathbb{R}^2$, $K = 3$.

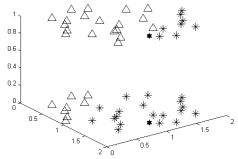
(b) Binary problem $\mathcal{C}_1$ against classes $\{\mathcal{C}_2, \mathcal{C}_3\}$.

(c) Binary problem $\{\mathcal{C}_1, \mathcal{C}_2\}$ against class $\mathcal{C}_3$.

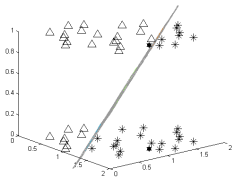
Fig. 3. Binary problems to be solved simultaneously with the data replication method.



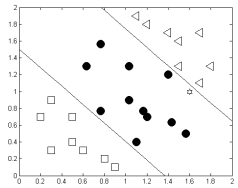
(a) Dataset in $\mathbb{R}^3$, with samples replicated ($h = 1$).



(b) Transformation into a binary classification problem.



(c) Linear solution to the binary problem.



(d) Linear solution in the original dataset.

Fig. 4. Data replication model in a toy example (from [8]).

dimensional) space according to (see Fig. 4(b)):

$$\begin{aligned} &\begin{bmatrix} \mathbf{x}_i^{(1)} \\ 0 \end{bmatrix} \in \overline{\mathcal{C}}_1, \ \begin{bmatrix} \mathbf{x}_i^{(2)} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{x}_i^{(3)} \\ 0 \end{bmatrix} \in \overline{\mathcal{C}}_2 \\ &\begin{bmatrix} \mathbf{x}_i^{(1)} \\ h \end{bmatrix}, \begin{bmatrix} \mathbf{x}_i^{(2)} \\ h \end{bmatrix} \in \overline{\mathcal{C}}_1, \ \begin{bmatrix} \mathbf{x}_i^{(3)} \\ h \end{bmatrix} \in \overline{\mathcal{C}}_2 \end{aligned} \tag{2}$$

In this step we are defining the two binary problems as a single binary problem in the augmented feature space. A linear two-class classifier can now be applied on the extended dataset, yielding a hyperplane separating the two classes, see Fig. 4(c). The intersection of this hyperplane with each of the subspace replicas can be used to derive the boundaries in the original dataset, as illustrated in Fig. 4(d).

To predict the class of an unseen example, classify both replicas of the example in the extended dataset with the binary classifier. From the sequence of binary labels one can infer the predicted label on the original ordinal classes

$$\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_1 \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_2 \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_2 \Longrightarrow \mathcal{C}_3$$

Note that only three sequences are possible [8]. The generalization for any problem in $\mathbb{R}^d$, with K ordinal classes and nonlinear boundaries can be found in [8].

Summing up, $(K-1)$ replicas in a $\mathbb{R}^{p+K-2}$ dimensional space are used to train a binary classifier. The target class of an unseen example can be obtained by adding one to the number of $\mathcal{C}_2$ labels in the sequence of binary labels resulting from the classification of the $(K-1)$ replicas of the example.

4.2. The Data Replication Method for Detecting Reject Regions

The scenario of designing a classifier with reject option shares many characteristics with the classification of ordinal data. It is also reasonable to assume for the reject option scenario that the three output classes are naturally ordered as $\mathcal{C}_1, \mathcal{C}_{reject}, \mathcal{C}_2$. As the intersection point of the two boundaries would indicate an example with the three classes equally probable—one would be equally uncertain between assigning $\mathcal{C}_1$ or $\mathcal{C}_{reject}$ and between assigning $\mathcal{C}_{reject}$ or $\mathcal{C}_2$—it is plausible to adopt a strategy imposing non-intersecting boundaries. In fact, as reviewed in Section 2, methods have been proposed with exactly such assumption. In the scenario of designing a classifier with reject option, we are interested on finding two boundaries: a boundary discriminating $\mathcal{C}_1$ from $\{\mathcal{C}_{reject}, \mathcal{C}_2\}$ and a boundary discriminating $\{\mathcal{C}_1, \mathcal{C}_{reject}\}$ from $\mathcal{C}_2$.

We proceed exactly as in the data replication method for ordinal data. We start by transforming the data from the initial space to an extended

space, replicating the data, according to the rule (see Fig. 5(a) and Fig. 5(b)):

$$\mathbf{x} \in \mathbb{R}^d \nearrow \begin{bmatrix} \mathbf{x} \\ h \end{bmatrix} \in \mathbb{R}^{d+1} \quad \searrow \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \in \mathbb{R}^{d+1}, \text{ where } h = \text{const} \in \mathbb{R}^+$$

If we design a binary classifier on the extended training data, without further considerations, one would obtain the same classification boundary in both data replicas. Therefore, we modify the misclassification cost of the observations according to the data replica they belong to. In the first replica (the extension feature assumes the value zero), we will discriminate $\mathcal{C}_1$ from $\{\mathcal{C}_{reject}, \mathcal{C}_2\}$; therefore we give higher costs to observations belonging to class $\mathcal{C}_2$ than to observations belonging to class $\mathcal{C}_1$. This will bias the boundary towards the minimization of errors in $\mathcal{C}_2$. In the second replica (the extension feature assumes the value h), we will discriminate $\{\mathcal{C}_1, \mathcal{C}_{reject}\}$ from $\mathcal{C}_2$; therefore we give higher costs to observations belonging to class $\mathcal{C}_1$ than to observations belonging to class $\mathcal{C}_2$. This will bias the boundary towards the minimization of errors in $\mathcal{C}_1$. In Fig. 5(c) this procedure is illustrated by filling the marks of the observations with higher costs. Table 1 summarizes this procedure.

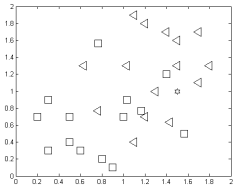
Replica #	points from $\mathcal{C}_1$	points from $\mathcal{C}_2$
1	$-1; C_\ell$	$+1; C_h$
2	$-1; C_h$	$+1; C_\ell$

Table 1

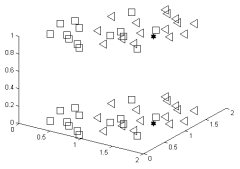
Labels and costs (C_ℓ and C_h represent a low and a high cost value, respectively) for points in different replicas in the extended dataset.

A two-class classifier can now be applied on the extended dataset, yielding a boundary separating the two classes, see Fig. 5(d). The intersection of this boundary with each of the subspace replicas can be used to derive the boundaries in the original dataset, as illustrated in Fig. 5(e).

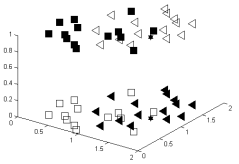
Summing up, with a proper choice of costs, the data replication method can be used to learn a reject region, defined by two non-intersecting boundaries. Note that the reject region is optimized during training and not heuristically defined afterward. Nonlinear (and non-intersecting) boundaries are treated exactly as the ordinal data scenario. Likewise, prediction follows the same rationale.



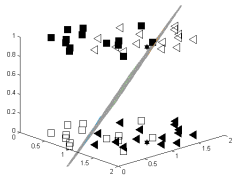
(a) Original binary dataset in $\mathbb{R}^2$.



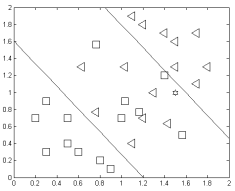
(b) Dataset in $\mathbb{R}^3$, with samples replicated ($h = 1$).



(c) Binary problem in $\mathbb{R}^3$, with filled points representing observations with higher cost of misclassification.



(d) Solution to the binary problem in $\mathbb{R}^3$.



(e) Solution with reject region in the original dataset.

Fig. 5. Proposed reject option model in a toy example.

