Combination of One-Class Support Vector Machines for Classification with Reject Option

Blaise Hanczar, Michèle Sebag

Combination of One-Class Support Vector Machines for Classification with Reject Option

Blaise Hanczar[1,3] and Michèle Sebag[2,3]

[1] CNRS, LRI UMR 8623, Universite Paris-Sud, Orsay, France
[2] TAO Project-team, INRIA Saclay & LRI, Universite Paris-Sud, Orsay, France
[3] LIPADE, Universite Paris Descartes, Paris, France
blaise.hanczar@parisdescartes.fr

Abstract. This paper focuses on binary classification with reject option, enabling the classifier to detect and abstain hazardous decisions. While reject classification produces in more reliable decisions, there is a tradeoff between accuracy and rejection rate. Two type of rejection are considered: ambiguity and outlier rejection. The state of the art mostly handles ambiguity rejection and ignored outlier rejection. The proposed approach, referred as CONSUM, handles both ambiguity and outliers detection. Our method is based on a quadratic constrained optimization formulation, combining one-class support vector machines. An adaptation of the sequential minimal optimization algorithm is proposed to solve the minimization problem. The experimental study on both artificial and real world datasets exams the sensitivity of the CONSUM with respect to the hyper-parameters and demonstrates the superiority of our approach.

Keywords: Supervised classification, Rejection option, Abstaining classifier, Support vector machines.

1 Introduction

One of the most interesting exploitation of the data is the construction of predictive classifiers [9]. For example, in genetic and molecular medicine, gene expression profiles can be used to differentiate different types of tumors with different outcomes and thus assist MD in the selection of an adapted therapeutic treatment if appropriate [20]. A huge number of methods from pattern recognition and machine learning have been developed and deployed on various domains. However, even though these methods produce classifiers with a good accuracy, these are often still insufficiently accurate to be used routinely. For example, a diagnostic or a choice of therapeutic strategy must be based on a very high confidence classifier; an error of the predictive model may lead to tragic consequences. A way of improving the reliability of such classifier is to use *abstaining classifiers* [12] also called *reject classifier* [19] or *selective classifier* [4]. Unlike standard classifiers that associate a predicted label to each example, only a subset of the examples are assigned to a class. Reject classifiers define a rejection

region including the examples of which confidence is low [1,18,13,7,11]. While reject classifiers have a higher accuracy than the standard classifiers, there is a trade-off between accuracy and rejection rate [8]. The higher the classifier accuracy, the higher the rejection rate.

A contribution of this paper is to investigate and handle two types of rejection: the ambiguity rejection and the outlier rejection. The ambiguity rejection corresponds to the cases where an example is close to several classes, we cannot decide between these classes with a high confidence. The ambiguity rejection region is generally a small region containing the class boundaries. The outlier rejection corresponds to the cases where an example is far from all classes. The outlier rejection region is a large region surrounding all classes. Let us to illustrate the difference between these two types of rejection. Assumes that a hospital use a classifier to identify the lymphoblastic from the myelogenous leukemia of patients suffering from acute leukemia. A classifier gives a probability of 0.49 for lymphoblastic and 0.51 for myelogenous for a given patient. Although, the probability of myelogenous is the highest, this class cannot be assigned to the patient, the difference of probability is too low. This patient must be rejected by the classifier because no reliable diagnosis can be done. It is an ambiguity rejection. Let another patient file be far from the distribution of both lymphoblastic and myelogenous. The patient is considered as an outlier and must be also rejected. It is likely that this patient has not an acute leukemia and should pass tests for other types of leukemia. It is a outlier rejection.

In this paper we propose a new approach of classification with reject option that defines both the outlier and ambiguity rejection regions. Section 2 introduces the formal background and the state of the art. Section 3 is an overview of CONSUM, together with the appropriate optimization algorithm for scalability on large datasets. Experiments are reported and discussed in section 4 shows. Conclusion and perspectives for future researches are given in section 5.

2 Classification with Reject Option

Let a binary classification problem with a training set $T = \{(x_1, y_1), ..., (x_N, y_N)\}$ where $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. A reject classifier is a function that returns a class for each example $\Psi : \mathbb{R}^d \rightarrow \{-1, +1, R\}$, R represents the reject class including two subclasses R_a and R_d for ambiguity and outlier rejection. An example can be positive, negative or an outlier (belongs to none of the two classes). When we use a reject classifier on test examples, 12 different classification results are possible (three actual classes $\times$ four predicted classes). To each of these cases a cost is associated (table 1). The cost of a good classification is zero. λ_{FP} and λ_{FN} stand for the cost for false positive and false negative. λ_{ON} and λ_{OP} are the costs for assigning respectively the positive or negative class to an outlier, usually we set $\lambda_{ON} = \lambda_{FN}$ and $\lambda_{OP} = \lambda_{FP}$. Finally λ_{Ra} is the cost of ambiguity rejection. λ_{Rd} is the cost for rejecting a positive or negative example as an outlier. Usually the costs of ambiguity and outlier rejection are equal $\lambda_{Ra} = \lambda_{Rd}$. Note that, in principle, classifying the ambiguity rejection class to an outlier is

Table 1. Cost matrix of a two classes classification problem with ambiguity and outlier reject option

	actual class		
	+1	**-1**	**O**
+1	0	λ_{FP}	λ_{OP}
-1	λ_{FN}	0	λ_{ON}
Ra	λ_{Ra}	λ_{Ra}	0
Rd	λ_{Rd}	λ_{Rd}	0

an error, but it has no impact on the classifier performance, this cost is therefore set to zero. Most of reject classification methods only consider the ambiguity rejection. In this case the cost matrix in table 1 is reduced to its first three rows and first two columns.

