

Support Vector Machines with Embedded Reject Option

Giorgio Fumera, and Fabio Roli

Department of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{fumera, roli}@diee.unica.it

Abstract. In this paper, the problem of implementing the reject option in support vector machines (SVMs) is addressed. We started by observing that methods proposed so far simply apply a reject threshold to the outputs of a trained SVM. We then showed that, under the framework of the structural risk minimisation principle, the rejection region must be determined during the training phase of a classifier. By applying this concept, and by following Vapnik's approach, we developed a maximum margin classifier with reject option. This led us to a SVM whose rejection region is determined during the training phase, that is, a SVM with embedded reject option. To implement such a SVM, we devised a novel formulation of the SVM training problem and developed a specific algorithm to solve it. Preliminary results on a character recognition problem show the advantages of the proposed SVM in terms of the achievable error-reject trade-off.

1 Introduction

The reject option is very useful to safeguard against excessive misclassifications in pattern recognition applications that require high classification reliability. In the framework of the minimum risk theory, Chow defined the optimal classification rule with reject option [1]. In the simplest case where the classification costs do not depend on the classes, Chow's rule consists in rejecting a pattern if its maximum a posteriori probability is lower than a given threshold [2]. The optimality of this rule relies on the exact knowledge of the a posteriori probabilities. However, in practical applications, the a posteriori probabilities are usually unknown [19]. Some classifiers, like neural networks and the k -nearest neighbours classifier, provide approximations of the a posteriori probabilities [3,4,19]. In such case, Chow's rule is commonly used, despite its non-optimality [19]. Other classifiers, like support vector machines (SVMs), do not provide probabilistic outputs. In this case, a rejection technique targeted to the particular classifier must be used.

So far, no work in the literature addressed the problem of defining a specific rejection technique for SVM classifiers. The reject option is currently implemented by using two approaches. The first one uses as measure of classification reliability the distance $d(\mathbf{x})$ of an input pattern \mathbf{x} from the optimal separating hyperplane (OSH), in the feature space induced by the chosen kernel. The rejection rule consists in rejecting

patterns for which $d(\mathbf{x})$ is lower than a predefined threshold [5]. Since the absolute value $|f(\mathbf{x})|$ of the output of a SVM is proportional to $d(\mathbf{x})$, this rule is implemented by applying a reject threshold to $|f(\mathbf{x})|$. The second approach for implementing the reject option in SVMs consists in mapping their outputs to posterior probabilities, so that Chow’s rule can be applied. Usually, for distance classifiers (like the Fisher’s linear discriminant) the mapping is implemented using a sigmoid function [6]. This method was also proposed for SVMs in [7], using the following form for the sigmoid function:

$$P(y = +1 | \mathbf{x}) = \frac{1}{1 + \exp(af(\mathbf{x}) + b)} , \quad (1)$$

where the class labels are denoted as $y = +1, -1$, while a and b are constant terms to be defined on the basis of sample data. A similar method was proposed in [8]. In this case the constants a and b are chosen so that $P(y = +1 | \mathbf{x}) = 0.5$, if $f(\mathbf{x}) > 0$, for patterns lying at a distance $1/\|\mathbf{w}\|$ from the OSH. An approximation of the class-conditional densities $p(f(\mathbf{x}) | y = +1)$ and $p(f(\mathbf{x}) | y = -1)$ with Gaussian densities having the same variance was proposed in [9]. The corresponding estimate of $P(y = +1 | \mathbf{x})$ is again a sigmoid function. A more complex method based on a Bayesian approach, the so-called evidence framework, was proposed in [10]. Nonetheless, also in this case the resulting estimate of $P(y = +1 | \mathbf{x})$ is a sigmoid-like function. We point out that all the mentioned methods provide estimates of the posterior probabilities that are monotonic functions of the output $f(\mathbf{x})$ of a SVM. This implies that Chow’s rule applied to such estimates is equivalent to the rejection rule obtained by directly applying a reject threshold on the absolute value of the output $|f(\mathbf{x})|$. Indeed, both rules provide a rejection region whose boundaries consist of a pair of hyperplanes parallel to the OSH and equidistant from it. The distance of such hyperplanes from the OSH depends on the value of the reject threshold. Accordingly, we can say that all the rejection techniques proposed so far for SVM classifiers consist in rejecting patterns whose distance from the OSH is lower than a predefined threshold.

