

Cost-sensitive learning in Support Vector Machines

Giorgio Fumera, and Fabio Roli

Department of Electrical and Electronic Engineering, University of Cagliari

Piazza d'Armi, 09123 Cagliari, Italy

email: {fumera,roli}@diee.unica.it

Abstract. In this paper, a cost-sensitive learning method for support vector machine (SVM) classifiers is proposed. We focus on a particular case of cost-sensitive problems, namely, classification with reject option. Standard learning algorithms, the one for SVMs included, are not cost-sensitive. In particular, they can not handle the reject option. However, we show that, under the framework of the structural risk minimisation induction principle, on which standard SVMs are based, the rejection region should be determined during the training phase of a classifier, by the learning algorithm. We apply this approach to develop a cost-sensitive SVM classifier, by following Vapnik's maximum margin method to the derivation of standard SVMs. This lead us to a SVM with embedded reject option. To implement such a SVM, we develop a novel formulation of the training problem, and a specific algorithm to solve it. Preliminary results on a character recognition problem seem to show the advantages of the proposed cost-sensitive SVM, in terms of the achievable error-reject trade-off.

1 Introduction

Cost-sensitive classification problems are characterised by different costs for different classification errors. Let be $c(i, j)$ the cost of deciding for class i when the true class is j . To minimise the expected value of the classification cost (named expected risk), the optimal decision rule is to assign an input pattern \mathbf{x} to the class k such that:

$$k = \arg \min_{i=1, \dots, m} \sum_{j=1}^m c(i, j) P(j | \mathbf{x}), \quad (1)$$

where m is the number of classes, and $P(j | \mathbf{x})$ denotes the j -th class posterior probability for pattern \mathbf{x} . It is worth pointing out that achieving the minimum of the expected risk does not coincide with minimising the error probability, unless the costs of errors and correct classifications do not depend on the classes. However, it is well-known that standard learning algorithms are designed to minimise the error probability. Accordingly, they are not cost-sensitive. This raises the question of how a classifier trained with a standard learning algorithm can be used for a cost-sensitive problem. This issue was addressed for the case of two-class problems in [1], where two approaches were considered. The first approach consists in changing the proportion of positive and negative training patterns, according to the classification costs. The learning algorithm is then applied to a rebalanced training set. The second approach consists in directly applying the classification rule (1) to the probability estimates provided by a classifier trained on the unchanged training set. In

[1] it was argued that the second approach is best suited for standard Bayesian and decision tree learning methods, on which the first approach has little effect.

Let us now consider another case of cost-sensitive classification problem, namely, classification with reject option. The reject option is useful for applications which require a high classification reliability. It consists in avoiding the automatic classification of patterns for which the reliability of the classification is considered not sufficient. Rejected patterns have obviously a cost, since they must be handled with different procedures (for instance, manual classification). In the following we consider the simplest case in which the costs of misclassifications, rejects, and correct classifications do not depend on the classes. These costs will be denoted respectively as c_E , c_R , and c_C (obviously, $c_E > c_R > c_C$). The optimal decision rule for this problem was defined by Chow [2]. Chow's rule consists in accepting a pattern \mathbf{x} , and assigning it to the class with maximum posterior probability, if it is higher than a predefined reject threshold T :

$$\text{assign } \mathbf{x} \text{ to class } i = \arg \max_j P(j | \mathbf{x}), \text{ if } P(i | \mathbf{x}) \geq T = \frac{c_E - c_R}{c_E - c_C}. \quad (2)$$

The pattern \mathbf{x} is otherwise rejected. As shown in (2), the trade-off between errors and rejections depends on the costs, through the value of T . This is therefore a cost-sensitive classification problem, even if all misclassifications have the same cost. Standard learning algorithms are not cost-sensitive even with regard to classification with reject option, since they are not designed to provide the reject decision. For classifiers based on standard learning algorithms, the reject option is usually implemented by using the second of the approaches described above. Namely, Chow's rule (2) is applied to the estimates of the a posteriori probabilities provided by a trained classifier. For instance, this is the case of neural networks and k -nearest neighbours classifiers [3]. For classifiers which do not provide estimates of the a posteriori probabilities, heuristic rejection rules targeted to the particular classifier are used. Anyway, such rules are usually based on reject thresholds applied to the outputs of the trained classifier. This corresponds again to the second of the above approaches.