4.2.1. Selecting the Misclassification Costs

In the reject option scheme, one aims to obtain a minimum error while minimizing the number of rejected cases. However, when the number of rejected cases decreases the classification error increases, and to decrease the classification error one typically has to increase the reject region. The right balance between these two conflicting goals depends on the relation of the associated costs.

Let $C_{i,q}^{(k)}$ represent the cost of erring a point $\mathbf{x}_i$ from class k in data replica q (or, equivalently, by hyperplane q). Points from class $\mathcal{C}_1$ misclassified by the first hyperplane ($\mathbf{w}^t\mathbf{x}+b_1 = 0$) but correctly classified by the second hyperplane ($\mathbf{w}^t\mathbf{x}+b_2 = 0$) incur in a loss $C_{i,1}^{(1)}$; points from class $\mathcal{C}_1$ misclassified by both hyperplanes incur in a loss $C_{i,1}^{(1)}+C_{i,2}^{(1)}$. Likewise, points from class $\mathcal{C}_2$ misclassified by the hyperplane 2 ($\mathbf{w}^t\mathbf{x}+b_2 = 0$) but correctly classified by the first hyperplane ($\mathbf{w}^t\mathbf{x}+b_1 = 0$) incur in a loss $C_{i,2}^{(2)}$; points from class $\mathcal{C}_2$ misclassified by

both hyperplanes incur in a loss $C_{i,1}^{(2)} + C_{i,2}^{(2)}$. The resulting loss matrix is given by

		predicted		
		$\mathcal{C}_1$	$\mathcal{C}_{reject}$	$\mathcal{C}_2$
true	$\mathcal{C}_1$	0	$C_{i,1}^{(1)}$	$C_{i,1}^{(1)} + C_{i,2}^{(1)}$
	$\mathcal{C}_2$	$C_{i,1}^{(2)} + C_{i,2}^{(2)}$	$C_{i,2}^{(2)}$	0

The typical adoption of the same cost for erring and rejecting on the two classes leads to the following simplified loss matrix:

		predicted		
		$\mathcal{C}_1$	$\mathcal{C}_{reject}$	$\mathcal{C}_2$
true	$\mathcal{C}_1$	0	C_l	C_h
	$\mathcal{C}_2$	C_h	C_l	0

Therefore, $C_{reject} = \frac{C_l}{C_h} = w_r$ is the cost of rejecting (normalized by the cost of erring). The data replication method with reject option tries to minimize the empirical risk $w_r R + E$, where R accounts for the rejection rate and E for the misclassification rate.

We conclude this section by highlighting that, in applications with asymmetric cost matrices (different errors have consequences of very different gravity), one can work directly with the original cost matrix, defining different costs in each replica of the data. In data case, the tradeoff between erring in $\mathcal{C}_1$ and rejecting will be different from the equivalent tradeoff for $\mathcal{C}_2$, biasing the reject region in one direction.

4.2.2. Prediction

To predict the class of an unseen example, classify both replicas of the example in the extended dataset with the binary classifier. From the sequence of binary labels one can infer the predicted label on the original ordinal classes

$$\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_1 \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_1 \Longrightarrow \mathcal{C}_{reject} \quad \overline{\mathcal{C}}_2, \overline{\mathcal{C}}_2 \Longrightarrow \mathcal{C}_2$$

Henceforth, the target class can be obtained by counting the number of $\overline{\mathcal{C}}_2$ labels in the sequence, $N_{\overline{\mathcal{C}}_2}$: if $N_{\overline{\mathcal{C}}_2}/2 + 1$ is integer, it yields the target class; otherwise the option is to reject.

4.3. Mapping the Data Replication Method to Learning Algorithms

In this section the method just introduced is instantiated in two important machine learning algorithms: support vector machines and multilayer perceptrons.

4.3.1. Mapping the Data Replication Method with Reject Option to SVMs

The learning task in a classification problem is to select a prediction function $f(\mathbf{x}, \alpha)$ from a family of possible functions that minimizes the expected *loss*, where α is a parameter denoting a particular function in the set.

The SVM classification technique has been originally derived by applying the SRM (structural risk minimization) principle to a two-class problem using the 0/1 (indicator) loss function:

$$L(\mathbf{x}, \alpha, y) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y \end{cases}$$

The simplest generalization of the indicator loss function to classification with reject option is the following loss function

$$L(\mathbf{x}, \alpha, y) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y \\ w_r, & \text{if } f(\mathbf{x}, \alpha) = reject \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y \text{ and } f(\mathbf{x}, \alpha) \neq reject \end{cases}$$

where w_r denotes the cost of rejection (with the cost of erring normalized to 1). Obviously $0 \leq w_r \leq 1$. The corresponding expected risk is

$$R = w_r P(reject) + P(error)$$

as derived in Equation (1) in Section 3. The expression of the empirical risk (R_{emp}) is

$$R_{\text{emp}} = w_r R + E \tag{3}$$

Let us formulate the problem of classifying with reject option in the spirit of SVMs. Starting from the generalization of the two-class separating hyperplane presented in the beginning of previous section, let us look for 2 parallel hyperplanes represented by vector $\mathbf{w} \in \mathbb{R}^d$ and scalars b_1, b_2, such that the feature space is divided into 3 regions by the decision boundaries $\mathbf{w}^t\mathbf{x} + b_r = 0,\ r = 1, 2$.

A pair of parallel hyperplanes which minimizes the empirical risk can be obtained by minimizing the following functional (where sgn (x) returns $+1$ if x is greater than zero; 0 if x equals zero; -1 if x is less than zero)

$$\min_{\mathbf{w}, b_i, \xi_i} \quad \frac{1}{2}\mathbf{w}^t\mathbf{w} + \sum_{q=1}^{2}\sum_{k=1}^{2}\sum_{i=1}^{\ell_k} C_{i,q}^{(k)} \text{ sgn } (\xi_{i,q}^{(k)}) \tag{4}$$

under the constraints

$$\begin{aligned}
&-(\mathbf{w}^t\mathbf{x}_i^{(1)} + b_1) \geq +1 - \xi_{i,1}^{(1)}\\
&+(\mathbf{w}^t\mathbf{x}_i^{(2)} + b_1) \geq +1 - \xi_{i,1}^{(2)}\\
&-(\mathbf{w}^t\mathbf{x}_i^{(1)} + b_2) \geq +1 - \xi_{i,2}^{(1)}\\
&+(\mathbf{w}^t\mathbf{x}_i^{(2)} + b_2) \geq +1 - \xi_{i,2}^{(2)}\\
&\xi_{i,q}^{(k)} \geq 0
\end{aligned}$$

In practice the regularization term $\text{sgn}\,(\xi_{i,q}^{(k)})$ is usually replaced by $\xi_{i,q}^{(k)}$ mainly for computational efficiency.

It is important to note that, although the formulation was constructed from the two-class SVM, it is no longer solvable with the same algorithms. Let us now examine the mapping of the data replication method with reject option on SVMs, which is solvable with a single standard binary SVM classifier.

The rejoSVM The insight gained from studying the toy example paves the way for the formal presentation of the instantiation of the data replication method with reject region in SVMs, rejoSVM.

Following the same procedure delineated in [8], it is straightforward to conclude that the formulation corresponding to the mapping of the data replication method with reject option in SVMs results in

$$\min_{\mathbf{w},b_i,\xi_i} \quad \frac{1}{2}\mathbf{w}^t\mathbf{w} + \frac{1}{2}\frac{1}{h^2}(b_2 - b_1)^2 + \sum_{q=1}^{2}\sum_{k=1}^{2}\sum_{i=1}^{\ell_k} C_{i,q}^{(k)}\ \text{sgn}\,(\xi_{i,q}^{(k)}) \quad (5)$$

with $b_2 = b_1 + w_{p+1}h$ and with the same set of constraints as in (4).