The above approaches are based on a reasonable assumption, namely, the classification reliability increases for increasing values of the distance of an input pattern from the class boundary constructed by a given classifier. However, this heuristic approach is not coherent with the theoretical foundations of SVMs, which are based on the structural risk minimisation (SRM) induction principle [11]. In this paper, we propose a different approach for introducing the reject option in the framework of SVM classifiers. Our approach is based on the observation that, under the framework of the SRM principle, the rejection region must be determined during the training phase of a classifier. On the basis of this observation, 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 training problem for such a SVM, and a training algorithm. In Section 4, we report the results of a preliminary experimental comparison between our SVM with embedded reject option, and the “external” rejection techniques proposed in the

literature for standard SVMs. The experiments were conducted on a large set of two-class character recognition problems. Conclusions are drawn in Section 5.

2 Support Vector Machines with Reject Option

In Sect. 2.1 we address the problem of classification with reject option under the framework of the SRM principle. It turns out that the SRM principle requires to determine the rejection region during the training phase of a classifier. We then apply this concept to the development of a SVM classifier with embedded reject option. To this aim, we exploit Vapnik's maximum margin approach to the derivation of standard SVMs. In Sect. 2.2 we propose a formulation of the training problem for such a classifier.

2.1 Classification with Reject Option in the Framework of the SRM Principle

The SRM principle was derived from a result of statistical learning theory, consisting in the definition of an upper bound for the expected risk of a given classifier. 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 a c -class problem without reject option, decision functions $f(\mathbf{x}, \alpha)$ take on exactly c values, corresponding to the c 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:

$$R(\alpha) = \sum_{j=1}^c \int L(\mathbf{x}, y^j, \alpha) p(\mathbf{x}, y^j) d\mathbf{x} . \quad (2)$$

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

$$R_{emp}(\alpha) = \frac{1}{l} \sum_{i=1}^l L(\mathbf{x}_i, y_i, \alpha) . \quad (3)$$

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 $1 - \eta$:

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

where

$$\varepsilon = 4 \frac{h \left(\ln \frac{2l}{h} + 1 \right) - \ln \eta}{l} , \quad (5)$$

and h denotes the VC dimension of the classifier [11]. 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 (4). 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.

Consider now the problem of classification with reject option. For a c -class problem, decision functions $f(\mathbf{x}, \alpha)$ take on $c+1$ values: c of them correspond to the c class labels, while the $c+1$ st one corresponds to the reject decision. Moreover, loss functions take on at least three values: correct classification, misclassification, and rejection. By the way, note that the expressions of the expected risk (2) and of the empirical risk (3) are valid also for classification with reject option. It is now easy to see that the upper bound (4) on the expected risk of a classifier holds also for this kind of decision and loss functions. Indeed, inequality (4) was derived under the only assumption of a bounded real-valued loss function [11]. This means that the SRM principle can be also applied to classification with reject option. We point out that, according to the above definition of classifier training under the SRM principle, the rejection region should be determined during the training phase of the classifier, besides the c decision regions.

On the basis of the above discussion, let us now address the problem of constructing a classifier with reject option by using the SRM principle, as an extension of the SVM classifier. 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 [11]:

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

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:

$$\mathbf{w} \cdot \mathbf{x} + b \pm \varepsilon = 0, \quad \varepsilon \geq 0 . \quad (8)$$

The corresponding decision function is then defined as follows:

$$\begin{aligned}
f(\mathbf{x}, \alpha) &= +1, & \text{if } \mathbf{w} \cdot \mathbf{x} + b \geq \varepsilon, \\
f(\mathbf{x}, \alpha) &= -1, & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq -\varepsilon, \\
f(\mathbf{x}, \alpha) &= 0, & \text{if } -\varepsilon < \mathbf{w} \cdot \mathbf{x} + b < \varepsilon,
\end{aligned} \tag{9}$$