Classifiers with rejection involve two main approaches: plug-in and embedded methods. The most popular approach consists to add a rejection rule to a classifier without rejection. These methods are called plug-in methods [1]. Two thresholds t_N and t_P are defined on the output of the classifier $f(x)$:

$$\Psi(x) = \begin{cases} -1 & \text{if} \quad f(x) \leq t_N \\ +1 & \text{if} \quad f(x) \geq t_P \\ R & \text{if} \quad t_N < f(x) < t_P \end{cases} \tag{1}$$

Chow has introduced the notion of abstaining classifier and his definition of the rejection region is based on the exact posterior probabilities [1]. The thresholds defining the optimal abstaining region are computed directly from the cost matrix. In practice, the exact posterior probabilities are not available since the class distribution is unknown. The Chow's rule must thus be used with an estimation of the posterior probabilities. To overcome the need for the exact posterior probabilities, Tortorella has proposed a method where the abstaining region is computed in selecting two points on the Receiver operating characteristic (ROC) curve describing the performance of the classifier[18]. The two points are identified by their tangent on the ROC curve computed from the cost of rejection and type of error. Note that, Santos-Pereira proved that both the Chow's rule and ROC rule are actually equivalent [15].

Unlike the plug-in methods, the embedded methods compute the rejection region during the learning process. It has been proved that in theory the embedded methods give better classifiers than plug'in rule [3]. Fumera and Roli replace the Hinge loss function of the SVM by a kind of step function where the steps represent the cost of good classification, rejection and miss-classification [5]. The main drawback of this approach is the difficulty of the optimization problem because of the non-convexity of the loss function. Since the natural loss function of reject classification is non-convex, Yuan and Wegkamp proposed to use a surrogate convex loss [22]. They show that these functions are infinite sample consistent (they share the same minimizer) with the original loss function in some conditions.

Grandvalet et al. have proposed to use a double Hinge loss function in a SVM that has the advantage to be convex and linear piece-wise [6].

Dubuisson and Masson introduced the notion of outlier rejection [2]. They defined in a parametric case a threshold on the likelihood in order to reject examples far from the centers of all classes. Landgrebe et al. studied the interaction between error rate, ambiguity and outlier rejection. They optimize the thresholds of the classifiers in plotting the 3D ROC surface and computing the volume under the surface [10].

3 Combination of Two One-Class Support Vector Machines

3.1 Formal Description

Our method involve two interdependent one-class support vector machines, one for each class. The aim of the one-class SVM is to construct the smallest region capturing most of the training examples [16] and is defined by the following optimization problem:

$$\begin{aligned} &min_{w,\rho,\xi_i} \frac{1}{2}||w||^2 - \rho + \frac{1}{VN}\sum_{i=1}^{N}\xi_i \\ &subject\ to \qquad \langle w, x_i\rangle \geq \rho - \xi_i \end{aligned}$$

The hyperplan, defined by w and ρ, separates the training examples from the origin with maximum margin into the feature space. All examples x such that $f(x) = \langle w, \phi(x)\rangle - \rho \geq 0$ are in the one-class SVM, if $f(x) < 0$ then x is out from the one-class SVM. The slack variables ξ_i's represent the empirical loss associated with x_i. $V \in [0, 1]$ represents the rate of outliers and controls the trade-off between the size of the one-class SVM and the number of training example outside from the one-class SVM. Without any loss of generality, we consider that the training set is labeled as follow: $y_i = +1$ for $i \in [1, p]$ and $y_i = -1$ for $i \in [p + 1, N]$. Moreover we assume that the cost of false positive and false negative are equal $\lambda_{FP} = \lambda_{FN} = \lambda_E$, $\lambda_{ON} = \lambda_{OP} = \lambda_E$ and $\lambda_{Ra} = \lambda_{Rd} = \lambda_R$.

Our method, called CONSUM, is based on the combination of two interdependent one-class SVM. The coupling of the one-class SVM allows a robust estimation of the rejection regions. The intuition behind this model is illustrated by the figure 1. It minimizes the following function:

$$\begin{aligned} L = &\frac{1}{2}||w_+||^2 - \rho_+ + \frac{1}{V_+N_+}\sum_{i=0}^{p}\xi_i + \frac{1}{2}||w_-||^2 - \rho_- + \frac{1}{V_-N_-}\sum_{i=p+1}^{N}\eta_i \\ &+\frac{C}{N}\left(C_r\sum_{i=1}^{N}\theta_i + C_e\sum_{i=1}^{N}\varepsilon_i\right) \end{aligned} \tag{2}$$

$$subject\ to \begin{cases} \langle w_+, \phi(x_i)\rangle \geq \rho_+ - \xi_i & \forall i \in [1,p] \\ \langle w_-, \phi(x_i)\rangle \geq \rho_- - \eta_i & \forall i \in [p+1,N] \\ y_i\langle w_+ - w_-, \phi(x_i)\rangle \geq \rho - \theta_i & \forall i \in [1,N] \\ y_i\langle w_+ - w_-, \phi(x_i)\rangle \geq -\rho - \varepsilon_i & \forall i \in [1,N] \\ \xi_i \geq 0,\ \eta_i \geq 0,\ \theta_i \geq 0,\ \varepsilon_i \geq 0 \end{cases} \quad (3)$$