This formulation for the high-dimensional dataset matches the previous formulation (4) up to an additional regularization member in the objective function. This additional member is responsible for the unique determination of the thresholds [8]. We see that the rejoSVM captures the essence of the SRM of SVMs, while being solvable with existing binary SVM classifiers.

4.3.2. Mapping the Data Replication Method with Reject Option to Neural Networks

The mapping of the data replication method with reject option to NNs, rejoNN, is easily accomplished with the architecture proposed for ordinal data in [8]. Nonintersecting boundaries were enforced by making use of a partially linear function $\overline{G}(\overline{\mathbf{x}}) = G(\mathbf{x}) + \underline{\mathbf{w}}^t\mathbf{e}_i$ defined in the extended space (where $\mathbf{e}_i$ equals the vector $[0\ 0\cdots\ 0\ h\ 0\cdots 0]^t$ with dimension $K-2$ and $h>0$ in the i-th position). Setting $G(\mathbf{x})$ as the output of a neural network, a flexible architecture for classification with reject option can be devised, as represented diagrammatically in Fig. 6.

For the mapping of the data replication method with reject option in SVMs and NNs, rejoSVM and rejoNN, if we allow the samples in all the classes to contribute to each threshold, the order inequalities on the thresholds are satisfied automatically, in spite of the fact that such constraints on the thresholds are not explicitly included in the formulation. The proof follows closely the derivation presented in [8] for the oNN algorithm.

4.4. Classifying ordinal data with reject option—a general framework

Although the reject option is usually only considered on binary data, it makes sense to extend it to multiclass data. In particular, the proposed approach extends nicely to ordinal data. In settings where we have K ordered classes, it may be interesting to define $K-1$ reject regions, between class k and class $k+1$, $k=1,\ldots,K-1$.

In the standard data replication method for ordinal data, one would have a data replica for each boundary to be defined ($K-1$ data replicas), requiring $K-2$ extension features. Now, as we need to have two boundaries between consecutive classes, we will use $2(K-1)$ data replicas, requiring $2(K-1)-1$ extension features. The goal is to find $2(K-1)$ boundaries $\mathbf{w}^t\mathbf{x}+b_i$, $i=1,\ldots,2(K-1)$, with reject regions defined between boundaries $2j-1$ and $2j$, $j=1,\ldots,K-1$.

Replicas q and $q+1$, $q=1,3,5,\ldots$ will have exactly the same binary labels, but different costs. Replicas q and $q+1$, $q=2,4,6,\ldots$ will have exactly the same costs, but different binary labels. The boundaries obtained from replicas $2q-1$ and $2q$ will both discriminate $\mathcal{C}_1,\ldots,\mathcal{C}_i$ against $\mathcal{C}_{i+1},\ldots,\mathcal{C}_K$. Table 2 summarizes this setting.

Similarly to the binary case, the prediction of the target class for an unseen examples uses the sequence of $2(K-1)$ labels $\in \{\overline{\mathcal{C}}_1, \overline{\mathcal{C}}_2\}^{2(K-1)}$ by classifying each of the $2(K-1)$ replicas in the ex-

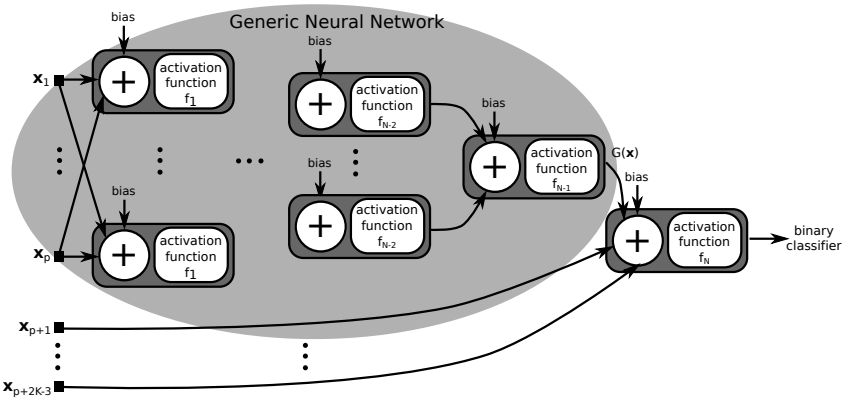


Fig. 6. Data replication method for neural networks with reject option (adapted from [8]).

Replica #	points from $\mathcal{C}_1$	points from $\mathcal{C}_2$	...	points from $\mathcal{C}_{K-1}$	$\mathcal{C}_K$
1	$-1; C_\ell$	$+1; C_h$	$+1; C_h$	$+1; C_h$	$+1; C_h$
2	$-1; C_h$	$+1; C_\ell$	$+1; C_h$	$+1; C_h$	$+1; C_h$
...					
2(K-1)-1	$-1; C_h$	$-1; C_h$	$-1; C_h$	$-1; C_\ell$	$+1; C_h$
2(K-1)	$-1; C_h$	$-1; C_h$	$-1; C_h$	$-1; C_h$	$+1; C_\ell$

Table 2

Labels and costs (C_ℓ and C_h represent a low and a high cost value, respectively) for points in different replicas in the extended dataset.

tended dataset with the binary classifier. The target class can be obtained by counting the number of $\overline{\mathcal{C}}_2$ labels in the sequence, $N_{\overline{\mathcal{C}}_2}$: if $N_{\overline{\mathcal{C}}_2}/2+1$ is integer, is yields the target class; otherwise the option is to reject.

5. Two classifiers approach for ordinal data with reject option

In this section, and for experimental comparison purposes, we introduce an extension to ordinal data of the two-classifier approach for binary data with reject option. The extension involves a simple adaptation of the method for ordinal data presented in [14]. Frank and Hall [14] proposed to use $(K-1)$ standard binary classifiers to address the K-class ordinal data problem. Toward that end, the training of the i^{th} classifier is performed by converting the ordinal dataset with classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$ into a binary dataset, discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$ (see Fig. 7).

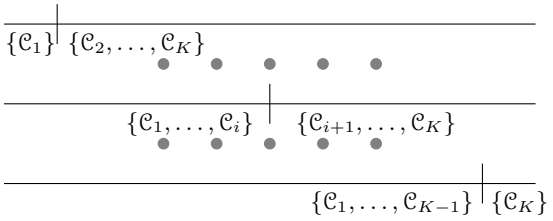


Fig. 7. Transformation of an ordinal data classification problem in (K-1) binary problems.

The i^{th} classifier represents the test $\mathcal{C}_x > \mathcal{C}_i$. To predict the class value of an unseen instance, the $K-1$ binary outputs are combined to produce a single estimation. The extension of the two classifiers approach for reject option to ordinal data involves replacing the i^{th} classifier in Frank and Hall method by two classifiers, both discriminating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$ but trained with different costs, exactly as given in Table 2 for our proposal. Observe that, under our approach, the $(2i-1)^{th}$ and $(2i)^{th}$ boundaries are also discrimi-

nating $\mathcal{C}_1, \ldots, \mathcal{C}_i$ against $\mathcal{C}_{i+1}, \ldots, \mathcal{C}_K$; the major difference lies in the independence of the boundaries found with Frank and Hall's method. This independence is likely to lead to intersecting boundaries.

6. Implementation

In the following subsections we will outline three algorithms regarding the reject option approaches identified in Section 2. First we present in Section 6.1 and in Section 6.2 the datasets used in our experimental study. In Section 6.3 we outline the general setup of the experiments conducted in this work. In Section 6.4 and in Section 6.5 we present the algorithms for the two and one classifier approach extended to the multiclass ordinal problem according to the description given in Section 5. Finally, in Section 6.6 it is presented the algorithm for the method for learning the reject region in an ordinal setting proposed in this paper. Before all these, we describe the benchmark datasets.