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 $y = 0$. 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 loss function:

$$L(\mathbf{x}, y, \alpha) = \begin{cases} 0, & \text{if } f(\mathbf{x}, \alpha) = y, \\ w_R, & \text{if } f(\mathbf{x}, \alpha) = 0, \\ 1, & \text{if } f(\mathbf{x}, \alpha) \neq y \text{ and } f(\mathbf{x}, \alpha) \neq 0, \end{cases} \tag{10}$$

where w_R denotes the cost of a rejection. Obviously $0 \leq w_R \leq 1$. The corresponding expected risk is [2]:

$$R(\alpha) = w_R P(\text{reject}) + P(\text{error}), \tag{11}$$

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 (3), for a given decision function and a given training set, is:

$$R_{emp}(\alpha) = w_R R + E, \tag{12}$$

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 (8), 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).

As pointed out in Sect. 1, also using the rejection rules proposed in the literature the rejection region is delimited by a pair of parallel hyperplanes. Note however that such hyperplanes are constrained to be always parallel and equidistant to a given hyperplane (the OSH), for any value of the reject rate. Instead, since the empirical risk depends on the parameter w_R , the position and orientation of the OSHR can change for varying values of the parameter w_R , as a result of the training phase.

In order to apply the SRM principle to the classifier with reject option defined by linear decision functions (9) and loss function (10), it would be necessary to evaluate its VC dimension h , and to find subsets of decision functions (9) with VC dimension lower than h . Since this was beyond the scope of our work, we propose an operative definition of the OSHR, based on Vapnik's maximum margin approach to the derivation of standard SVMs. Our approach is suggested by the similarity between the classifier defined by (9,10), and the one without reject option defined by linear decision functions (6) and indicator loss function (7). For this last classifier, it has been shown

that, for linearly separable classes, the VC dimension depends on the margin with which the training samples can be separated without errors. This suggested the concept of optimal separating hyperplane as the one which separates the two classes with maximum margin [12]. The extension of the concept of OSH to the general case of non-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 [13].

By analogy, we assume that the OSHR can be defined as a pair of parallel hyperplanes (8) which minimise the empirical risk (12), 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 (8), we define the margin of an accepted pattern as its distance from the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$.

2.2 The Optimal Separating Hyperplanes with Reject Option

In this section we show that the OSHR, as defined on the basis of the above assumption, is the solution of an optimisation problem similar to that of standard SVMs. To this aim, we first recall how the optimisation problem for standard SVMs was obtained.

As said above, the OSH for non-linearly separable classes was defined as the hyperplane which minimises the number of training errors, and separates the remaining correctly classified samples with maximum margin [13]. It was shown that the number of training errors can be minimised by minimising the functional

$$\sum_{i=1}^l \theta(\xi_i), \quad (13)$$

under the constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l, \\ \xi_i &\geq 0, \quad i = 1, \dots, l, \end{aligned} \quad (14)$$

where θ is the step function defined as

$$\theta(u) = \begin{cases} 0, & \text{if } u \leq 0, \\ 1, & \text{if } u > 0. \end{cases} \quad (15)$$

Note that a training error is defined as a pattern \mathbf{x}_i for which $\xi_i > 0$. An hyperplane which simultaneously separates with maximum margin the correctly classified samples, can be found by minimising under constraints (14) the following functional, for sufficiently large values of the constant C :

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + CF \left(\sum_{i=1}^l \theta(\xi_i) \right), \quad (16)$$

where $F(u)$ is a monotonic convex function.

Let us now consider the problem of finding the OSHR as defined at the end of Sect. 2.1. First, a pair of parallel hyperplanes (8) which minimises the empirical risk (12) can be obtained by minimising the following functional, analogous to (13):

$$\sum_{i=1}^l h(\xi_i, \varepsilon) = \sum_{i=1}^l w_R \theta(\xi_i - 1 + \varepsilon) + (1 - w_R) \theta(\xi_i - 1 - \varepsilon), \quad (17)$$

under the constraints

$$\begin{aligned} y_i(\mathbf{w} \cdot \mathbf{x}_i + b) &\geq 1 - \xi_i, \quad i = 1, \dots, l, \\ \xi_i &\geq 0, \quad i = 1, \dots, l, \\ 0 &\leq \varepsilon \leq 1. \end{aligned} \quad (18)$$