In this paper, we focus on the problem of implementing the reject option in support vector machine (SVM) classifiers [4]. SVMs are a recently introduced technique for pattern recognition, regression, and density estimation problems. The SVM learning method is based on the structural risk minimisation (SRM) induction principle, which was derived from the statistical learning theory [5]. SVMs have proven to be effective in many practical applications. However, SVMs are not cost-sensitive, like traditional classifiers. In particular, despite of their strong theoretical foundations, only heuristic rules have been proposed so far for implementing the reject option. Since SVMs do not provide estimates of the a posteriori probabilities, Chow's rule (2) can not be directly applied. The rejection rules proposed in the literature are based on two methods. Both methods exploit the fact that the absolute value of the output of a SVM is proportional to the distance of an input pattern from the class boundary estimated by the classifier. The first method consists in rejecting patterns for which the absolute value of the SVM output is lower than a predefined threshold [6]. The second method consists in mapping the SVM outputs to posterior probabilities, so that Chow's rule can be applied [7,8,9,10]. The mapping is implemented by using sigmoidal-like functions, as usually happens for distance classifiers [11]. We point out that these two methods are equivalent, since the estimates of the a posteriori probabilities obtained using a sigmoidal-like function are monotonic functions of the SVM output. Therefore, both methods reject patterns whose distance from the class boundary is lower than a given threshold.

In this paper, we show that the SRM induction principle suggests a different approach for implementing a cost-sensitive classifier with reject option. This approach consists in embedding the reject option into the learning algorithm. On the basis of this result, and by following Vapnik’s maximum margin approach to the derivation of standard SVMs, we derive a SVM with embedded reject option (section 2). In section 3 we propose a formulation of the modified training problem, and a learning algorithm. In section 4, we report the results of a preliminary experimental comparison between our cost-sensitive SVM with embedded reject option, and the “external” rejection technique proposed in the literature for standard SVMs. Conclusions are drawn in section 5.

2 Cost-sensitive SVMs with embedded reject option

In section 2.1 we summarise the SRM principle, and the derivation of the standard SVM classifier. In section 2.2 we address the problem of classification with reject option under the SRM principle. We show that the SRM principle requires that the rejection region must be determined during the training phase, by the learning algorithm. We then apply this concept to develop a cost-sensitive SVM classifier with embedded reject option.

2.1 The SRM principle and the derivation of standard SVMs

The SRM induction principle was derived from a result of statistical learning theory, consisting in the definition of an upper bound for the expected risk of any given classifier [5]. In statistical learning theory, a classifier is characterised by the set of decision functions it can implement, $f(\mathbf{x}, \alpha)$, where α is a parameter denoting one particular function of the set. For an m -class problem without reject option, decision functions $f(\mathbf{x}, \alpha)$ take on exactly m values, corresponding to the m class labels. Given a loss function $L(\mathbf{x}, y, \alpha)$ (where y denotes the class label of pattern \mathbf{x}), the expected risk $R(\alpha)$ obtained by using any function $f(\mathbf{x}, \alpha)$ is:

$$R(\alpha) = \sum_{j=1}^c \int L(\mathbf{x}, y^j, \alpha) p(\mathbf{x}, y^j) d\mathbf{x} . \quad (3)$$

The corresponding empirical risk, $R_{emp}(\alpha)$, is an approximation of $R(\alpha)$ constructed on the basis of a given sample $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)$:

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^l L(\mathbf{x}_i, y_i, \alpha) . \quad (4)$$

It has been shown that, for any real-valued bounded loss function $0 \leq L(\mathbf{x}, y, \alpha) \leq B$, the following inequality holds true for any function $f(\mathbf{x}, \alpha)$, with probability at least $1 - \eta$:

$$R(\alpha) \leq R_{emp}(\alpha) + \frac{B\varepsilon}{2} \left(1 + \sqrt{1 + \frac{4R_{emp}(\alpha)}{B\varepsilon}} \right), \quad (5)$$

where

$$\varepsilon = 4 \frac{h \left(\ln \frac{2l}{h} + 1 \right) - \ln \eta}{l},$$

and h denotes the VC dimension of the classifier [5]. The SRM principle is aimed at controlling the generalisation capability of a classifier (that is, minimising the expected risk

$R(\alpha)$) by minimising the right-hand side of inequality (5). To this aim, a trade-off between the VC dimension of the classifier and the empirical risk is required. Therefore, training a classifier in the framework of the SRM principle consists in finding the decision function $f(\mathbf{x}, \alpha)$ which provides the best trade-off between the VC dimension and the empirical risk.

