2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Robust Support Vector Machine

Trung Le, Dat Tran, Wanli Ma, Thien Pham, Phuong Duong, and Minh Nguyen

***Abstract*—Support Vector Machine (SVM) is a well-known kernel-based method for binary classification problem. SVM aims at constructing the optimal middle hyperplane which induces the largest margin. It is proven that in a linearly separable case, this middle hyperplane offers the high accuracy on universal datasets. However, real world datasets often contain overlapping regions and therefore, the decision hyperplane should be adjusted according to the profiles of the datasets. In this paper, we propose Robust Support Vector Machine (RSVM), where the hyperplanes can be properly adjusted to accommodate the real world datasets. By setting the value of the adjustment factor properly, RSVM can handle well the datasets with any possible profiles. Our experiments on the benchmark datasets demonstrate the superiority of the RSVM for both binary and one-class classification problems.**

***Index Terms*—Kernel-based method, Support Vector Machine, One-class Support Vector Machine.**

I. Introduction

Support Vector Machine (SVM) [1, 2] is a well-known kernel-based method for classification problem. SVM aims at constructing the optimal hyperplane such that the margin, i.e., the distance from the closest data sample of the training set to the hyperplane, is maximized. It is proven that in a linearly separable case, the optimal (middle) hyperplane offers the high accuracy not only on the currently collected training set but also on the universal dataset [3, 4]. However, for any real dataset with the overlap of positive and negative regions, the decision hyperplane should be adjusted according to the profile of the dataset. This is the same idea in the generative linear models of Bayes inference, e.g. naive Bayes, etc., in which the decision hyperplane can be adjusted according the prior and posterior probabilities of the classes [5].

It is known that SVM offers good performance for balanced datasets, but may perform poorly on imbalanced datasets. To solve the problem, One-class Support Vector Machine (OCSVM) [6] and Support Vector Data Description (SVDD) [7, 8] were proposed. OCSVM tries constructing the optimal hyperplane such that the margin, i.e., the distance from the origin to the hyperplane, is maximized. The obvious drawback of OCSVM is that it regards all negative data samples the same as a common symbol (the origin) and certainly cannot efficiently utilize the information carried by them. In contrast, SVDD is able to utilize the negative data samples, but the derivation of SVDD in its original paper [7] is not theoretically sound, because the constrains may not be convex, and then Karush-Kuhn-Tucker (KKT) theorem cannot be applied for the derivation.

In this paper, we propose Robust Support Vector Machine (RSVM) where the optimal hyperplanes can be adjusted to fit the profiles of the datasets. In RSVM, two margins are adjustable and proportional with a certain ratio μ, called *the adjustment factor*. By setting the appropriate values for μ, RSVM can adjust well to both the balanced and imbalanced datasets. We also suggest, in this paper, the procedure to calculate the optimal value of μ.

Trung Le, Thien Pham, Phuong Duong, and Minh Nguyen are with the Faculty of Information Technology, the HCMc University of Pedagogy, Hochiminh city, Vietnam (email: {trunglm, thienph, phuongdth, and minhnn}@hcmup.edu.vn).

Dat Tran and Wanli Ma are with the Faculty of Education, Science, Technology and Mathematics, the University of Canberra, Australia (email:{dat.tran, wanli.ma}@canberra.edu.au).

II. Support Vector Machine

SVM [1, 2] aims at constructing an optimal hyperplane, which can separate the positive and negative classes such that the margin, i.e., the distance from the closest sample of the training set to the hyperplane, is maximized. The optimization problem of SVM is as follows:

$$\begin{array}{ll} \underset{w,\rho}{min} & \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i\right) \\ s.t.: & \forall_{i=1}^{N} : y_i\left(w^T x_i - \rho\right) \geq 1 - \xi_i \\ & \forall_{i=1}^{N} : \xi_i \geq 0 \end{array}$$

where $\xi = [\xi_i]_{i=1}^{N}$ is the vector of slack variables, C is the trade-off parameter.