Indeed, consider any given pair of parallel hyperplanes (8). It is easy to see that the corresponding values of ξ_i and ε for which constraints (18) are satisfied and functional (17) is minimised, make this functional equal to the empirical risk (12). This can be explained by looking at Fig. 1, where the behaviour of $h(\xi_i, \varepsilon)$ is shown. Constraints (18) imply that, if a pattern \mathbf{x}_i is accepted and correctly classified by the decision function (9), the minimum of the corresponding term $h(\xi_i, \varepsilon)$ in (17) is achieved for $0 \leq \xi_i \leq 1 - \varepsilon$. This means that $h(\xi_i, \varepsilon) = 0$, according to loss function (10). Analogously, if \mathbf{x}_i is rejected, then $1 - \varepsilon < \xi_i \leq 1 + \varepsilon$, and $h(\xi_i, \varepsilon) = w_R$. If \mathbf{x}_i is accepted and misclassified, then $\xi_i > 1 + \varepsilon$, and $h(\xi_i, \varepsilon) = 1$. Therefore, minimising functional (17) under constraints (18) is equivalent to minimise the empirical risk (12).

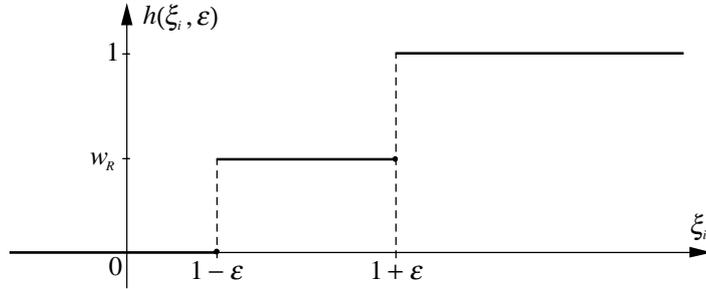


Fig. 1. The behaviour of function $h(\xi_i, \varepsilon)$ is shown, for $\varepsilon = 0.5$ and $w_R = 0.5$.

In order to simultaneously maximise the margin of samples accepted and correctly classified, it is necessary to use a function $h'(\xi_i, \varepsilon)$ slightly different than the $h(\xi_i, \varepsilon)$ defined above:

$$h'(\xi_i, \varepsilon) = w_c \theta(\xi_i) + (w_R - w_c) \theta(\xi_i - 1 + \varepsilon) + (1 - w_R) \theta(\xi_i - 1 - \varepsilon), \quad (19)$$

where w_c is a constant term such that $0 < w_c < w_R$. The only difference with $h(\xi_i, \varepsilon)$ is that $h'(\xi_i, \varepsilon)$ gives a non-null cost w_c to patterns for which $0 < \xi_i \leq 1 - \varepsilon$ (see also Fig. 1). It is easy to see that these are accepted and correctly classified patterns which lie at a distance less than $1/\|\mathbf{w}\|$ to the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$. The corresponding value of $\sum_{i=1}^l h'(\xi_i, \varepsilon)$ is then an upper bound for the empirical risk (12). Note that also functional (13) gives a non-null cost to patterns correctly classified which lie at a distance less than $1/\|\mathbf{w}\|$ to the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$.

Now the problem of finding the OSHR can be formulated as follows. Minimise the functional:

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l h'(\xi_i, \varepsilon) , \quad (20)$$

under constraints (18). For sufficiently large C and sufficiently small w_R the solution of the above problem is a pair of hyperplanes (8) which minimises the empirical risk (12), and separates the samples accepted and correctly classified with maximum margin.