The SVM classification technique has been originally derived by applying the SRM principle to a two-class problem, using a classifier implementing linear decision functions:

$$f(\mathbf{x}, \alpha) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b), \quad (6)$$

and using the 0/1 (indicator) loss function [5]:

$$L(\mathbf{x}, y, \alpha) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y, \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y. \end{cases} \quad (7)$$

For linearly separable classes, it has been shown that the VC dimension of the above classifier depends on the margin with which the training samples can be separated without errors. This led to the concept of optimal separating hyperplane (OSH), as the one which separates the two classes with maximum margin [12]. The heuristic extension of the OSH to the general case of not linearly separable classes, was based on the idea of finding the hyperplane which minimises the number of training errors, and separates the remaining correctly classified samples with maximum margin [4]. It has been shown that such an hyperplane can be found by minimising the functional

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l \theta(\xi_i), \quad (8)$$

under the constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l, \\ \xi_i &\geq 0, \quad i = 1, \dots, l, \end{aligned} \quad (9)$$

for sufficiently large values of the constant C , where θ is the step function defined as

$$\theta(u) = \begin{cases} 0, & \text{if } u \leq 0, \\ 1, & \text{if } u > 0. \end{cases}$$

Functional (8) represents a trade-off between the VC dimension (which depends on the margin, defined as $1/\|\mathbf{w}\|$) and the empirical risk (that is, the number of training errors, which is approximated by $\sum_{i=1}^l \theta(\xi_i)$). The trade-off is controlled by the regularisation term C . The above optimisation problem is NP-complete. A computationally tractable approximation was obtained by substituting the step function in (8) with the continuous and convex function $\sum_{i=1}^l \xi_i^\sigma$, $\sigma \geq 1$. Simple quadratic optimisation problems, whose solution is unique and sparse, correspond to the choices $\sigma = 1, 2$ [4]. The OSH can efficiently be found by solving the associated Lagrangian dual problem, which, for $\sigma = 1$, consists in maximising the functional

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j), \quad (10)$$

under the constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0, \\ 0 &\leq \alpha_i \leq C, \quad i = 1, \dots, l, \end{aligned} \quad (11)$$

where the α_i 's are the Lagrange multipliers. The resulting expression of the OSH is:

$$\sum_{i=1}^l y_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}) + b = 0 . \quad (12)$$

It turns out that the above problem has a sparse solution: the only non-zero α_i correspond to misclassified training patterns, or to correctly classified ones, whose distance from the OSH $\mathbf{w} \cdot \mathbf{x} + b$ is less than the margin $1/\|\mathbf{w}\|$. Such patterns are named support vectors. It is worth pointing out that, besides its sparsity and uniqueness, the solution of the above training problem is also characterised by the so-called Karush-Kuhn-Tucker (KKT) necessary and sufficient conditions [4]. The concept of OSH can be extended to non-linear decision surfaces, by using the so-called kernel functions, that is, functions satisfying Mercer's conditions [5]. A non-linear kernel function $K(\mathbf{x}, \mathbf{y})$ implicitly represents the inner product of the images of \mathbf{x} and \mathbf{y} in a new (usually unknown) feature space with higher dimension (even infinite) than the original one. The OSH can be implicitly constructed in the new feature space, by solving problem (), where the inner products $(\mathbf{x}_i \cdot \mathbf{x}_j)$ are replaced by $K(\mathbf{x}_i, \mathbf{x}_j)$. The corresponding decision surface in the original feature space has the same expression (12), obtained by replacing again the terms $(\mathbf{x}_i \cdot \mathbf{x}_j)$ with $K(\mathbf{x}_i, \mathbf{x}_j)$. Therefore the kernel function determines the kind of decision surface in the original feature space.

Although standard optimisation techniques exist to solve quadratic programming problems, they are not efficient for problem (10,11), since they require to store the entire kernel matrix $K(\mathbf{x}_i, \mathbf{x}_j)$ in memory. Therefore, several iterative optimisation algorithms targeted to problem (10,11) have been proposed in the literature, based on different heuristics [13]. In particular, these algorithms exploit the KKT conditions to speed up the convergence, and to implement the stopping criterion.

2.2 SVMs with embedded reject option

Let us now consider the problem of classification with reject option under the SRM principle. In this case, decision functions $f(\mathbf{x}, \alpha)$ take on $m+1$ values, where the $m+1$ st one corresponds to the reject decision. Moreover, loss functions take on at least three values, as seen in section 1. Note that the expressions of the expected risk (3) and of the empirical risk (4) are valid also for classification with reject option. It is now easy to see that the upper bound (5) on the expected risk of a classifier holds also for this kind of decision and loss functions. Indeed, inequality (5) was derived under the only assumption of a bounded real-valued loss function [5]. This means that the SRM principle can be applied also to classification with reject option. The key point is that, according to the definition of classifier training under the SRM principle, given in section 2.1, also the rejection region must be determined during the training phase of the classifier, that is, by the learning algorithm.