6.1. Benchmark binary datasets

The performance of the classification methods were assessed over four binary datasets. The first two were synthetically generated; the remaining two datasets include real data from diverse applications.

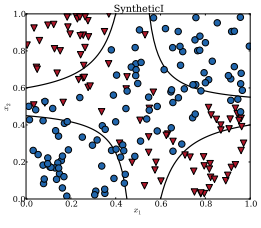


Fig. 8. Sample of 100 examples from `syntheticI` dataset.

For the first synthetic dataset—henceforth called `syntheticI`—, we began by generating 400 example points $\mathbf{x} = [x_1 \;\; x_2]^t$ in the unit square $[0,1] \times [0,1] \subset \mathbb{R}^2$ according to a uniform distribution. Then, we assigned to each example $\mathbf{x}$ a class $y \in \{-1, +1\}$ corresponding to

$$(b_{-2}, b_{-1}, b_0, b_1) = (-\infty; -0.5; 0.25; +\infty)$$

$$\varepsilon_1 \sim N(0, 0.125^2)$$

$$\alpha = 10(x_1 - 0.5)(x_2 - 0.5)$$

$$t = \min_{r \in \{-1,0,+1\}} \{r : b_{r-1} < \alpha + \varepsilon_1 < b_r\}$$

$$\varepsilon_2 \sim \text{Uniform}(b_{-1}, b_0)$$

$$y = \begin{cases} t, & t \neq 0 \\ +1, & t = 0 \wedge \varepsilon_2 < \alpha \\ -1, & t = 0 \wedge \varepsilon_2 > \alpha \end{cases}$$

This distribution creates two plateau uniformly distributed and a transition zone of linearly decreasing probability, delimited by hyperbolic boundaries. Fig. 8 depicts a sample of 100 examples drawn according to this distribution. The two boundaries correspond to $\alpha = b_{-1}$ and $\alpha = b_0$.

A second synthetic dataset of 400 points—`syntheticII`—was generated from two Gaussian in $\mathbb{R}^2$: $\mathbf{y}_{-1} \sim N\left(\begin{bmatrix}-2\\-2\end{bmatrix}, \begin{bmatrix}9 & 0\\0 & 9\end{bmatrix}\right) + \varepsilon$ and $\mathbf{y}_{+1} \sim N\left(\begin{bmatrix}+2\\+2\end{bmatrix}, \begin{bmatrix}25 & 0\\0 & 25\end{bmatrix}\right) + \varepsilon$ corresponding to classes $\{-1, +1\}$ respectively, where ε follows a uniform distribution in $[0.025, 0.25]$.

The third dataset, encompassing 1144 observations, expresses the aesthetic evaluation of Breast Cancer Conservative Treatment [7,25]. For each patient submitted to BCCT, 30 measurements were recorded, capturing visible skin alterations or changes in breast volume or shape. In this work we used only 4 measures as identified in [25] as the most relevant ones. The aesthetic outcome of the treatment for each and every patient was classified in one of the four categories: Excellent$\succ$Good$\succ$Fair$\succ$Poor. For the experimental work with binary models, the multiclass problem was transformed into a binary one, by aggregating Excellent and Good in one class, and the Fair and Poor cases in another class. Another dataset consisting of the English alphabet, publicly available on the UCI machine learning repository, is composed of $20,000$ instances with 16 features describing the 26 capital letters. Each instance is mainly defined by statistical moments and edge counts. In our experiments we used a subset of the whole dataset comprehending only the discrimination of the letter A versus the letter H.

6.2. Benchmark multiclass datasets

To evaluate the generalization of our approach, we extended the `syntheticI` dataset into another different dataset, `syntheticIII`, generated similarly as `syntheticI` but now with five classes.

$$(b_{0.5}, b_1, b_{1.5}, b_2, b_{2.5}, b_3, b_{3.5}, b_4, b_{4.5}, b_5)$$
$$= (-\infty; -1.5; -1.25; -1; -0.5; -0.1; 0.1, 0.5; 1.1; +\infty)$$

Another dataset named `syntheticIV` was used in our experiments. This dataset is an extension of the syntheticII with one additional class generated accordingly to the Gaussian distribution with mean $[7\ 7]^t$ and covariance $\Sigma = 4\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Regarding to the `BCCT` dataset, we used the original multiclass problem: Excellent$\succ$Good$\succ$Fair$\succ$Poor. Finally, we also used the `LEV`[4] dataset which contains examples of anonymous lecturer evaluations, taken at the end of MBA courses and is composed by 4 features and 5 classes.

6.3. Methodology

We randomly split each dataset into training and test sets; in order to study the effect of varying the size of the training set, we considered three possibilities: 5%, 25% and 40% of all the data available. The splitting of the data into training and test sets was repeated 50 times in order to obtain more stable results for accuracy by averaging and also to assess the variability of this measure. The best parametrization of each model was found by 'grid-search', based on a 5-fold cross validation scheme conducted on the training set. Finally, the error of the model was estimated on the test set. The 'grid-search' was performed over the $C = 2^{-5}, \ldots, 2^3$ and $\gamma = 2^{-3}, \ldots, 2^1$ values when using the RBF kernel for the SVMs methods on the `BCCT` (multiclass) and `LEV` datasets and polynomial of degree 2 for the synthetic datasets. For the neural network techniques, we performed a 'grid-search' over the number of neurons (5 to 25) with one-hidden layer. Regarding specifically to rejoNN, we also had to tune the h and s parameters. The s parameter controls the size of the extended dataset, by controlling the classes that are present in each replica [7]. The range of tested values were 1, 1.5 and 2 for h, and 2 and 4 for s in the binary datasets. We fixed the values for $h = 10$ and $s = 3$ in the ordinal datasets. To train the networks on all methods we used the resilient back-propagation algorithm available in MATLAB ™. For the binary datasets the number of epochs for all methods was set to be 15 whereas for the ordinal datasets we had to tune the best number without degrading the overall results. We experimentally verified that the number of epochs never exceeded 100 for rejoNN and the remaining MLP techniques. We have also used a network with K outputs, one corresponding to each class, and target values of 1 for the correct class and 0 otherwise.

6.4. Design of two independent *classifiers*

One of the standard procedures identified in Section 2 to define the reject region is through the design of independent classifiers. This approach can be straightforwardly extended to the ordinal problems and is described in Algorithm 1. We first train a first classifier with a set of weights heavily penalizing the false negative errors in order to obtain truly negative predictions; then, train a second classifier with a set of weights heavily penalizing the false positive errors in order to obtain truly positive predictions—see Table 2 (here the replicas correspond to the different discriminants). In the end, we will have two classifiers, each one specialized in a given class.

6.5. Design of a single classifier

The algorithm structure for learning the reject region with a single classifier is described in Algorithm 2.

First we train a model and the reject region is determined only *after*. If the classifier provides some approximation to the posterior class probabilities, then a pattern is rejected if the maximum of the two posterior probabilities is lower than a given threshold. Otherwise, it is used a rejection threshold targeted to a particular classifier.

Algorithm 1. Algorithm structure for the two classifiers approach.