Let us now consider the computational issues connected to the optimisation problems for the OSH (16,14) and the OSHR (20,18). Both problems are NP-complete. A computationally tractable approximation of problem (16,14) for the OSH was obtained by substituting the step function in (16) with the continuous and convex function $\sum_{i=1}^l \xi_i^\sigma$, $\sigma \geq 1$. Simple quadratic optimisation problems correspond to the choices $F(u) = u$, and $\sigma = 1, 2$. In particular, the solution of the corresponding problems is unique [13]. Unfortunately, a convex approximation of the objective function (20) for the OSHR seems not feasible. Indeed, a convex approximation of the function $h'(\xi_i, \varepsilon)$ does not allow to adequately represent the empirical risk (12), that is, the trade-off between errors and rejections. Obviously, using a non-convex approximation, the uniqueness of the solution of the corresponding problem would not be guaranteed. Moreover, a non-convex optimisation problem might not exhibit one of the main properties of SVMs, namely the sparseness of the solution.

Nevertheless, to compare the error-reject trade-off achievable by the SVM-like classifier with reject option defined in this section, and by the rejection techniques for standard SVMs described in Sect. 1, we devised a non-convex approximation for functional (20), described in Sect. 3. We then developed a specific algorithm for solving the corresponding optimisation problem.

3 Formulation of the Training Problem

A good non-convex approximation of $h'(\xi_i, \varepsilon)$ (19) can be obtained by substituting the step function $\theta(u)$ with a sigmoid function

$$S_\alpha(u) = \frac{1}{1 + e^{-\alpha u}}, \quad (21)$$

for sufficiently large values of the constant α . To solve the corresponding optimisation problem, the technique of the lagrangian dual problem can be used. However, the above approximation would lead to a trivial solution of the dual problem, namely 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:

$$h''(\xi_i, \varepsilon) = w_C S_\alpha(\xi_i) + (w_R - w_C) S_\alpha(\xi_i - 1 + \varepsilon) + (1 - w_R) S_\alpha(\xi_i - 1 - \varepsilon) + a\xi_i^2. \quad (22)$$

For sufficiently small a , the behaviour of $h''(\xi_i, \varepsilon)$ adequately represents the trade-off between errors and rejections as $h'(\xi_i, \varepsilon)$ (see Figs. 1 and 2).

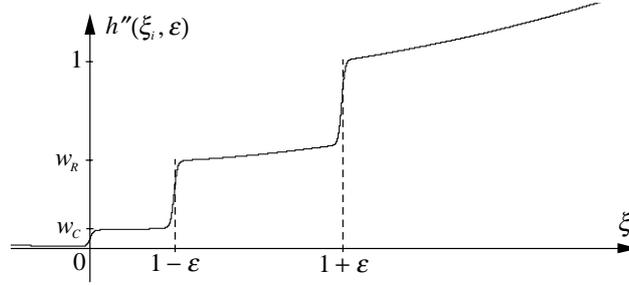


Fig. 2. The behaviour of function $h''(\xi_i, \varepsilon)$ is shown, for $\varepsilon = 0.5$ and $w_R = 0.5$, as in Fig. 1, $w_C = 0.1$, $\alpha = 100$ and $a = 0.05$.

Note that the introduction of the term $a\xi_i^2$ makes the constraint $\xi_i \geq 0$ not necessary.

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

$$\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l h''(\xi_i, \varepsilon), \quad (23)$$

under constraints

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = 1, \dots, l, \quad (24)$$

$$0 \leq \varepsilon \leq 1.$$

Let us now consider the lagrangian dual problem. The corresponding Lagrange function, leaving out the constraints $0 \leq \varepsilon \leq 1$, is:

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \varepsilon; \boldsymbol{\alpha}) = \frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^l h''(\xi_i, \varepsilon) - \sum_{i=1}^l \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i]. \quad (25)$$

The solution of problem (23,24) can be found by minimising the Lagrange function with respect to \mathbf{w} , b , ξ and ε , under constraints $0 \leq \varepsilon \leq 1$, and then maximising it with respect to the non-negative Lagrange multipliers α [14]. Note that the Lagrange function is the sum of a convex function of \mathbf{w} and b , and a non-convex function of ξ and ε . Accordingly, its minimum with respect to \mathbf{w} and b can be found by imposing stationarity:

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \xi, \varepsilon; \alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i = 0, \quad \mathbf{w} = \sum_{i=1}^l \alpha_i y_i \mathbf{x}_i, \\ \frac{\partial L(\mathbf{w}, b, \xi, \varepsilon; \alpha)}{\partial b} &= -\sum_{i=1}^l \alpha_i y_i = 0. \end{aligned} \quad (26)$$