Therefore, in order to design a classifier for a cost-sensitive problem with reject option, the SRM principle suggests to implement a cost-sensitive learning algorithm. This would result in a classifier with embedded reject option. In the following, we apply this approach to SVMs classifiers. To this aim, we generalise the classifier characterised by linear decision functions (6), and indicator loss function (7), to make it capable to provide also the rejection decision.

The simplest generalisation of linear decision functions (6) to classification with reject option are functions defined by means of pairs of parallel hyperplanes, so that the rejection region is the space delimited by such hyperplanes. Formally, let us denote a pair of parallel hyperplanes as:

$$\mathbf{w} \cdot \mathbf{x} + b \pm \varepsilon = 0, \quad \varepsilon \geq 0. \quad (13)$$

The corresponding decision function is then defined as follows:

$$\begin{aligned} f(\mathbf{x}, \alpha) &= +1, & \text{if } \mathbf{w} \cdot \mathbf{x} + b \geq \varepsilon, \\ f(\mathbf{x}, \alpha) &= -1, & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq -\varepsilon, \\ f(\mathbf{x}, \alpha) &= 0, & \text{if } -\varepsilon < \mathbf{w} \cdot \mathbf{x} + b < \varepsilon, \end{aligned} \quad (14)$$

where α denotes the parameters \mathbf{w} , b , ε , while the class labels are denoted with $y = +1$ and $y = -1$, and the reject decision is denoted with $y = 0$. Note that the distance between the hyperplanes, that is, the width of the rejection region, is equal to $2\varepsilon / \|\mathbf{w}\|$. Analogously, the simplest extension of the indicator loss function (7) to classification with reject option is the following:

$$L(\mathbf{x}, y, \alpha) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y, \\ c_R, & \text{if } f(\mathbf{x}, \alpha) = 0, \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y \text{ and } f(\mathbf{x}, \alpha) \neq 0. \end{cases} \quad (15)$$

Obviously $0 \leq c_R \leq 1$. The corresponding expected risk is [2]:

$$R(\alpha) = c_R P(\text{reject}) + P(\text{error}), \quad (16)$$

where $P(\text{reject})$ and $P(\text{error})$ denote respectively the misclassification and reject probabilities achieved using the function $f(\mathbf{x}, \alpha)$. Accordingly, the expression of the empirical risk (4), for a given decision function and a given training set, is:

$$R_{emp}(\alpha) = c_R R + E, \quad (17)$$

where R and E denote respectively the misclassification and reject rates achieved by $f(\mathbf{x}, \alpha)$ on training samples. According to the SRM principle, training this classifier consists in finding the pair of parallel hyperplanes (13), which provide the best trade-off between the VC dimension and the empirical risk. Let us call such a pair the optimal separating hyperplanes with reject option (OSHR).

We point out that also using the rejection rules proposed in the literature, the rejection region is delimited by a pair of parallel hyperplanes. However, such hyperplanes are constrained to be always parallel and equidistant to a given hyperplane (the OSH), for any value of the reject rate. Instead, the position and orientation of the OSHR can change for varying values of c_R , as a result of the training phase, since the empirical risk depends on the parameter c_R .

In order to apply the SRM principle to the classifier with reject option defined by linear decision functions (14) and loss function (15), it would be necessary to evaluate its VC dimension h , and to find subsets of decision functions (14) with VC dimension lower than h . Since this is beyond the scope of this work, we propose a definition of the OSHR based on the maximum margin approach followed by Vapnik in the derivation of standard SVMs. We assume that the OSHR can be defined as a pair of parallel hyperplanes (13) which minimise the empirical risk (17), and separate with maximum margin the samples correctly classified and *accepted*. We remind that a pattern \mathbf{x}_i is accepted if $|\mathbf{w} \cdot \mathbf{x}_i + b| \geq \varepsilon$. For a pair of parallel hyperplanes (13), we define the margin of an accepted pattern as its distance from the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$. Under the above assumption, it can be shown that the OSHR is the solution of an optimisation problem similar to that of standard SVMs [14]. In particular, a pair of parallel hyperplanes (13) which minimise the empirical risk (17), and

maximise the margin of samples accepted and correctly classified, can be found by minimising the following functional:

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l h(\xi_i, \varepsilon), \quad (18)$$

under the constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l, \\ \xi_i &\geq 0, \quad i = 1, \dots, l, \\ 0 &\leq \varepsilon \leq 1, \end{aligned} \quad (19)$$

where

$$h(\xi_i, \varepsilon) = c_C \theta(\xi_i) + (c_R - c_C) \theta(\xi_i - 1 + \varepsilon) + (1 - c_R) \theta(\xi_i - 1 - \varepsilon), \quad (20)$$

for sufficiently large C . It can be shown that the functional $\sum_{i=1}^l h(\xi_i, \varepsilon)$ approximates the empirical risk (17), that is, the error-reject trade-off, for sufficiently small c_C , where c_C is a constant term such that $0 < c_C < c_R$ [14].

The above optimisation problem (as problem (8,9) for standard SVMs) is NP-complete. A convex approximation of its objective function (18) would lead to a quadratic programming problem similar to that of standard SVMs. However, a convex approximation of $h(\xi_i, \varepsilon)$ is not feasible, since it would not allow to adequately represent the error-reject trade-off. Unfortunately, a non-convex optimisation problem would not guarantee the main properties of SVMs, namely the uniqueness and sparseness of the solution. Nevertheless, our main goal was to compare the error-reject trade-off achievable by our cost-sensitive SVM classifier with embedded reject option, and by the rejection technique for standard not cost-sensitive SVMs described in section 1. Therefore, we devised a non-convex approximation for functional (18), and developed a specific algorithm for solving the corresponding optimisation problem.

3 Formulation of the training problem

A good non-convex approximation of $h(\xi_i, \varepsilon)$ (20) can be obtained by substituting the step function $\theta(u)$ with a sigmoid function

$$S_\alpha(u) = \frac{1}{1 + e^{-\alpha u}},$$

for sufficiently large values of the constant α . To solve the corresponding optimisation problem, the technique of the Lagrange multipliers can be used. However, the above approximation would lead to a trivial solution of the dual problem: all the Lagrange multipliers would be equal to zero. To avoid this, we introduce in $h(\xi_i, \varepsilon)$ a term equal to $a\xi_i^2$, where a is a constant value. We then obtain:

$$h'(\xi_i, \varepsilon) = c_C S_\alpha(\xi_i) + (c_R - c_C) S_\alpha(\xi_i - 1 + \varepsilon) + (1 - c_R) S_\alpha(\xi_i - 1 - \varepsilon) + a\xi_i^2.$$

For sufficiently small a , the behaviour of $h'(\xi_i, \varepsilon)$ adequately represents the trade-off between errors and rejections as $h(\xi_i, \varepsilon)$ (20) [14]. Note that the introduction of the term $a\xi_i^2$ makes the constraint $\xi_i \geq 0$ unnecessary.

We have therefore approximated the problem of finding the OSHR as follows. Minimise the functional:

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l h'(\xi_i, \varepsilon), \quad (21)$$

under constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l, \\ 0 &\leq \varepsilon \leq 1. \end{aligned} \quad (22)$$

It can be shown [14] that the associated Lagrangian dual problem consists in maximising the following functional:

$$W(\alpha_1, \dots, \alpha_l) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + C \min_{\substack{\xi_i \\ 0 \leq \varepsilon \leq 1}} \sum_{i=1}^l \left(h''(\xi_i, \varepsilon) - \frac{\alpha_i}{C} \xi_i \right), \quad (23)$$

under constraints:

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0, \\ \alpha_i &\geq 0, \quad i = 1, \dots, l. \end{aligned} \quad (24)$$

It turns out that the weight vector \mathbf{w} of the OSHR (13) has the same expansion on training vectors as in standard SVMs:

$$\mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i.$$

This allows to deal with non-linear decision surfaces by using kernel functions. Indeed, both in the objective function (23) and in the expression of the OSHR (13), training points appear only in the form of inner products.

From the computational viewpoint, the drawback of the dual problem above is due to the fact the primal objective function (21) is not convex. This implies that the dual objective function is not known in analytical form, as can be seen in (23). To evaluate the dual objective function for given values of the Lagrange multipliers, another constrained optimisation problem must be solved. Moreover, as pointed out above, the sparsity and the uniqueness of the solution are not guaranteed. Furthermore, no necessary and sufficient conditions exist, analogous to the KKT ones for standard SVMs, to characterise the solution of the primal (21,22) and dual (23,24) problems. Nevertheless, since the objective function (21) of the primal problem is continuous, the objective function of the dual problem (23) is concave, and therefore has no local maxima [15].