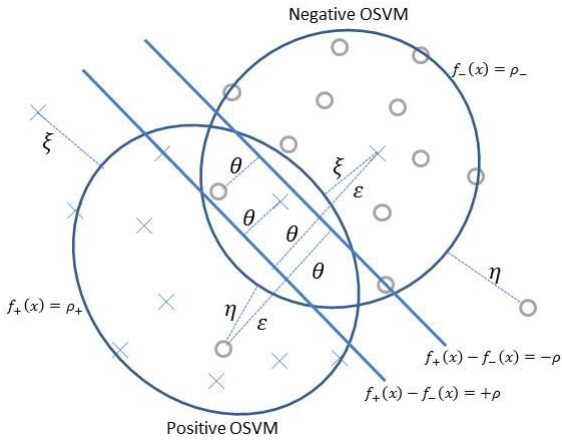

Fig. 1. The two one-class support vector machines classifier CONSUM

Terms w_+, and ρ_+ define the one-class SVM of the positive class, the ξ_i are the slack variables for the violation of the first constraint related to positive examples that are not in the positive one-class SVM. Terms w_-, and ρ_- define the one-class SVM of the negative class, the η_i are the slack variables for the violation of the second constraint related to negative examples that are not in the negative one-class SVM. The θ_i are the slack variables of the third constraint related to the examples in the ambiguity rejection region. The $\varepsilon_i$are the slack variables of the third constraint related to miss-classifications. The interaction between the two one-class SVM is gouverned by the third and fourth constraints, they define a region around the separator $\langle w_+ - w_-, x_i\rangle = 0$. It is similar to the margin region in a standard SVM, excepted that the size of margin (equal to $\frac{2\rho}{||w_+ - w_-||}$) is not maximized in our model. The margin size is independently optimized by the parameter ρ. This region is an approximation of the ambiguity rejection region, i.e. the intersection of the two one-class SVM.

The optimization problem is solved in dual form in introducing the Lagrangian multipliers α_i, γ_i, μ_i, α'_i, γ'_i, μ'_i for $i = 1..N$. The Lagrangian function can be write as:

$$
\begin{aligned}
L = & \frac{1}{2}||w_+||^2 - \rho_1 + \frac{1}{V_+ N_+}\sum_{i=1}^{p}\xi_i + \frac{1}{2}||w_-||^2 - \rho_0 + \frac{1}{V_- N_-}\sum_{i=p+1}^{N}\eta_i \\
& + \frac{C}{N}\left(C_r\sum_{i=1}^{N}\theta_i + C_e\sum_{i=1}^{N}\epsilon_i\right) \\
& - \sum_{i=1}^{p}\alpha_i(\langle w_+, \phi(x_i)\rangle - \rho_+ + \xi_i) - \sum_{i=p+1}^{N}\alpha_i(\langle w_-, \phi(x_i)\rangle - \rho_- + \eta_i) \\
& - \sum_{i=1}^{N}\gamma_i(\langle w_+ - w_-, \phi(x_i)\rangle y_i - \rho + \theta_i) - \sum_{i=1}^{N}\mu_i(\langle w_+ - w_-, \phi(x_i)\rangle y_i - \rho + \varepsilon_i) \\
& - \sum_{i=1}^{p}\alpha_i'\xi_i - \sum_{i=p+1}^{N}\alpha_i'\eta_i - \sum_{i=1}^{N}\gamma_i'\theta_i - \sum_{i=1}^{N}\mu_i'\varepsilon_i
\end{aligned}
$$

This function is maximized with respect to the Lagrange multiplier and minimized with the respect to primal variables: $max_{\alpha_i,\gamma_i,\mu_i} min_{w_+,w_-,\rho_+,\rho_-,\rho,\xi,\eta,\theta,\varepsilon} L$. Setting the derivatives of L with the respect to the primal variables equal to zero, one yields:

$$w_+ = \sum_{i=1}^{p}\alpha_i x_i + \sum_{i=1}^{N}\gamma_i x_i y_i + \sum_{i=1}^{N}\mu_i x_i y_i \tag{4}$$

$$w_- = \sum_{i=p+1}^{N}\alpha_i x_i - \sum_{i=1}^{N}\gamma_i x_i y_i - \sum_{i=1}^{N}\mu_i x_i y_i \tag{5}$$

$$\sum_{i=1}^{p}\alpha_i = 1; \qquad \sum_{i=p+1}^{N}\alpha_i = 1; \qquad \sum_{i=1}^{N}\gamma_i - \sum_{i=1}^{N}\mu_i = 0 \tag{6}$$

$$0 \le \alpha_i \le \frac{1}{V_+ N_+}\ \forall i \in [1,p]\ ; \qquad 0 \le \alpha_i \le \frac{1}{V_- N_-}\ \forall i \in [p+1, N] \tag{7}$$

$$0 \le \gamma_i \le \frac{CC_r}{N}\ ; \qquad 0 \le \mu_i \le \frac{CC_e}{N} \tag{8}$$

Substituting (4)-(8) into L and denoting $K'_{ij} = y_i y_j K_{ij}$ leads to the dual problem that is a quadratic programming problem:

$$
\begin{aligned}
min_{\alpha_i,\gamma_i,\mu_i} \quad & \frac{1}{2}\sum_{i,j=1}^{p}\alpha_i\alpha_j K'_{ij} + \frac{1}{2}\sum_{i,j=p+1}^{N}\alpha_i\alpha_j K'_{ij} + \sum_{i,j=0}^{N}\gamma_i\gamma_j K'_{ij} + \sum_{i,j=0}^{N}\mu_i\mu_j K'_{ij} \\
& + \sum_{i,j=0}^{N}\alpha_i\gamma_j K'_{ij} + \sum_{i,j=0}^{N}\alpha_i\mu_j K'_{ij} + 2\sum_{i,j=0}^{N}\gamma_i\mu_j K'_{ij} \qquad (9) \\
& \text{subject to the constraints (6) (7) and (8)}
\end{aligned}
$$

The classifier is defined by the following decision functions:

$$f_+(x_j) = \langle w_+, x_j \rangle - \rho_+ = \sum_{i=1}^{p} \alpha_i K_{ij} + \sum_{i=0}^{N} \gamma_i K_{ij} y_i + \sum_{i=0}^{N} \mu_i K_{ij} y_i - \rho_+ \quad (10)$$

$$f_-(x_j) = \langle w_-, x_j \rangle - \rho_- = \sum_{i=p+1}^{N} \alpha_i K_{ij} - \sum_{i=0}^{N} \gamma_i K_{ij} y_i - \sum_{i=0}^{N} \mu_i K_{ij} y_i - \rho_- \quad (11)$$