1: **Input**: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}, \mathcal{D}^* = \{\mathcal{X}^*, \mathcal{Y}^*\}$ the training and testing datasets, respectively ($\mathcal{D}, \mathcal{D}^*$ are disjoin datasets).
2: **Output**: $\mathcal{Y}^*_{w_r}$ testing set prediction $\forall w_r \in]0, \ldots, 0.5[$.
3: **for** $w_r \in]0, \ldots, 0.5[$ **do**
4: **for all** possible combinations of model parameters, p_i **do**
5: Split $\mathcal{D}$ in 5 equal partitions, $\mathcal{D}^{(v)} = \{\mathcal{X}^{(v)}, \mathcal{Y}^{(v)}\}, v = \{1, \ldots, 5\}$, such that $\mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(5)}$ are disjoin sets
6: **for** $v \leftarrow 1$, 5 **do**
7: **for** $k \leftarrow 1$, $K-1$ **do**
8: `costs` $\leftarrow$ set costs according Table 2
9: $\mathcal{Y}_o \leftarrow \{\}$
10: **for all** $y \in \mathcal{Y}^{(1,\ldots,5)\setminus v}$ **do**
11: **if** $y \leq k$ **then** $\mathcal{Y}_o \leftarrow \mathcal{Y}_o \cup \{-1\}$
12: **else**$\mathcal{Y}_o \leftarrow \mathcal{Y}_o \cup \{+1\}$
13: **end if**
14: **end for**
15: $\mathcal{M}_{2k-1} \leftarrow$ `TrainModel`$(\mathcal{X}, \mathcal{Y}_o$,`costs`$)$
16: $\mathcal{M}_{2k} \leftarrow$ `TrainModel`$(\mathcal{X}, \mathcal{Y}_o$,`costs`$)$
17: **end for**
18: validate $\mathcal{M}_1 \cup \ldots \cup \mathcal{M}_{2(K-1)}$ performance according to Equation (3) given $\mathcal{D}^v$
19: **end for**
20: save the parametrization resulting of the best mean validation performance
21: **end for**
22: train the 2(K-1) models, $\mathcal{M}_k$, with dataset $\mathcal{D}$ according lines 7–17
23: $\mathcal{Y}_k \leftarrow \{\}$
24: **for all** models $\mathcal{M}_k, k = \{1, \ldots, 2(K-1)\}$ **do** ▷ predict and change negative responses to zero
25: $\mathcal{Y}_k \leftarrow \mathcal{Y}_k \cup$ `TestModel`$(\mathcal{X}^*, \mathcal{M}_k)$
26: **end for**
27: **if** $\mod \left(\sum_{k=1}^{2(K-1)} \mathcal{Y}_k, 2\right)$ equals 0 **then** $\mathcal{Y}^*_{w_r} \leftarrow 1 + \left(\sum_{k=1}^{2(K-1)} \mathcal{Y}_k\right)/2$
28: **else**$\mathcal{Y}^*_{w_r} \leftarrow$ Reject
29: **end if**
30: **end for**

Algorithm 2. Algorithm structure for the one classifier approach.

1: **Input**: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ the training dataset and $\mathcal{X}^*$ the testing set.
2: **Output**: $\mathcal{Y}^*_{w_r}$ testing set prediction $\forall w_r \in]0, \ldots, 0.5[$.
3: $\mathcal{M} \leftarrow$ `TrainModel`$(\mathcal{X}, \mathcal{Y})$ ▷ train model according a standard 5 fold cross-validation procedure to find best model parametrization
4: Obtain the posterior probabilities $(\mathcal{P}_1, \ldots, \mathcal{P}_K)$ of $\mathcal{X}$ given model $\mathcal{M}$
5: **for** $w_r \in]0, \ldots, 0.5[$ **do**
6: obtain `bestthreshold` $\in [0.5, \ldots, 1]$, that minimizes Equation (3) given $\mathcal{D}$ and $\mathcal{P}$
7: $(\mathcal{Y}_{pred}, \mathcal{P}_{max}) \leftarrow$ `TestModel`$(\mathcal{X}^*, \mathcal{M})$, where $\mathcal{P}_{max} = \max(\mathcal{P}_1, \ldots, \mathcal{P}_K)$
8: **if** $\mathcal{P}_{max} <$ `bestthreshold` **then** $\mathcal{Y}^*_{w_r} \leftarrow$Reject
9: **else**$\mathcal{Y}^*_{w_r} \leftarrow \mathcal{Y}_{pred}$
10: **end if**
11: **end for**

Algorithm 3. Algorithm structure for the rejoSVM classifier approach.

1: **Input**: $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ the training dataset and $\mathcal{X}^*$ the testing set composed by N instances.
2: **Output**: $\mathcal{Y}^*_{w_r}$ testing set prediction $\forall w_r \in]0, \ldots, 0.5[$.
3: **for all** $w_r \in]0, \ldots, 0.5[$ **do**
4: $(\mathcal{X}^{rep}, \mathcal{Y}^{rep}, C^{rep}) \leftarrow$ replicate dataset $\mathcal{D}$ according Table 2 ▷ C^{rep} is all the $C^{(k)}_{i,j}$ as represented in Table 2 and in Equation (5)
5: $(\mathcal{X}^{*rep}) \leftarrow$ replicate dataset $\mathcal{D}^*$ ▷ Optimize function from Equation (5) or the NN represented in Fig. 6
6: $\mathcal{M} \leftarrow$ `TrainModel`$(\mathcal{X}^{rep}, \mathcal{Y}^{rep}, C^{rep})$
7: $\mathcal{Y}_1 \leftarrow$ `TestModel`$(\mathcal{X}^{*rep}, \mathcal{M})$ ▷ convert $\mathcal{Y}_1$ replicas prediction to a single K class
8: $Y^{*(j)}_{w_r} \leftarrow 1 + \sum_{i=1}^{p+K-2} y_1^{(i)}, \quad \forall j = 1, \ldots, N, \quad y_1 \in \mathcal{Y}_1$
9: **end for**

6.6. Design of rejoSVM and rejoNN

To learn the reject option based on the data replication method proposed in [8], we have to modify the misclassification costs of the observations according to the data replica they belong to. Such is performed according Table 2 as already mentioned in Section 4.2. This can be easily done by adjusting the C tradeoff with the misclassification costs as represented in Equation (5).

For the neural network approach, rejoNN, we changed the error function, $e_k(n)$, where we modify the misclassification costs according to the data replica as before. Formally,

$$e_k(n) = (d_k(n) - y_k(n))C_n \tag{6}$$

where $d_k(n)$ is the response given by output neuron k for the input pattern n and $y_k(n)$ the desired response (true label). C_n corresponds to a given $C^{(k)}_{i,q}$ from Equation (5) represented here for syntax simplicity.

The algorithm structure for learning the reject region as proposed in this paper is described in Algorithm 3.

Function `TrainModel` in line 6 of Algorithm 3 can be a single binary classifier according to Equation (5) in the case of a binary SVM. The formulation for the multiclass case can be found in [8] subject to the costs present in Table 2.

7. Experimental Results

In the following subsections, experimental results are provided for several models based on SVMs and NNs, when applied to diverse datasets, ranging from synthetic to real data, for binary and ordinal data. The set of models under comparison include the proposed rejoSVM and rejoNN methods, the "one classifier" approach and "two classifiers" approach (SVM and Multi-Layer Perceptrons—MLPs), and Fumera [16] method.

The major reason for comparing our proposal (rejoSVM, rejoNN) against Fumera [16] resides on the most fundamental principles which both methods share. The minimization of the empirical risk with the optimum reject rule proposed by Chow [9] as succinctly presented in Section 3 represents the same basis for both methods. However, and to the best of our knowledge, the most recent works do not explore this concept and hence a fair comparison would not be possible.

"One classifier" and "two classifiers" are naïve reject option learning schemes as referred in Section 2. The "one classifier" was also used in Fumera [16] as baseline. As a remark, the "two classifiers" approach is formed by 2(K-1) classifiers. However, and for the sake of simplicity, we will refer to it only as "two classifiers" approach as mentioned in Section 5.