We exploited the last characteristic above to develop an algorithm for solving the dual problem (23,24). Our algorithm is derived from the sequential minimal optimisation (SMO) algorithm, developed for standard SVMs [16]. More details about our algorithm can be found in [17]. It is worth noting that our algorithm was not optimised in terms of computational efficiency, since our primary goal was to evaluate the error-reject trade-off achievable by our technique. In its current implementation, our algorithm has a computational cost comparable to that of standard SVMs training algorithms, for training sets up to one thousand patterns.

4 Experimental results

In this section we present the results of preliminary experiments aimed at comparing the error-reject trade-off achievable by our cost-sensitive SVM with embedded reject option, and by standard SVMs with the “external” rejection technique described in section 1. We remark that, when using the loss function (15), 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). Indeed, minimising the expected risk (16) is equivalent to maximise the classification accuracy for any value of the reject probability [2]. The trade-off between errors and rejections depends on the cost of a rejection c_R . This implies that different points of the A-R curve correspond to different values of c_R .

The experiments were carried out with the Letter data set, taken from the University of California at Irvine machine learning database repository (<http://www.ics.uci.edu/~mllearn/MLRepository.html>). It consists of 20,000 patterns representing the 26 capital letters in the English alphabet, based on 20 different fonts. Each pattern is characterised by 16 features. In our experiments, we considered all possible pairs of classes as two-class problems. We focused only on non-linearly separable problems, since these are the most significant for testing the performance of rejection techniques. The non-linearly separable problems are 193 out of 325, as identified in [18]. For each two-class problem, we randomly subdivided the patterns of the corresponding classes in a training set and a test set of equal size.

As explained in section 1, the rejection technique proposed in the literature consists in rejecting the patterns for which the output of a trained standard SVM is lower than a predefined threshold D . To implement this technique, we trained standard SVMs using the software SVM^{light} [19], available at <http://svmlight.joachims.org>. The value of the regularisation parameter C was automatically set by SVM^{light}. In our experiments, we used a linear kernel (that is, a linear decision surface in the original feature space). The A-R curve achievable by using this technique was obtained by computing the values of D which minimise the empirical risk (17), evaluated on the training set, for different values of the rejection cost c_R . The corresponding values of D were then used to classify the test set. We considered values of the reject rate up to 30%, since usually these are the values of interest in practical applications. However, for 115 out of the 193 non-linearly separable problems, only one point of the A-R curve with a rejection rate lower than 30% was found, due to the particular distribution of training samples in the feature space. We considered therefore only the remaining 78 problems, which are reported in Table 1.

To implement our method we used our training algorithm, with a linear kernel, and a value of the C parameter equal to 0.1. The A-R curve was obtained by training a classifier for each different value of c_R . For any given value of w_R , the result of the training phase was a pair of parallel hyperplanes (13), which were used to classify the test set using decision function (14).

The results for the 78 problems of Table 1 can be summarised as follows. For 40 problems (the 51% of the considered problems, reported in the first row of Table 1), our technique achieved on the test set a higher classification accuracy for any value of the reject rate. Four examples are shown in Figure 1 (a)-(d). For 27 problems (the 35% of the considered problems, reported in the second row of Table 1) neither of the two techniques outperformed the other one. Indeed, both techniques exhibited on the test set higher accuracy values for different ranges of the reject rate. Examples are shown in Figure 1 (e),(f). The technique proposed in the literature outperformed our technique only for 12 problems out of 78 (the 14% of the considered problems, reported in the third row of Table 1), as in the example shown in Figure 1 (g).

Table 1. The 78 two-class non-linearly separable problems obtained from the Letter data set and considered in our experiments. Each problem refers to a pair of letters.

AH	BE	BJ	BK	BP	BV	CE	CL	CU	DJ	DO	DX	EK	EX	EZ
FY	GT	HY	IP	JR	JS	KO	KT	LO	OR	OV	PR	PV	RS	TY
UV	XZ	BI	BS	DH	GK	FI	HJ	IZ	KV					
AU	ET	TX	BF	DR	GM	HT	HW	KS	LX	MV	NU	QX	CO	KM
BR	DB	FS	FX	GO	GV	JQ	MU	PQ	PS	PY	RV			
ST	DN	EQ	ER	ES	FT	HK	HU	JZ	KX	SX				