$f_+(x_j)$ returns a positive value if the example x_j is contained in the positive one-class SVM and negative value otherwise. If $f_+(x_j) = 0$, then x_j is on the one-class SVM boundary and we have $0 < \alpha_j < \frac{1}{V_+ N_+}$. This kind of example is exploited in order to recover ρ_+, since x_j satisfies $\rho_+ = \langle w_+, x_j \rangle$. ρ_- is computed in the same way. The final decision of CONSUM is based on the two decision functions.

$$\Psi(x) = \begin{cases} +1 & \text{if} \quad f_+(x) > 0 \wedge f_-(x) \leq 0 \\ -1 & \text{if} \quad f_+(x) \leq 0 \wedge f_-(x) > 0 \\ R_a & \text{if} \quad f_+(x) > 0 \wedge f_-(x) > 0 \\ R_d & \text{if} \quad f_+(x) \leq 0 \wedge f_-(x) \leq 0 \end{cases} \quad (12)$$

Our model critically depends on the choice of the hyper-parameters V_+, V_-, C, C_r and C_e. V_+ and V_- control the rate of outliers in the training set. By default we consider that the training set contains no outliers. We want these parameters low, we set them to $V_+ = V_- = 0.05$. V_+ and V_- are not set to zero because it would overconstraints the general formulation (2-3). C_r and C_e control the trade-off between the ambiguity rejection and the error rate. The optimal trade-off is given by λ_E and λ_R. In our model the loss of an ambiguity rejected example is $CC_r\theta_i$ and the loss of an error is $C(C_r\theta_i + C_e\varepsilon_i)$. We want that C_e, C_r respect the ratio between the error and rejection costs $\frac{C_r + C_e}{C_r} = \frac{\lambda_E}{\lambda_R}$ and be normalized such that $C_r + C_e = 1$. That leads to $C_r = \frac{\lambda_R}{\lambda_E}$ and $C_e = \frac{\lambda_E - \lambda_R}{\lambda_E}$. The parameters C controls the importance of the error and ambiguity rejection loss, it plays the same role as the C in the usual SVM model. This parameter will be optimized during the model fitting in an inner cross-validation procedure.

3.2 Optimization Algorithm

Our model is formulated as a quadratic programming problem with linear constraints (9). Several approaches are available to solve this type of problems. We propose a modified version of the sequential minimal optimization (SMO) algorithm that was originally developed for SVM training [14], a version for one-class SVM has also been proposed [16]. It has the advantage to have a lower complexity than the other methods and does not need to keep the whole Gram matrix in memory. It is therefore adapted to large datasets. The principle of SMO is to divide the original optimization problem into several optimization tasks of the smallest size. In our case the smallest size is two, we have to optimize over pairs of multipliers. According to the contraints (6), (7) and (8), there are three different

types of multiplier pairs $\{(\alpha_u, \alpha_v)/u, v \in [1, p]])\}$, $\{(\alpha_u, \alpha_v)/u, v \in [p+1, N]\}$ and all pairs form from the γ and μ. Note that this last types of pairs can divided into three subtypes $\{(\gamma_u, \gamma_v)\}$, $\{(\mu_u, \mu_v)\}$ and $\{(\gamma_u, \mu_v)\}$. The optimization algorithm consists to select a pair of multipliers and optimize only these two multipliers; this process is iterated until a stopping criterion.

Initialization. The multipliers can be initialized with any values, the only conditions is that they respect the constraints (7-8). At the beginning of our algorithm, the α_i's for $i \in [1, p]$ are initialized to $\frac{1}{N_+}$, the α_i's for $i \in [p+1, N]$ are initialized to $\frac{1}{N_-}$ and the γ_i's and μ_i's are initialized to $\frac{1}{N}$.

Stopping Criterion. The optimization algorithm is stopped when the gain of loss is null. However, in our algorithm we have to use a nonzero accuracy tolerance such that two values are considered equal if they differ by less than e. In practice the procedure is stopped at the iteration t when $L^t - L^{t-1} < e$. In our experiments $e = 0.0001$

Multipliers Pair Optimization Step. Our algorithm is a succession of multiplier pair optimization tasks. Suppose that at a given iteration, the pair of multipliers γ_u and γ_v is selected to be optimized. All others multipliers are considered as constants during this task. From the constraint (6) we have $\gamma_u + \gamma_v = \sum_{i \neq u,v}^{N} \gamma_i - \sum_{i=0}^{N} \mu_i = D$. The quadratic problem (9) is written in function on γ_u and γ_v :

$$L = \gamma_i^2 K'_{uu} + \gamma_j^2 K'_{vv} + 2\gamma_i\gamma_j K'_{uv} + 2\gamma_u G_u + 2\gamma_v G_v + G + \gamma_u A_u^+ + \gamma_v A_v^+ + A^+ + \gamma_u A_u^- + \gamma_v A_v^- + A^- + 2\gamma_u M_u + 2\gamma_v M_v + 2M - \gamma_u - \gamma_v + X$$

$$A_x^+ = \sum_{i=1}^{p} \alpha_i K'_{ix} \quad A^+ = \sum_{\substack{i=1 \\ j \neq u,v}}^{p} \alpha_i\gamma_j K'_{ij} \quad A_x^- = \sum_{i=p+1}^{N} \alpha_i K'_{ix} \quad A^+ = \sum_{\substack{i=p+1 \\ j \neq u,v}}^{N} \alpha_i\gamma_j K'_{ij}$$

$$G_x = \sum_{i \neq x} \gamma_i K'_{ix} \quad G = \sum_{i,j \neq u,v} \gamma_u\gamma_v K'_{i,j} \quad M_x = \sum_{i \neq x} \mu_i K'_{ix} \quad M = \sum_{i,j \neq u,v} \mu_u\mu_v K'_{i,j}$$

$$X = \frac{1}{2}\sum_{i,j=1}^{p} \alpha_i\alpha_j K'_{ij} + \frac{1}{2}\sum_{i,j=p+1}^{N} \alpha_i\alpha_j K'_{ij} + \sum_{i,j=1}^{N} \alpha_i\mu_j K'_{ij} + \sum_{i,j=1}^{N} \mu_i\mu_j K'_{ij} + \sum_{i=1}^{N} \mu_i$$