The work was performed in a reproducible research manner, and the MATLAB ™code needed to reproduce all reported results is available at `http://www.dcc.fc.up.pt/~rsousa`. The proposed rejoSVM is based on the binary SVM from the Bioinformatics Toolbox and the rejoNN is based on the Neural Network Toolbox. We thank G. Fumera for providing the source code (in C/C++) of his method. Note that this method is for SVMs only and the provided implementation works only with linear kernels.

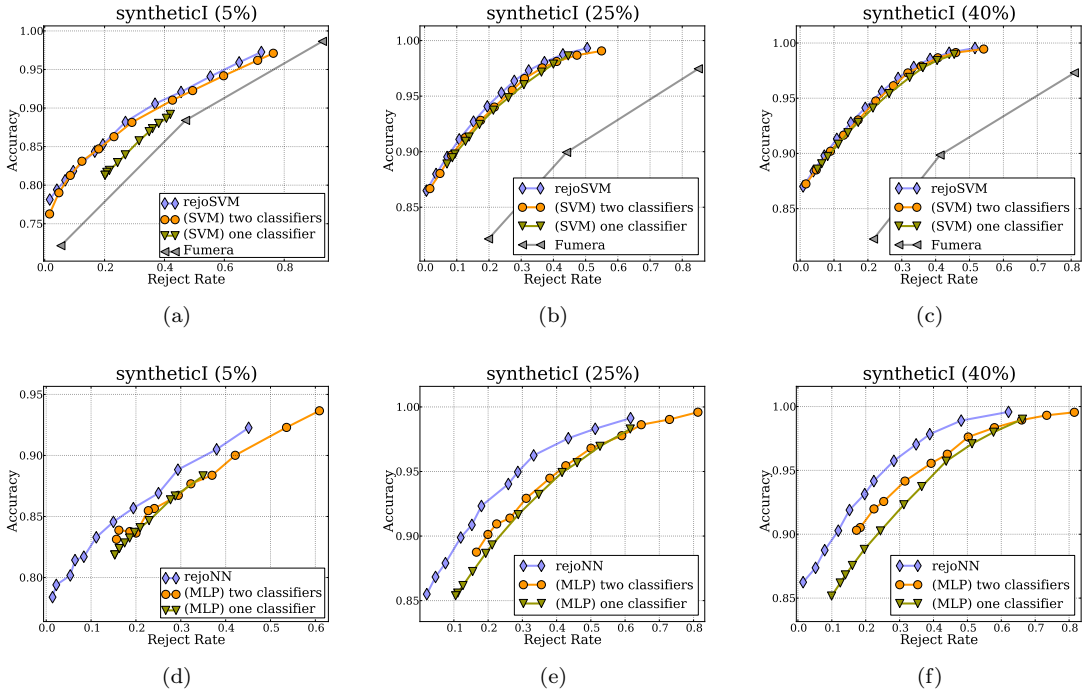


Fig. 9. The A-R curves for the `syntheticI` dataset. (a)–(c): SVM methods only; (d)–(f): NN methods only. 5%, 25% and 40% of training data, respectively.

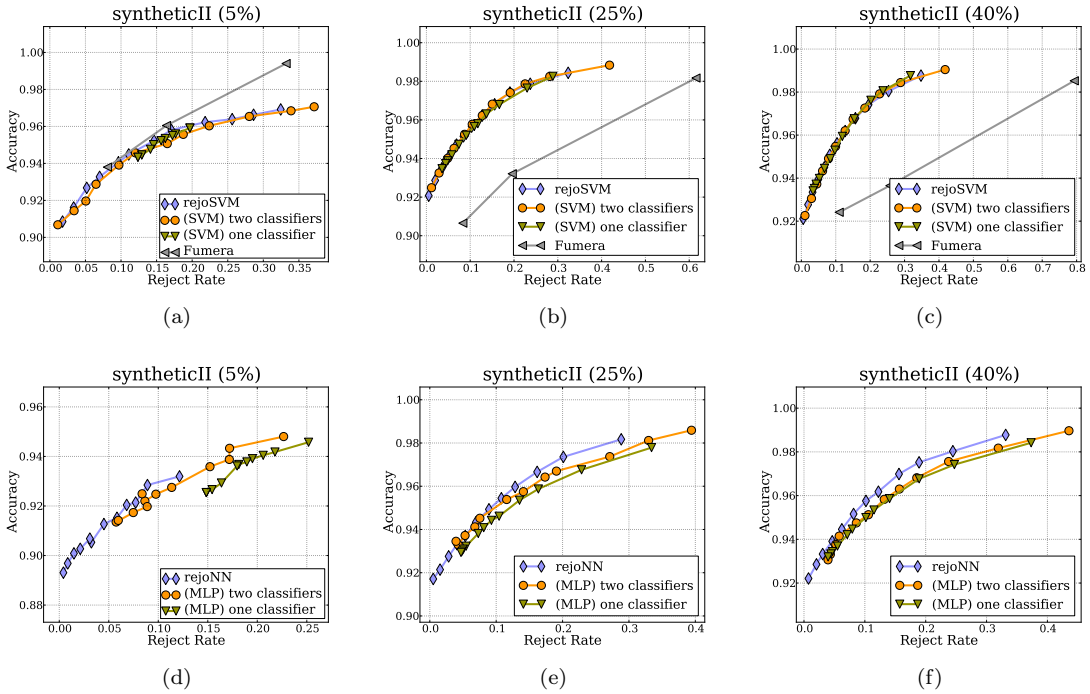


Fig. 10. The A-R curves for the `syntheticII` dataset. (a)–(c): SVM methods only; (d)–(f) NN methods only. 5%, 25% and 40% of training data, respectively.

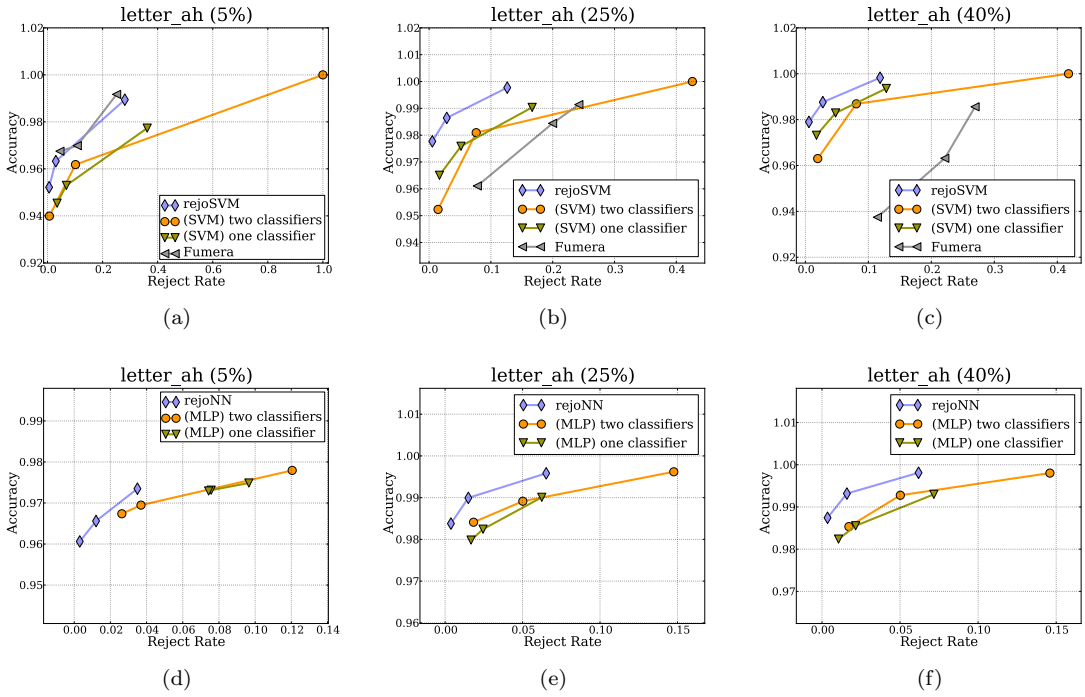


Fig. 11. The A-R curves for the `letter AH` dataset. (a)–(c): SVM methods only; (d)–(f) NN methods only. 5%, 25% and 40% of training data, respectively.