We point out that the first of the above equations implies that the weight vector \mathbf{w} has the same expansion on training vectors as in standard SVMs. The minimum of the Lagrange function with respect to ξ and ε , under the constraints $0 \leq \varepsilon \leq 1$, can not be found analytically. This implies that the dual objective function is not known in analytical form. Substituting back the relations (26) into the Lagrangian, we obtain the following expression for the dual objective function:

$$W(\alpha_1, \dots, \alpha_l) = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) + C \min_{\substack{\xi_i \\ 0 \leq \varepsilon \leq 1}} \sum_{i=1}^l \left(h''(\xi_i, \varepsilon) - \frac{\alpha_i}{C} \xi_i \right) \quad (27)$$

The dual problem consists then in maximising the above function, under constraints

$$\begin{aligned} \sum_{i=1}^l y_i \alpha_i &= 0, \\ \alpha_i &\geq 0, \quad i = 1, \dots, l. \end{aligned} \quad (28)$$

The dual problem is similar to the one of standard SVMs. Note that it allows to deal with non-linear decision surfaces by using kernels. Indeed, both in the objective function (27) and in the expression of the weight vector \mathbf{w} (26), training points appear only in the form of inner products. Formally, the differences are the additional term

$$C \min_{\substack{\xi_i \\ 0 \leq \varepsilon \leq 1}} \sum_{i=1}^l \left(h''(\xi_i, \varepsilon) - \frac{\alpha_i}{C} \xi_i \right) \quad (29)$$

in the objective function (27), and the absence of the constraints $\alpha_i \leq C$, $i = 1, \dots, l$. From the computational viewpoint, the drawbacks of the above dual problem are due to the fact that the primal objective function is not convex. As pointed out above, this implies that the dual objective function (27) is not known in analytical form. To evaluate it for given values of the Lagrange multipliers α , the constrained optimisation problem (29) must be solved. Moreover, the sparsity and the uniqueness of the solution are not guaranteed. Furthermore, no necessary and sufficient conditions exist, analogous to the Karush-Kuhn-Tucker ones for standard SVMs, to characterise the solution of the primal (23,24) and dual (27,28) problems. Nevertheless, since the

objective function (23) of the primal problem is continuous, the objective function of the dual problem (27) is concave, and therefore has no local maxima [14].

We exploited the last characteristic above to develop an algorithm for solving the dual problem (27,28). Our algorithm is derived from the sequential minimal optimisation (SMO) algorithm, developed for standard SVMs [15]. Basically, SMO iteratively maximises the dual objective function of the standard SVMs dual problem by updating at each step only two lagrange multipliers, while enforcing the constraints. It is worth noting that the corresponding optimisation problem can be solved analytically. The choice of the two multipliers at each step is determined by a heuristic, and the KKT conditions are used as stopping criterion.

Since the objective function of problem (27,28) is concave, although it is not known in analytical form, it can be maximised using the same iterative strategy of SMO. It is easy to see that constraints (28) enforce any pair of multipliers to lie on a line segment, or on a half-line. The maximum of the concave objective function with respect to a given pair of multipliers can then be found by using the golden section method. To evaluate the objective function, a specific algorithm was developed for solving the optimisation problem (29). To select a pair of multipliers at each iteration, and to implement a stopping criterion, specific heuristics were used, which exploit the characteristics of problem (27,28). More details about our algorithm can be found in [16].