Note that A_x^+, A^+, A_x^-, A_-, G_x, G, M_x, M, X are constants in this step. In using $\gamma_u = D - \gamma_v$, L in expressed only in function on γ_v and compute its derivative:

$$\frac{\partial L}{\partial \gamma_v} = 2(\gamma_v - D)K'_{uu} + 2(D - 2\gamma_v)K'_{uv} + 2\gamma_v K'_{vv} + A_v^+ y_v - A_u^+ y_u + A_u^- y_u - A_v^- y_v + 2G_v y_v - 2G_u y_u + 2M_v y_v - 2M_u y_u$$

Setting the derivative to zero leads the optimal value of γ_v:

$$\gamma_v^{new} = \frac{2D(K_{uu} - K_{uv}) + A_u^+ - A_v^+ + A_u^- - A_v^- + 2G_u - 2G_v + 2M_u - 2M_v}{2(K_{uu} + K_{vv} - 2K_{uv}Y_{uv})}$$

It is simpler to rewrite this equation in introducing the decision functions f_+ and f_- :

$$\gamma_u^{new} = \gamma_u - \Delta\ , \qquad \gamma_v^{new} = \gamma_v + \Delta$$
$$\Delta = \frac{y_v(f_-(x_v) - f_+(x_v)) + y_u(f_+(x_u) - f_+(x_u))}{2(K_{uu} + K_{vv} - 2K_{uv}Y_{uv})}$$

Let recall that the multipliers γ_i are still subject to the constraint (8). If γ_u^{new} or γ_v^{new} is outside from the interval $[0, \frac{CC_r}{N}]$, the constraint optimum is found by projecting it into the bound of the allowed interval, then Δ, γ_u^{new} and γ_v^{new} are recomputed. One the multiplier pair has been optimized, the decision functions should be recomputed. However it is not necessary to use the formulas (10-11), for saving computing resources we can updated them by:

$$f_+^{new}(x_i) = f_+(x_i) + \delta_+ \text{ with } \delta_+ = \Delta(K_{ui}y_u - K_{vi}y_v)$$
$$f_-^{new}(x_i) = f_+(x_i) + \delta_- \text{ with } \delta_- = \Delta(K_{vi}y_v - K_{ui}y_u)$$

This procedure is repeated at each iteration, however the formulas differs slightly in function on the type of multiplier pairs. The table 2 gives the formulas of Δ,δ_+ and δ_- for each type of pair.

Table 2. Δ, δ_+ and δ_- used in the different cases of pair optimization tasks

pairs	Δ	δ_+	δ_-
α_u,α_v u,v∈[1,p]	$\frac{f_+(x_u)-f_+(x_v)}{K_{uu}+K_{vv}-2K_{uv}Y_{uv}}$	$\Delta(K_{vi} - K_{ui})$	0
α_u,α_v u,v∈[p+1,N]	$\frac{f_-(x_u)-f_-(x_v)}{K_{uu}+K_{vv}-2K_{uv}Y_{uv}}$	0	$\Delta(K_{vi} - K_{ui})$
γ_u,γ_v	$\frac{y_v(f_-(x_v)-f_+(x_v))+y_u(f_+(x_u)-f_-(x_u))}{2(K_{uu}+K_{vv}-2K_{uv}Y_{uv})}$	$\Delta(K_{vi}y_v - K_{ui}y_u)$	$\Delta(K_{ui}y_u - K_{vi}y_v)$
μ_u,μ_v	$\frac{y_v(f_-(x_v)-f_+(x_v))+y_u(f_+(x_u)-f_-(x_u))}{2(K_{uu}+K_{vv}-2K_{uv}Y_{uv})}$	$\Delta(K_{vi}y_v - K_{ui}y_u)$	$\Delta(K_{ui}y_u - K_{vi}y_v)$
γ_u,μ_v	$\frac{y_u(f_-(x_u)-f_+(x_u))+y_v(f_-(x_v)-f_+(x_v))}{2(K_{uu}+K_{vv}+2K_{uv}Y_{uv})}$	$\Delta(K_{ui}y_u + K_{vi}y_v)$	-$\Delta(K_{ui}y_u + K_{vi}y_v)$

Selection of the Multiplier Pairs. At each iteration a multiplier pair is selected to be optimized. The pair selection could be random but an intelligent selection procedure allows to speed up the optimization process. There are three different types of pair, each of these pair types will be handled successively. Let's focus on the optimization of the first type of pair, i.e. $\{(\alpha_u, \alpha_v)/u, v \in [1, p]\}$. We randomly select a positive example x_u that does not respect the following condition :

$$\begin{cases} \text{if } \alpha_u = 0 \text{ then } f_+(x_u) > \rho_+ \\ \text{if } 0 < \alpha_u < \frac{1}{V_+N_+} \text{ then } f_+(x_u) = \rho_+ \\ \text{if } \alpha_u = \frac{1}{V_+N_+} \text{ then } f_+(x_u) < \rho_+ \end{cases}$$

If all positive examples respect this condition, x_u is randomly selected among the positive examples. This condition is checked in using the nonzero accuracy tolerance introduced in the section 4.2. The second multipliers α_v is the one maximizing the numerator of Δ i.e. $v = argmax_{i \in \oplus} |f_+(x_u) - f_+(x_i)|$. Then the values of α_u, α_v, ρ_+ and $f_+(x_i)$ are updated. The idea is to select a pair that will produce a large change (large Δ) in the values of α_u and α_v. Only the numerator of Δ is computed for the pair selection because it is very fast since we already have the value $f_+(x_u)$ and $f_+(x_v)$. If we use the real value of Δ, the denominator have to be computed which would be much more slower. Other pairs of the same type $\{(\alpha_u, \alpha_v)/u, v \in \oplus)\}$ are selected and optimized until the gain of loss becomes small i.e. $L^t - L^{t-1} < 10e$. Then the same optimization procedure is run on the two other types of pair. The whole procedure is iterated until the stopping criterion is reached.