7.1. Results

The performance of a classifier with reject option can be represented by the classification accuracy achieved for any value of the reject rate (the so-called Accuracy-Reject curve). The trade-off between errors and rejections depends on the rejection cost w_r. Meaning that the w_r parameter (corresponding to each breaking-point in the A-R curves) is associated to the cost of rejecting an instance in a given problem producing thus different reject rates. We considered values of w_r less than 0.5, as above this value it is preferable to just try to guess randomly [9]. In some cases, only three values of w_r were used due to computational issues.

Fig. 9 to Fig. 16 summarize the results obtained for all datasets. A first main assertion is that in overall rejoSVM and rejoNN performed better than any of the other methods under comparison, over the full range of values for w_r, specially, on the binary datasets. Moreover, since only linear kernels were implemented in the Fumera method, we extended the datasets with second order terms $x_i x_j$ when evaluating this method. In this extended space, the optimal solutions for the synthetic datasets are indeed linear. On the ordinal datasets, rejoSVM and rejoNN achieved competitive results with standard procedures.

With the increase of the training dataset size, as expected, we see that none of the methods outperform the others. A major conclusion based on this empirical analysis is that rejoSVM performs well with few training instances. Nonetheless, this can cause some irregularities on the curves, specially on neural networks, as can be depicted in Fig. 9(d) and Fig. 10(d). In Fig. 11 it is shown the evolution of the different reject methodologies with

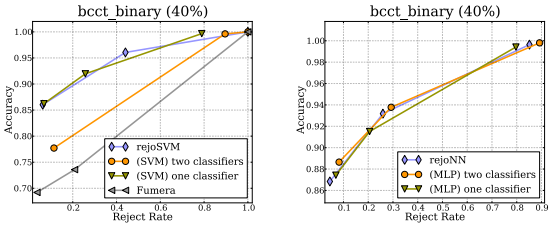
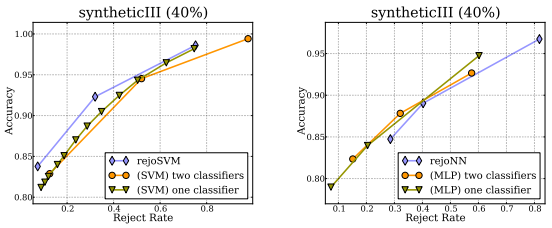


Fig. 12. The A-R curves for the binary `BCCT` dataset. (a) SVM methods, and (b) NN methods only with 40% of training data.

the real-world dataset Letter A vs. H. The performance trend of rejoSVM and rejoNN in comparison with the other approaches shows the benefit of capturing the reject regions during the training phases. In terms of the applicability on the incorporation of the reject option in medical aiding systems, one can verify that rejecting roughly 20% of the training set size (5% of 1144 observations) one can attain an accuracy on the order of 90%—see Fig. 12(a). Fig. 13 shows a clear gain

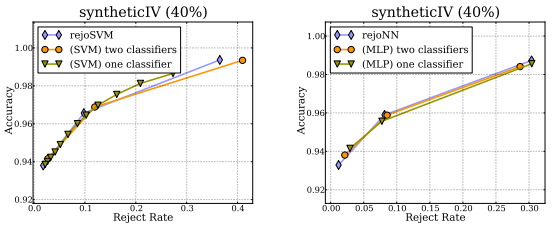


(a) (b)

Fig. 13. The A-R curves for the `syntheticIII` dataset. (a) SVM methods, and (b) NN methods only with 40% of training data.

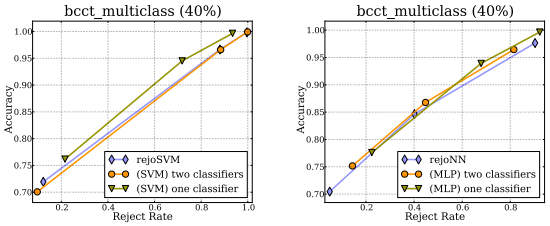
of rejoSVM whereas rejoNN presents competitive results. The same conclusions can be drawn by analysing Fig. 14 and Fig. 16. On the full `BCCT` class set depicted in Fig. 15, despite all methods performing increasingly better with an increasing training dataset size, "one classifier" approach attains the best results. However, in the Neural Network approaches, rejoNN achieves competitive results.

It is also observable that, in general, SVM based methods outperform the neural network counterparts, in line with the current view in the research community. When restricting the attention to neural network methods, the proposed rejoNN



(a) (b)

Fig. 14. The A-R curves for the `syntheticIV` dataset. (a) SVM methods, and (b) NN methods only with 40% of training data.



(a) (b)

Fig. 15. The A-R curves for the multiclass `BCCT` dataset. (a) SVM methods and, (b) NN methods only with 40% of training data.

exhibits often the best performance. Moreover, it is important to emphasize that rejoSVM and rejoNN approaches have the advantage of simplicity, using a single direction for all boundaries, and interpretability. The insight of looking to the reject option problem as an ordinal class setting can promote new lines of research.

Finally, we highlight that the proposed framework: 1) has the capability to detect reject regions with a single standard binary classifier; 2) does not required the addition of any confidence level, or thresholds, to define the trust regions; and 3) does not generate ambiguity regions as the "two classifiers" approach, as it was presented in Fig. 2(a).

8. Conclusion

Despite the myriad of techniques that handle the incorporation of a reject option in their approaches, many of them do not fully account the pioneer work of [9]. In this paper, we proposed an extension of the data replication method [8] that directly embeds reject option. This extension was derived by taking a new perspective of the classification with reject option problem, viewing the three output classes as naturally ordered. A pair of non-intersecting boundaries delimits the rejection region provided by our model. Our proposal has the advantages of using a standard binary classifier and embedding the design of the reject region during the training process. Moreover, the method allows a flexible definition of the position and orientation of the boundaries, which can change for different values of the cost of rejections w_r. This method was mapped into neural networks and support vector machines with very positive results. This work can be a useful contribution in the area

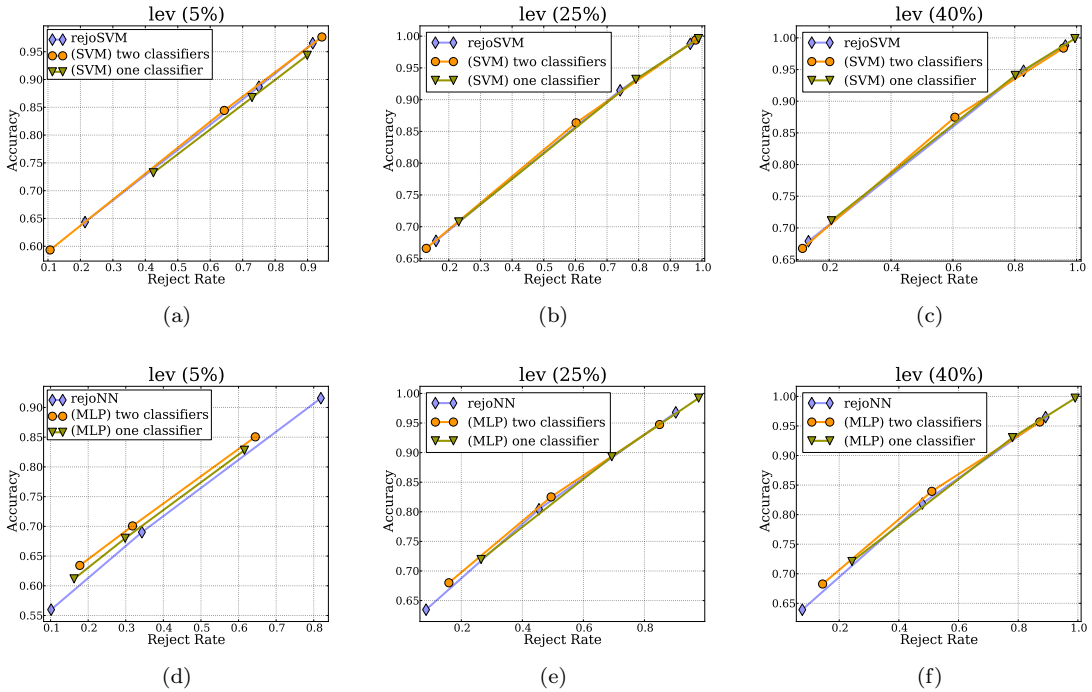


Fig. 16. The A-R curves for the LEV dataset. (c): SVM methods only; (f): NN methods only. 5%, 25% and 40% of training data, respectively.