The decision function is of the following form:

$$f(x) = sign(w^T x - \rho)$$

III. One-class Support Vector Machine

OCSVM [6] builds an optimal hyperplane that can separate the origin and the positive class such that the margin, i.e., the distance from the origin to the hyperplane, is maximized. The optimization problem of OCSVM is given by:

$$\begin{array}{ll} \underset{w,\rho}{min} & \left(\frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu N}\sum_{i=1}^{N}\xi_i\right) \\ s.t.: & \forall_{i=1}^{N} : w^T x_i \geq \rho - \xi_i \\ & \forall_{i=1}^{N} : \xi_i \geq 0 \end{array}$$

where the training set $X = \{x_1, x_2, \ldots, x_N\}$ contains only positive data, ν is a constant.

The decision function is of the following form:

$$f(x) = sign(w^T x - \rho)$$

IV. Robust Support Vector Machine

A. The Basic Idea

To propose the model of RSVM, we first examine the model of SVM from a different viewpoint. We start with a linearly separable case with the following lemma.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

Lemma 1. *Given the linearly separable training set* $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ *. Let us denote* m^+, m^- *by the distances from the closest samples of positive and negative classes, respectively, to the optimal hyperplane. The following holds:*

$$m^+ = m^- = m = \max_{1 \le i \le N} \left(\frac{y_i \left(w^T x_i - \rho \right)}{\|w\|} \right)$$

Proof: We prove by contradiction. Suppose that $m^+ > m^-$, by slightly moving the optimal hyperplane toward the positive class, we gain a new hyperplane, which is parallel to the optimal hyperplane, but has a larger margin. Therefore, $m^+ = m^-$. Furthermore, we also have $m = min\{m^+, m^-\} = m^+ = m^-$. ■

According to *Lemma 1*, we can reformulate SVM in separable case as follows:

$$\begin{array}{rc} \max\limits_{w,\rho} & (m^+ + m^-) \\ s.t.: & m^+ = m^- \\ & m^+ = \min\limits_{y_i=1} \left(\frac{y_i \left(w^T x_i - \rho \right)}{\|w\|} \right) \\ & m^- = \min\limits_{y_i=-1} \left(\frac{y_i \left(w^T x_i - \rho \right)}{\|w\|} \right) \\ & \forall_{i=1}^N : y_i \left(w^T x_i - \rho \right) \ge 0 \end{array} \quad (1)$$

To extend SVM, we allow the two margins adjustable and proportional by *the adjustment factor*, i.e., $m^+ = \mu m^-$. The optimization problem in Eq. (1) becomes:

$$\begin{array}{rc} \max\limits_{w,\rho} & (m^+ + m^-) \\ s.t.: & m^+ = \mu m^- \\ & m^+ = \min\limits_{y_i=1} \left(\frac{y_i \left(w^T x_i - \rho \right)}{\|w\|} \right) \\ & m^- = \min\limits_{y_i=-1} \left(\frac{y_i \left(w^T x_i - \rho \right)}{\|w\|} \right) \\ & \forall_{i=1}^N : y_i \left(w^T x_i - \rho \right) \ge 0 \end{array} \quad (2)$$

B. The Derivation of RSVM

To derive the optimization problem in Eq. (2), because the positive and negative margins m^+ and m^- are invariant if we scale (w, ρ) by a factor k, without loss of generality, we can assume that: $\min\limits_{y_i=1} y_i \left(w^T x_i - \rho \right) + \min\limits_{y_i=-1} y_i \left(w^T x_i - \rho \right) = 2$. The optimization problem in Eq. (2) thus becomes:

$$\begin{array}{c} \min\limits_{w,\rho} \left(\frac{1}{2} \|w\|^2 \right) \\ s.t.: \forall_{i=1}^N y_i = 1 : y_i \left(w^T x_i - \rho \right) \ge \frac{2\mu}{\mu+1} \\ \forall_{i=1}^N y_i = -1 : y_i \left(w^T x_i - \rho \right) \ge \frac{2}{\mu+1} \end{array} \quad (3)$$

We extend the optimization problem in Eq. (3) by using the slack variables to accommodate the soft model as follows:

$$\begin{array}{c} \min\limits_{w,\rho,\xi} \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \right) \\ s.t.: \forall_{i=1}^N y_i = 1 : y_i \left(w^T x_i - \rho \right) \ge \frac{2\mu}{\mu+1} - \xi_i \\ \forall_{i=1}^N y_i = -1 : y_i \left(w^T x_i - \rho \right) \ge \frac{2}{\mu+1} - \xi_i \\ \forall_{i=1}^N : \xi_i \ge 0 \end{array} \quad (4)$$