4 Experimental Results

The aim of our experiments was to compare the performance of our SVM with embedded reject option, with that of standard SVMs with the “external” reject technique described in Sect. 1. We remark that, when using the loss function (10), 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). It has been shown that minimising the expected risk (11) 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 w_R . This implies that different points of the A-R curve correspond to different values of w_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/~mlern/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 [17]. 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 Sect. 1, the main rejection technique proposed in the literature consists in training a standard SVM, and rejecting the patterns \mathbf{x} for which $|f(\mathbf{x})| < D$, where D is a predefined threshold. To implement this technique, we trained SVMs using the software SVM^{light} [18], available at <http://svmlight.joachims.org>. The value of the parameter C was automatically set by SVM^{light}. In our experiments, we used a linear kernel. The A-R curve achievable using this technique was obtained by computing the values of D which minimise the empirical risk (12), evaluated on the training set, for different values of the rejection cost w_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 the training algorithm summarised in Sect. 3, 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 w_R . For any given value of w_R , the result of the training phase was a pair of parallel hyperplanes (8), which were used to classify the test set using decision function (9). Our algorithm took an average time of about five minutes to carry out a training phase, for about 800 training patterns, on a PIII 800 workstation running a Linux OS. SVM^{light} took about one minute. However, it is worth noting that the algorithm we used was not optimised in terms of computational efficiency.

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				

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 Fig. 3(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 Fig. 3(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 Fig. 3(g).

The above results show that our SVM embedding reject option allows achieving a better error-reject trade-off than using standard SVMs in the most of cases (51% of the considered problems). For the 35% of the considered two-class problems, there was no clear winner. The superiority of one method over the other depends on the reject rate required. As pointed out in Sect. 2, the rejection region obtained using both methods is delimited by a pair of parallel hyperplanes. However, our method allows for a greater flexibility in defining their position and orientation, which can change for different values of the cost of a rejection w_R . The preliminary experimental results reported above seem to prove that this greater flexibility is useful to achieve a better error-reject trade-off.

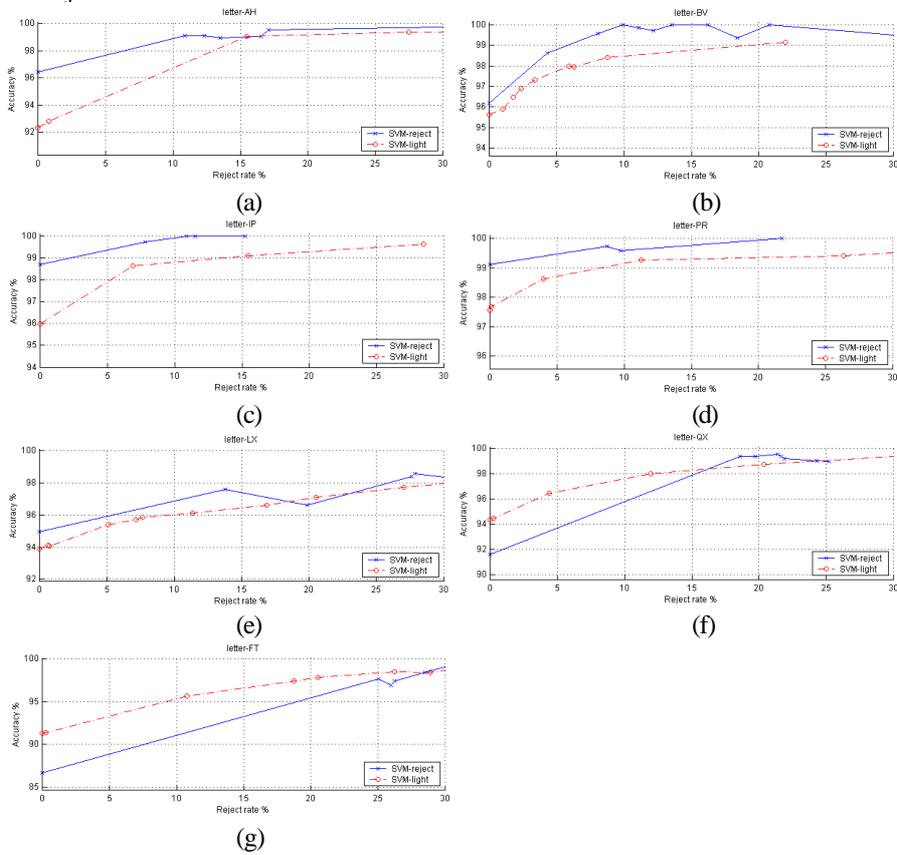


Fig. 3. The A-R curves for seven two-class problems are shown. The A-R curves obtained using the proposed method are denoted with *SVM-reject*, while the ones obtained using the rejection technique proposed in the literature are denoted with *SVM-light*.