and the availability of the code under the reproducible research guidelines can encourage others to make use of and to build on it.

Acknowledgements

This work is financed by the ERDF - European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the Fundação para a Ciência e Tecnologia (FCT) within project PTDC/SAU-ENB/114951/2009. The first author would like to thank Guilherme de Alencar Barreto for comments and suggestions provided. Finally, we would also like to acknowledge the insightful comments of the two anonymous reviewers.

References

[1] Ali Ahmadi, Sigeru Omatu, Toru Fujinaka, and Toshihisa Kosaka. Improvement of reliability in banknote classification using reject option and local PCA. *Information Sciences*, 168(1-4):277–293, 2004.

[2] Peter L. Bartlett and Marten H. Wegkamp. Classification with a Reject Option using a Hinge Loss. *Journal Machine Learning Research*, 9:1823–1840, 2008.

[3] Riccardo Bellazzi, Ameen Abu-Hanna, and Jim Hunter, editors. *Artificial Intelligence in Medicine, 11th Conference on Artificial Intelligence in Medicine, AIME 2007, Amsterdam, The Netherlands, July 7-11, 2007, Proceedings*, volume 4594 of *Lecture Notes in Computer Science*. Springer, 2007.

[4] Arie Ben-David and Leon Sterling. Generating rules from examples of human multiattribute decision making should be simple. *Expert Systems with Applications*, 31(2):390 – 396, 2006.

[5] Abdenour Bounsiar, Pierre Beauseroy, and Edith Grall-Maës. General solution and learning method for binary classification with performance constraints. *Pattern Recognition Letters*, 29(10):1455–1465, 2008.

[6] Abdenour Bounsiar, Edith Grall-Maës, and Pierre Beauseroy. A kernel based rejection method for supervised classification. In *International Journal of Computational Intelligence*, pages 312–321, 2006.

[7] Jaime S. Cardoso and Maria J. Cardoso. Towards an intelligent medical system for the aesthetic evaluation of breast cancer conservative treatment. *Artificial Intelligence in Medicine*, 40:115–126, 2007.

[8] Jaime S. Cardoso and Joaquim F. Pinto da Costa. Learning to classify ordinal data: the data replication method. *Journal of Machine Learning Research*, 8:1393–1429, 2007.

[9] C. Chow. On optimum recognition error and reject tradeoff. *Information Theory, IEEE Transactions on*, 16(1):41–46, 1970.

[10] Pandu Ranga Rao Devarakota, Bruno Mirbach, and Björn Ottersten. Reliability estimation of a statistical classifier. *Pattern Recognition Letters*, 29:243–253, February 2008.

[11] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, 2001.

[12] César Ferri, Peter Flach, and José Hernández-Orallo. Delegating classifiers. In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004.

[13] Cèsar Ferri and José Hernández-Orallo. Cautious classifiers. In *Proceedings of the 1st International Workshop on ROC Analysis in Artificial Intelligence (ROCAI-2004)*, pages 27–36, 2004.

[14] Eibe Frank and Mark Hall. A simple approach to ordinal classification. In *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, pages 145–156, London, UK, 2001. Springer-Verlag.

[15] Keinosuke Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[16] Giorgio Fumera and Fabio Roli. Support Vector Machines with Embedded Reject Option. In *SVM '02: Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines*, pages 68–82, London, UK, 2002. Springer-Verlag.

[17] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Multiple reject thresholds for improving classification reliability. In *Proceedings of the Joint IAPR International Workshops on Advances in Pattern Recognition*, pages 863–871, London, UK, 2000. Springer-Verlag.

[18] Giorgio Fumera, Fabio Roli, and Giorgio Giacinto. Reject option with multiple thresholds. *Pattern Recognition*, 33(12):2099–2101, 2000.

[19] João Gama and Pavel Brazdil. Cascade Generalization. *Machine Learning*, 41(3):315–343, December 2000.

[20] Yves Grandvalet, Alain Rakotomamonjy, Joseph Keshet, and Stéphane Canu. Support vector machines with a reject option. In Daphne Koller, Dale Schuurmans, Yoshua Bengio, and Léon Bottou, editors, *NIPS*, pages 537–544. MIT Press, 2008.

[21] Thomas C. W. Landgrebe, David M. J. Tax, Pavel Paclík, and Robert P. W. Duin. The interaction between classification and reject performance for distance-based reject-option classifiers. *Pattern Recognition Letters*, 27:908–917, June 2006.

[22] Thomas C. W. Landgrebe, David M. J. Tax, Pavel Paclik, Robert P.W. Duin, and Colin Andrew. A combining strategy for ill-defined problems. In *Fifteenth Ann. Sympos. of the Pattern Recognition Association of South Africa*, pages 57–62, November 2004.

[23] H. Le Capitaine and C. Fré andlicot. An optimum class-rejective decision rule and its evaluation. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 3312–3315, August 2010.

[24] Ajalmar R. R. Neto, Ricardo Sousa, Guilherme Barreto, and Jaime S. Cardoso. Diagnostic of pathology on the vertebral column with embedded reject option. In *Proceedings of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*, 2011.

[25] Helder Oliveira, Andre Magalhaes, Maria J. Cardoso, and Jaime S. Cardoso. An accurate and interpretable model for bcct.core. In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6158–6161, 2010.

[26] Tadeusz Pietraszek. Optimizing abstaining classifiers using ROC analysis. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 665–672, New York, NY, USA, 2005. ACM.

[27] Ricardo Sousa, Beatriz Mora, and Jaime S. Cardoso. An ordinal data method for the classification with reject option. In *Proceedings of The Eighth International Conference on Machine Learning and Applications (ICMLA 2009)*, 2009.

[28] D. M. J. Tax and R. P. W. Duin. Growing a multiclass classifier with a reject option. *Pattern Recognition Letters*, 29:1565–1570, July 2008.

[29] Lyn C. Thomas, David B. Edelman, and Jonathan N. Crook. *Credit Scoring and Its Applications*. SIAM, 2002.

[30] Francesco Tortorella. Reducing the classification cost of support vector classifiers through an ROC-based reject rule. *Pattern Analysis and Applications*, 7:128–143, July 2004.

[31] Francesco Tortorella. A ROC-based reject rule for dichotomizers. *Pattern Recognition Letters*, 26:167–180, January 2005.

[32] Ming Yuan and Marten Wegkamp. Classification methods with reject option based on convex risk minimization. *Journal of Machine Learning Research*, 11:111–130, March 2010.

[33] R. Zhang and D. Metaxas. RO-SVM: Support vector machine with reject option for image categorization. In *BMVC06*, pages 1209–1218, 2006.