4 Experimental Validation

4.1 Experiment Settings

The goal of the experiments is to investigate the sensitivity of CONSUM with respect to the hyper-pameters and to compare its performance to the state of the art. There exist very few methods that construct classifiers with both ambiguity and outlier rejection. Since our method is based on the combination of one-class SVM, we tested the association of two one-class SVM (2OSVM), this approach has been proposed in [17]. A one-class SVM is independently constructed for each class and the decision rule is the same than in Eq.(12). The second method tested is the support vector machine with the ambiguity rejection computed by the Chow rule. The construction of the outlier rejection is more problematic since it has been very few studied in the literature. The best way to have a fair comparison with CONSUM is to base the outlier rejection on a one-class SVM. A one-class SVM is computed on all training examples (both positive and negative class), each example outside from the one-class SVM will considered as an outlier[1]. In our experiments the different methods use a Gaussian kernel whose variance σ is determined in an inner cross-validation loop. The costs of miss-classification and rejection are set to $\lambda_E = 4$ and $\lambda_R = 1$

We did experiments on artificial datasets in order to show the behavior of the different methods and analyze the impact of parameters of our approach. The artificial data are generated from Gaussian distributions with independent

[1] Note that since the Chow rule computes two thresholds on the classifier output to define the ambiguity rejection (Eq.(1)), some researchers have proposed to add two other thresholds to define the outlier rejection. These thresholds T_N, T_P are at the extremes of the classifier output, all examples that are not in the interval $[T_N, T_P]$ are considered as outliers. We think that this approach is not good and specially with Gaussian kernel. Indeed, the higher values will be obtained by examples that are in the center of the distribution of positive class and have the highest probabilities to belong to the positive class. It is therefore not correct to consider these examples as outlier rejections.

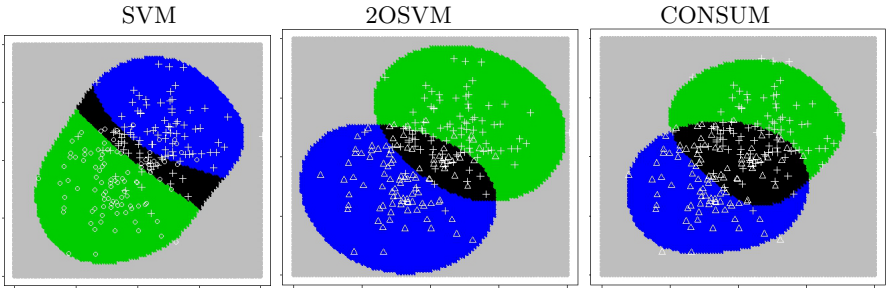

Fig. 2. Classifiers constructed on the artificial dataset with the different methods. The green, blue, black and gray region represents respectively the positive, negative, ambiguity rejection and outlier rejection region. The triangles and crosses represent respectively the positive and negatives examples of the training set.

variables. To simulate the presence of outliers, we add to the test set examples of a third class whose the distribution is uniform on the input space. Some experiments are based on 2-dimension data in order to support visualization of the classifiers.

4.2 Sensitivity Analysis

The figure 2 shows the classifiers obtained with the different methods on the artificial problem. The green, blue, black and gray region represents respectively the positive, negative, ambiguity and outlier rejection region. We see that CONSUM fits the ambiguity and outlier rejection regions better than the other methods. The difference between the methods can be mainly seen in the shape of the ambiguity rejection. The ambiguity rejection region of SVM is spread around the boundary of the non-reject SVM. The ambiguity region of 2OSVM and CONSUM contains only region where positive and negative examples are mixed. The difference between 2OSVM and CONSUM is the size of the ambiguity rejection. The ambiguity region of CONSUM is larger and less regular. In 2OSVM, the two one-class SVM are independent, there is no way to control the trade-off between error and ambiguity rejection. In CONSUM, this trade-off is controlled by the parameters C_E, C_R and the constraints (3), it has more degree of fredoom than 2OSVM. The fact that the size of ambiguity region is large in CONSUM comes from the missclassification cost that is much higher than the rejection cost in our simulation.

One of the crucial points of classification with reject option is the control of the trade-off between the error rate and the rejection rate. In our model, this is done by ρ_+ and ρ_-. We investigate the behavior of the error rate, ambiguity rejection rate and outlier rejection rate in function on these thresholds. A CONSUM classifier has been constructed on artificial data, we vary the value of ρ_+ and ρ_- from 0 to $max\{f_+(x_i), f_-(x_i)|i = 1..N\}$ and observe the impact of the error rate, ambiguity and outlier rejection in the figure 3. The ambiguity rejection is

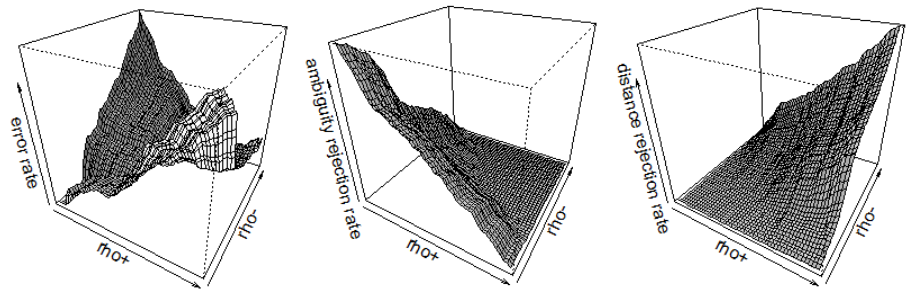

Fig. 3. Evolution of the rate of error, ambiguity rejection and outlier rejection in function of the thresholds ρ_+ and ρ_- of the CONSUM classifier