C. The Solution

We apply Karush-Kuhn-Tucker (KKT) theorem to solve the optimization problem in Eq. (4). The Lagrange function is of the following form:

$$\begin{aligned} L(w, \rho, \xi, \alpha, \beta) = & \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\ & - \sum_{i=1}^N \alpha_i \left(y_i \left(w^T x_i - \rho \right) - \theta_i + \xi_i \right) - \sum_{i=1}^N \beta_i \xi_i \end{aligned}$$

where $\theta_i = \begin{cases} \frac{2\mu}{\mu+1} & if\ y_i = 1 \\ \frac{2}{\mu+1} & if\ y_i = -1 \end{cases}$

By setting the derivatives to zero, we achieve:

$$\frac{\delta L}{\delta w} = 0 \rightarrow w = \sum_{i=1}^N y_i \alpha_i x_i \quad (5)$$

$$\frac{\delta L}{\delta \rho} = 0 \rightarrow \sum_{i=1}^N y_i \alpha_i = 0 \quad (6)$$

$$\forall_{i=1}^N : \frac{\delta L}{\delta \xi_i} = 0 \rightarrow \alpha_i + \beta_i = C \quad (7)$$

$$\begin{aligned} \forall_{i=1}^N : \alpha_i \ge 0,\ & y_i \left(w^T x_i - \rho \right) - \theta_i + \xi_i \ge 0 \\ & , \alpha_i \left(y_i \left(w^T x_i - \rho \right) - \theta_i + \xi_i \right) = 0 \end{aligned} \quad (8)$$

$$\forall_{i=1}^N : \beta_i \ge 0,\ \xi_i \ge 0,\ \beta_i \xi_i = 0 \quad (9)$$

By substituting Eqs. (5 - 7) to the Lagrange function, we obtain the following optimization problem:

$$\begin{array}{c} \min\limits_{\alpha} \left(\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j x_i^T x_j \alpha_i \alpha_j - \sum_{i=1}^N \theta_i \alpha_i \right) \\ s.t.: \sum_{i=1}^N y_i \alpha_i = 0 \\ \forall_{i=1}^N : 0 \le \alpha_i \le C \end{array} \quad (10)$$

The Kernel trick can be employed to transform the optimization problem of Eq. (10) in the input space to that in the feature space as follows:

$$\begin{array}{c} \min\limits_{\alpha} \left(\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j K(x_i, x_j) \alpha_i \alpha_j - \sum_{i=1}^N \theta_i \alpha_i \right) \\ s.t.: \sum_{i=1}^N y_i \alpha_i = 0 \\ \forall_{i=1}^N : 0 \le \alpha_i \le C \end{array} \quad (11)$$

To compute ρ, let us denote $I = \{i : 1 \le i \le N \wedge 0 < \alpha_i < C\}$. For all $i \in I$, according to KKT conditions in Eqs. (8,9), we have:

$$y_i\left(w^T x_i-\rho\right)=\theta_i \rightarrow \rho=w^T x_i-y_i\theta_i=\sum_{j=1}^{N} y_j\alpha_j x_i^T x_j-y_i\theta_i \tag{12}$$

If a general kernel function is in use, by referring the Kernel trick, Eq. (12) becomes:

$$\rho=\sum_{j=1}^{N} y_j\alpha_j K(x_j,x_i)-y_i\theta_i \tag{13}$$

In practice, to avoid favoring any particular sample, we take average all on the right hand sides of Eq. (13).

D. RSVM Incorporate both SVM and OCSVM

By setting $\mu=1$, the optimization problem in Eq. (10) becomes:

$$\begin{gathered}\min_{\alpha}\ \left(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j x_i^T x_j\alpha_i\alpha_j-\sum_{i=1}^{N}\alpha_i\right)\\ s.t.:\ \sum_{i=1}^{N} y_i\alpha_i=0\\ \forall_{i=1}^{N}: 0\le\alpha_i\le C\end{gathered} \tag{14}$$

The optimization problem in Eq. (14) coincides with that of SVM.

On the other hand, by setting $\mu=0$, the optimization problem in Eq. (10) becomes:

$$\begin{gathered}\min_{\alpha}\ \left(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K(x_i,x_j)\alpha_i\alpha_j-\sum_{i=1}^{N}\theta_i\alpha_i\right)\\ s.t.:\ \sum_{i=1}^{N} y_i\alpha_i=0\\ \forall_{i=1}^{N}: 0\le\alpha_i\le C\end{gathered} \tag{15}$$

where $\theta_i=\frac{1-y_i}{2}$.