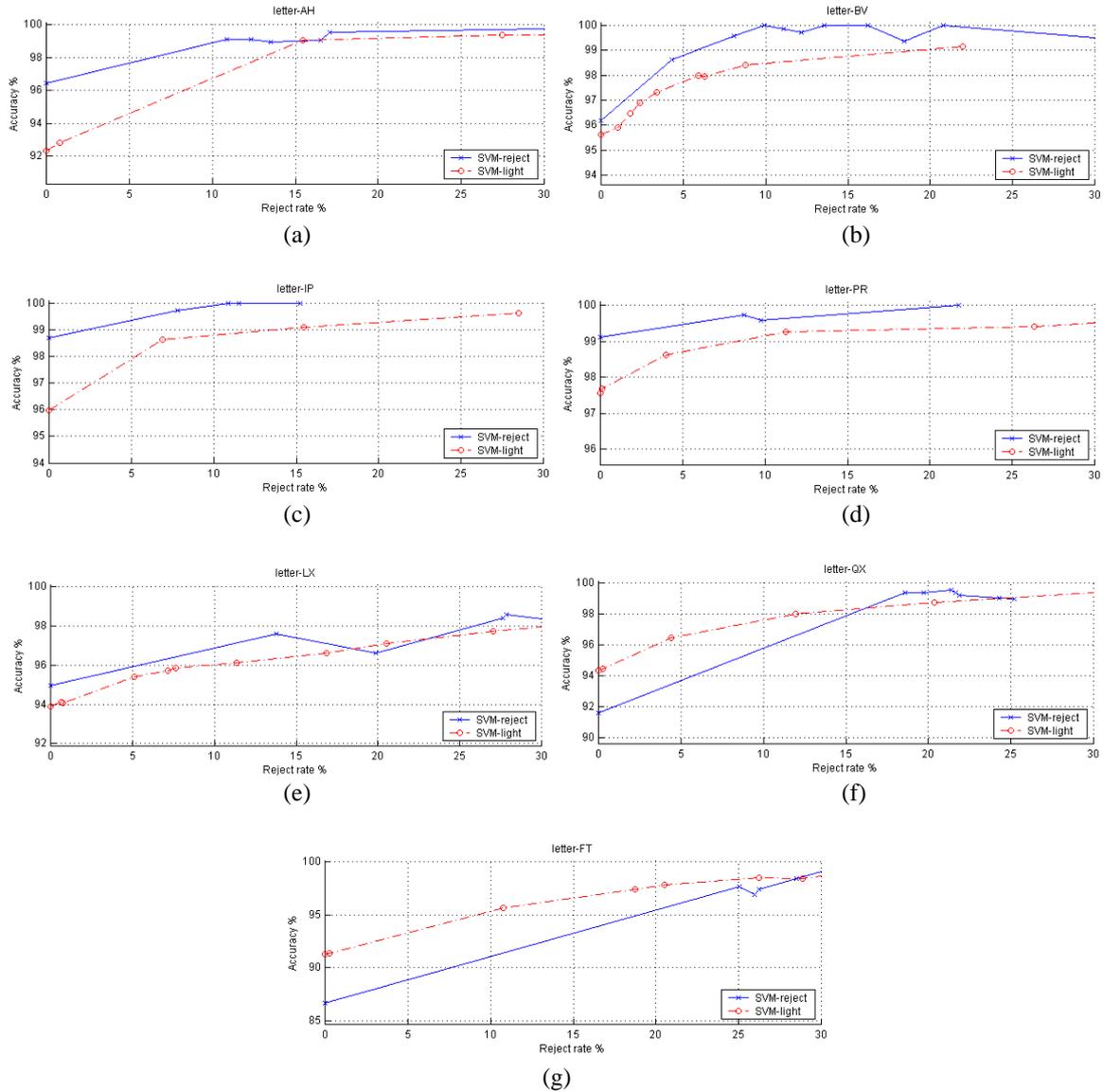


Figure 1. The A-R curves for seven two-class problems are shown. The A-R curves obtained using the proposed method are denoted with *SVM-reject*, while the ones obtained using the rejection technique proposed in the literature are denoted with *SVM-light*.