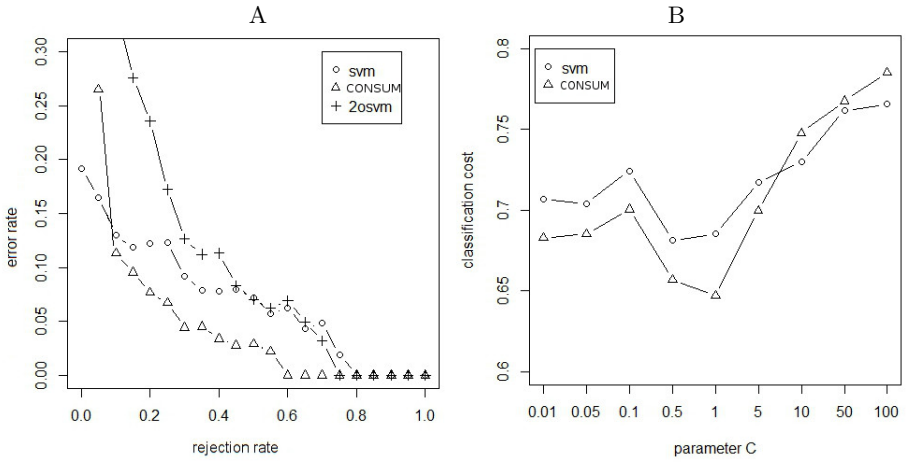

Fig. 4. A: Trade-off between error and rejection rate of the different methods. B: Classification cost in function on the parameter C of SVM and CONSUM (There is no parameter C in 2OSVM).

decreasing with ρ_+ and ρ_-, on the opposite the outlier rejection is increasing with ρ_+ and ρ_-. The error rate is null when $\rho_+ = \rho_- = 0$, this corresponds to the trivial case where all points are outlier rejected, there is therefore no missclassification. On the opposite when both ρ_+ and ρ_- reach their maximum, the two classes greatly overlap, there are few missclassifications. When ρ_+ is maximum and ρ_- is null, the positive class dominates the negative class. All non-rejected negative examples are false positive, the error rate is therefore very high. We have the same thing when ρ_- is maximum and ρ_+ is null. This simulation illustrates the relation between the error rate and the different types of rejection of our model.

When the cost matrix of the classification problem is not known, the most convenient method to compare several classifiers with reject option is to plot their error-rejection curve (ERC). The ERC gives the error of a classifier in

function the rejection rate (both ambiguity and outlier). When we consider both ambiguity and outlier rejection, several different classifiers and rejection regions may give the same rejection rate. For a given rejection rate, there are generally several error rates. If we test all values of ρ_+ and ρ_-, the performances of the classifier are illustrated by a scatter plot in the error-rejection space. If we keep only the lowest error for each rejection rate value, the performance of the classifier is represented by a curve. It is the Pareto front of the scatter plot. These curves(ERC) can be used to compare the performance of different classifiers. The figure 4A gives the ERC of the SVM, 2OSVM and CONSUM classifiers on artificial data. For reject.rate$\geq$0.1 CONSUM is the best classifier and reaches error.rate=0 for reject.rate=0.6. 2OSVM is never competitive, it reaches the performance for SVM for reject.rate$\geq$0.45 and the performance of CONSUM for reject.rate$\geq$0.75. Notes that CONSUM has no points for reject.rate=0 because the model does not return empty rejection regions whatever the values of ρ_+ and ρ_-. In theory it is possible to reach reject.rate=0 if both all test points are in one of the one-class SVM and there is no overlap between the two one-class SVM. In practice this case is very rare.

In the next experiments we compare the performances of the classifiers in computing their classification cost on a test set of size N_{ts}. Let's $I(x) = 1$ if x is true, 0 otherwise; the classifier cost is defined by:

$$cost(\Psi) = \frac{1}{N_{ts}} \sum_{i=1}^{N_{ts}} \lambda_E I\left(\Psi(x_i) \neq R_a \vee R_d\right) I\left(\Psi(x_i) \neq y_i\right) + \lambda_R I\left(\Psi(x_i) = R_a \vee R_d\right)$$

The figure 4B gives the classification cost of SVM and CONSUM in function on their hyper-parameter C. The role of C is similar in the SVM and CONSUM classifier. In SVM this parameter controls the trade-off between the maximization of the margin and the good classification of the examples. In CONSUM, it controls the trade-off between the optimization of the two one-class SVM and the constraints on the miss-classifications and rejections of the training examples. We see that the curves of SVM and CONSUM has a similar shape, they reach their minimum around $0.5 < C < 1$. At their optimal parameter, CONSUM gives better results than SVM. In our next experiments, the parameters C is determined by an inner cross-validation procedure.

4.3 Comparative Study

We made some experiments on real data with three different rejection scenarios. We used six datasets from the UCI repository. The table 3 shows the classification cost of the different methods on the six datasets. The first scenario is the classification with no rejection. We use the usual SVM, for 2OSVM and CONSUM a class is assigned to an example according to $\langle w_+ - w_-; x_i \rangle$. Results are in the first three columns of the table 3. The best results are obtains mainly by the SVM. However CONSUM obtains good results, its performances are close to the SVM. In the second scenario, only ambiguity rejection is considered. The Chow rule is added to the SVM, for 2OSVM and CONSUM the outlier rejection

Table 3. Classification cost on artificial and real data experiments. The best results are in bold.