To effectively exploit the advantages of our method in terms of the achievable error-reject trade-off, even for problems with larger data sets, the issues related to its computational cost must be addressed. In particular, the optimisation problem we proposed in Sect. 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 an extension of SVMs that directly embeds reject option. This extension was derived by taking into account a theoretical implication of the SRM 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 algorithm to solve it. A pair of parallel hyperplanes delimits the rejection region provided by our SVM. The same holds for the rejection region provided by the commonly used rejection technique. However, our method allows for a greater flexibility in defining the position and orientation of such hyperplanes, which can change for different values of the cost of rejections w_R . The experimental results show that this enhanced flexibility allows achieving a better error-reject trade-off.

On the basis of these results, further work should focus on defining a formulation of the training problem with lower computational complexity, and on developing an efficient optimisation algorithm for solving it.

References

1. Chow, C.K.: An optimum character recognition system using decision functions. IRE Trans. on Electronic Computers 6 (1957) 247-254
2. Chow, C.K.: On optimum error and reject tradeoff. IEEE Trans. on Information Theory 16 (1970) 41-46
3. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press (1995)
4. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M., Suter, B.: The multilayer perceptron as an approximation to a Bayes optimal discriminant function. IEEE Trans. on Neural Networks 4 (1990) 296-298
5. Mukherjee, S., Tamayo, P., Slonim, D., Verri, A., Golub, T., Mesirov, J.P., Poggio, T.: Support vector machine classification of microarray data. Tech. report. Massachusetts Institute of Technology (1998)
6. Duin, R.P.W., Tax, D.M.J.: Classifier conditional posterior probabilities. In: Amin, A., Dori, D., Pudil, P., Freeman, H. (eds.): Advances in Pattern Recognition. Lecture Notes in Computer Science 1451, Springer, Berlin (1998) 611-619
7. Platt, J.C.: Probabilistic outputs for support vector machines and comparison to regularised likelihood methods. In: Smola, A.J., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.): Advances in Large Margin Classifiers. Mit Press (1999)
8. Madevska-Bogdanova, A., Nikolic, D.: A new approach on modifying SVM outputs. Proc. of the IEEE-INNS-ENNS Int. Joint Conference on Neural Networks, Vol. 6 (2000) 395-398

9. Hastie, T., Tibshirani, R.: Classification by pairwise coupling. Technical Report. Stanford University and University of Toronto (1996)
10. Kwok, J.T.-Y.: Moderating the outputs of support vector machines. *IEEE Transactions on Neural Networks* 10 (1999) 1018-1031
11. Vapnik, V.N.: *Statistical Learning Theory*. Wiley, New York (1998)
12. Vapnik, V.N.: *Estimation of Dependencies Based on Empirical Data*, Addendum 1. Springer-Verlag, New York (1982)
13. Cortes, C., Vapnik, V.N.: Support vector networks. *Machine Learning* 20 (1995) 1-25
14. Bazaraa, M.S., Sherali, H.D., Shetty, C.M. *Nonlinear Programming. Theory and Algorithms*. Wiley (1992)
15. Platt, J.C.: Fast training of support vector machines using sequential minimal optimisation. In: B. Schölkopf, Burges, C.J.C., Smola, A.J. (eds.): *Advances in Kernel Methods - Support Vector Learning*. MIT Press (1999)
16. Fumera, G.: *Advanced Methods for Pattern Recognition with Reject Option*. Ph.D. Thesis. University of Cagliari (2002)
17. Basu, M., Ho, T.K. The learning behavior of single neuron classifiers on linearly separable or nonseparable input. *Proc. of the 1999 Int. Joint Conference on Neural Networks*. Washington, DC (1999)
18. Joachims, T.: Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C.J.C., Smola, A.J. (eds.): *Advances in Kernel Methods - Support Vector Learning*. MIT Press (1999) 169-184
19. Fumera, G., Roli, F., Giacinto, G.: Reject option with multiple thresholds. *Pattern Recognition* 33 (2000) 165-167