In the experiments above, our cost-sensitive SVM with embedded reject option allowed achieving a better error-reject trade-off than standard not cost-sensitive SVMs, in the most of cases (51% of the considered problems). As explained in section 2, the rejection region obtained using both methods is delimited by a pair of parallel hyperplanes. However, our cost-sensitive method allows for a greater flexibility in defining their position and

orientation, which can change for different values of the cost of a rejection c_R . The preliminary experimental results reported above seem to prove that this greater flexibility is useful for achieving a better error-reject trade-off. To allow our cost-sensitive learning method to scale to larger training sets, the issues related to its computational cost must be addressed. In particular, the optimisation problem we proposed in section 3 is more complex than the one of standard SVMs, due to the non-convexity of its objective function. Either a different formulation of this problem, or a more efficient algorithm, can make its computational cost comparable to that of algorithms for standard SVMs.

5 Conclusions

In this paper, we proposed a cost-sensitive SVM classifier that directly embeds the reject option. This extension was derived by taking into account a theoretical implication of the SRM induction principle when applied to classification with reject option, and by following Vapnik's maximum margin approach to the derivation of standard SVMs. We devised a novel formulation of the training task as a non-convex optimisation problem, and developed a specific learning algorithm to solve it. We showed that our cost-sensitive learning method allows for a greater flexibility in defining the decision boundaries, with respect to the rejection technique used for standard SVMs. Preliminary experimental results seem to prove that this enhanced flexibility allows achieving a better error-reject trade-off.

References

- [1] C. Elkan, The Foundations of Cost-Sensitive Learning, Proceedings of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI), Seattle, Washington, USA, (2001) 973–978.
- [2] C.K. Chow, On optimum error and reject tradeoff, IEEE Trans. on Information Theory, vol. 16 (1970) 41-46.
- [3] G. Fumera, F. Roli, and G. Giacinto, Reject option with multiple thresholds, Pattern Recognition, vol. 33 (2000) 165-167.
- [4] C. Cortes, and V.N. Vapnik, Support vector networks, Machine Learning, vol. 20 (1995) 1-25.
- [5] V.N. Vapnik, Statistical Learning Theory, Wiley, New York (1998).
- [6] S. Mukherjee, P. Tamayo, D. Slonim, A. Verri, T. Golub, J.P. Mesirov, and T. Poggio, Support vector machine classification of microarray data., Tech. report, Massachusetts Institute of Technology (1998).
- [7] J.C. Platt, Probabilistic outputs for support vector machines and comparison to regularised likelihood methods, in.: Advances in Large Margin Classifiers, A.J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans Eds., Mit Press (1999).
- [8] A. Madevska-Bogdanova, and D. Nikolic, A new approach of modifying SVM outputs, Proceedings of the Int. Joint Conference on Neural Networks (IJCNN'00), Como, Italy, (2000) Vol. 6, 395-398.
- [9] T. Hastie, and R. Tibshirani, Classification by pairwise coupling, Technical Report, Stanford University and University of Toronto (1996).
- [10] J.T.-Y. Kwok, Moderating the outputs of support vector machines, IEEE Transactions on Neural Networks, vol. 10 (1999) 1018-1031.
- [11] R.P.W. Duin, and D.M.J. Tax, Classifier conditional posterior probabilities, in: Advances in Pattern Recognition, A. Amin, D. Dori, P. Pudil, and H. Freeman, Eds., Lecture Notes in Computer Science 1451, Springer, Berlin (1998) 611-619.

- [12] V.N. Vapnik, Estimation of Dependencies Based on Empirical Data, Addendum 1, Springer-Verlag, New York (1982).
- [13] N. Cristianini, and J. Shawe-Taylor, An introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press, Cambridge, UK (2000).
- [14] G. Fumera, and F. Roli, Support Vector Machines with Embedded Reject Option, to appear, Proceedings of the Int. Workshop on Pattern Recognition with Support Vector Machines (SVM2002), Niagara Falls, Canada (2002).
- [15] M.S. Bazaraa, H.D. Sherali, and C.M. Shetty, Nonlinear Programming. Theory and Algorithms, Wiley (1992).
- [16] J.C. Platt, Fast training of support vector machines using sequential minimal optimisation, in: Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C.J.C. Burges, and A.J. Smola Eds., MIT Press (1999).
- [17] G. Fumera, Advanced Methods for Pattern Recognition with Reject Option, Ph.D. Thesis, University of Cagliari, Italy (2002).
- [18] M. Basu, and T.K. Ho, The learning behavior of single neuron classifiers on linearly separable or nonseparable input, Proc. of the Int. Joint Conference on Neural Networks (IJCNN'99), Washington, DC (1999).
- [19] T. Joachims, Making large-scale SVM learning practical, in: Advances in Kernel Methods - Support Vector Learning, B. Schölkopf, C.J.C. Burges, and A.J. Smola Eds., MIT Press (1999) 169-184.