		no reject			ambiguity reject			amb.& outlier reject		
datasets	dimension	SVM	2OSVM	CONSUM	SVM	2OSVM	CONSUM	SVM	2OSVM	CONSUM
Artif.	2000×50	**1.209**	1.454	1.411	**0.997**	1.169	1.093	0.939	0.933	**0.884**
Artif.	2000×100	0.930	1.092	**0.919**	**0.744**	0.826	0.756	0.778	0.843	**0.750**
wdbc	569×30	0.325	0.410	**0.312**	0.307	0.279	**0.194**	0.325	0.508	**0.194**
spam	4601×57	**0.262**	0.268	0.309	0.115	0.212	**0.061**	0.248	0.294	**0.152**
madelon	2000×500	**1.302**	0.1430	1.332	1.100	1.283	**1.017**	0.967	0.953	**0.919**
pop	540×18	**0.261**	0.284	0.272	0.250	0.250	**0.238**	0.355	0.403	**0.351**
transfusion	748×4	0.946	1.002	**0.926**	**0.801**	0.964	0.902	**0.839**	0.959	0.851
bank	1374×4	**0.066**	0.080	0.092	**0.051**	0.074	0.061	0.208	**0.179**	0.182

region is not used. Results are given in the three middle columns. CONSUM has the best performance for four datasets and SVM for two datasets. In the last scenario, we consider both ambiguity and outlier rejection, outliers are added to the test set as in the artificial dataset. A one-class SVM is added to the SVM, 2OSVM and CONSUM are used normally. Results are in the three last columns. These last results are the most important since the scenario represents the practical cases. We see that CONSUM obtains the best performances. The results show that our method gives both a reliable representation of the two classes by one-class SVM and a good trade-off between the rejection and the error rate. It is interesting to note that when there is no rejection, SVM is much better than the other classifiers. When the rejection rate is significant CONSUM becomes better than SVM. These results confirms the conclusion of [3] the best classifier with rejection option is not the best classifier without rejection on which a rejection rule is added. Note also when we compare the "no reject" scenario to the two others, we conclude that the use of a rejection option improve greatly the performance of the classifier whatever the method used.

5 Conclusion

We have introduced a new approach for classification with rejection option CONSUM that constructs simultaneously both the outlier and ambiguity rejection regions. The outlier rejection is generally ignored in the state of the art, but it is essential when the test set contains outliers, that is common in real applications. We showed that CONSUM can be viewed as a quadratic programming problem and we proposed an optimization algorithm adapted for large datasets. The results showed that our method improves the performance of classification.

In this paper we assumed that the cost of false positive and false negative are equal ($\lambda_{FN} = \lambda FP$) and the cost of positive rejection and negative rejection are equal. If the classification problem is cost sensitive or unbalanced, we may want to assigned different costs to the two classes. In this case we introduce different costs for the ambiguity rejection ($\lambda_{RNa} \neq \lambda RPa$) and for the outlier rejection ($\lambda_{RNd} \neq \lambda RPd$). The rejection constraints of the minimization problem (3) are split into constraints for positive examples and constraints for negative examples. Different penalties are assigned to the loss of each classes. The minimization problem can still be solved by the optimization algorithm presented in section

4. The only difference is that the multipliers γ and μ of positive and negative examples have to be optimized separately as with the α.

Our future works should focus on the multi-class problem. We can easily define a one-class SVM for each class but the constraints for the miss-classifications and ambiguity rejection are defined only for binary problems. The number of hyper-parameters is the main challenges since it increase quadratically with the number of classes. Several methods has been proposed to solve the problem of multi-class SVM [21]. They could be a source of inspiration in order to propose a multi-class version of CONSUM.

References

1. Chow, C.: On optimum recognition error and reject tradeoff. IEEE Transactions on Information Theory 16(1), 41–46 (1970)
2. Dubuisson, B., Masson, M.: A statistical decision rule with incomplete knowledge about classes. Pattern Recognition 26(1), 155–165 (1993)
3. El-yaniv, R., Wiener, Y.: Agnostic selective classification. In: Neural Information Processing Systems NIPS (2011)
4. El-Yaniv, R., Wiener, Y.: On the foundations of noise-free selective classification. Journal of Machine Learning Research 99, 1605–1641 (2010)
5. Fumera, G., Roli, F., Giacinto, G.: Multiple reject thresholds for improving classification reliability. In: Amin, A., Pudil, P., Ferri, F., Iñesta, J.M. (eds.) SPR/SSPR 2000. LNCS, vol. 1876, p. 863. Springer, Heidelberg (2000)
6. Grandvalet, Y., Rakotomamonjy, A., Keshet, J., Canu, S.: Support vector machines with a reject option. In: NIPS pp. 537–544 (2008)
7. Hanczar, B., Dougherty, E.: Classification with reject option in gene expression data. Bioinformatics 24(17), 1889–1895 (2008)
8. Hansen, L.K., Liisberg, C., Salamon, P.: The error-reject tradeoff. Open Systems & Information Dynamics 4, 159–184 (1997)
9. Kelley, L., Scott, M.: The evolution of biology. A shift towards the engineering of prediction-generating tools and away from traditional research practice. EMBO Reports 9(12), 1163–1167 (2008)
10. Landgrebe, T., Tax, D., D.M.J., P., R.P.W., D.: The interaction between classification and reject performance for distance-based reject-option classifiers. Pattern Recognition Letters 27(8), 908–917 (2006)
11. Nadeem, M., Zucker, J., Hanczar, B.: Accuracy-rejection curves (arcs) for comparing classification methods with a reject option. Journal of Machine Learning Research - Proceedings Track 8, 65–81 (2010)
12. Pietraszek, T.: Optimizing abstaining classifiers using roc analysis. In: Proceedings of the 22nd International Conference on Machine Learning, ICML 2005, pp. 665–672 (2005)
13. Pietraszek, T.: On the use of roc analysis for the optimization of abstaining classifiers. Machine Learning 68(2), 137–169 (2007)
14. Platt, J.C.: Sequential minimal optimization: A fast algorithm for training support vector machines. Tech. rep., Advances in Kernel Methods - Support Vector Learning (1998)
15. Santos-Pereira, C.M., Pires, A.M.: On optimal reject rules and roc curves. Pattern Recognition Letters 26(7), 943–952 (2005)