The fact that the positive margin, i.e., m^+, is 0 implies that the negative margin or the margin, i.e., $m^-=m$, which is measured by the distance from the closest negative data sample to the hyperplane, is maximized. Although the optimization problem in Eq. (15) is not that of OCSVM, however it can certainly be used for novelty detection and is somehow better than OCSVM since it enables the use of the negative data samples, while OCSVM regards all negative data samples as a common symbol and obviously cannot efficiently take the advantage from these data.

V. Calculating the Adjustment Factor μ

A. Problem Statement

In this section, we discuss how to calculate *the optimal value of the adjustment factor* μ for RSVM. The value of μ can be obtained by solving the following optimization problem:

$$\begin{gathered}\min_{w,\rho,\mu,\xi}\ \left(\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i\right)\\ s.t.:\ \forall_{i=1}^{N} y_i=1: y_i\left(w^T x_i-\rho\right)\ge\frac{2\mu}{\mu+1}-\xi_i\\ \forall_{i=1}^{N} y_i=-1: y_i\left(w^T x_i-\rho\right)\ge\frac{2}{\mu+1}-\xi_i\\ \forall_{i=1}^{N}:\xi_i\ge 0\\ \mu\ge 0\end{gathered} \tag{16}$$

where μ is regarded as a variable.

B. The Solution

We apply KKT theorem to derive the optimization problem in Eq. (16). The Lagrange function is of the following form:

$$L\left(w,\rho,\mu,\xi,\alpha,\beta,\gamma\right)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{N}\xi_i$$
$$-\sum_{i=1}^{N}\alpha_i\left(y_i\left(w^T x_i-\rho\right)-\theta_i+\xi_i\right)-\sum_{i=1}^{N}\beta_i\xi_i-\gamma\mu$$

Setting the derivatives to 0, we gain:

$$\frac{\delta L}{\delta w}=0\rightarrow w=\sum_{i=1}^{N} y_i\alpha_i x_i \tag{17}$$

$$\frac{\delta L}{\delta\rho}=0\rightarrow\sum_{i=1}^{N} y_i\alpha_i=0 \tag{18}$$

$$\forall_{i=1}^{N}:\frac{\delta L}{\delta\xi_i}=0\rightarrow\alpha_i+\beta_i=C \tag{19}$$

$$\frac{\delta L}{\delta\mu}=0\rightarrow\sum_{y_i=1}\frac{2\alpha_i}{(\mu+1)^2}-\sum_{y_i=-1}\frac{2\alpha_i}{(\mu+1)^2}-\gamma=0 \tag{20}$$

$$\forall_{i=1}^{N}:\alpha_i\ge 0,\ y_i\left(w^T x_i-\rho\right)-\theta_i+\xi_i\ge 0$$
$$,\alpha_i\left(y_i\left(w^T x_i-\rho\right)-\theta_i+\xi_i\right)=0 \tag{21}$$

$$\forall_{i=1}^{N}:\beta_i\ge 0,\ \xi_i\ge 0,\ \beta_i\xi_i=0 \tag{22}$$

$$\gamma\mu=0 \tag{23}$$

Eqs. (18, 20) imply that $\gamma=0$.

Substituting Eqs. (17-20) to the Lagrange function, we achieve:

$$\begin{aligned}L&=-\frac{1}{2}\|w\|^2+\sum_{i=1}^{N}\theta_i\alpha_i\\ &=-\frac{1}{2}\|w\|^2+\sum_{y_i=1}\left(2-\frac{2}{\mu+1}\right)+\sum_{y_i=-1}\frac{2}{\mu+1}\\ &=-\frac{1}{2}\|w\|^2+2\sum_{y_i=1}\alpha_i\\ &=-\frac{1}{2}\|w\|^2+\sum_{i=1}^{N}\alpha_i\end{aligned}$$

since Eq. (18) implies that $\sum_{y_i=1}\alpha_i = \sum_{y_i=-1}\alpha_i$.

We end up with the following optimization problem:

$$\begin{array}{c} \min\limits_{\alpha} \left(\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j x_i^T x_j \alpha_i \alpha_j - \sum_{i=1}^{N}\alpha_i\right) \\ s.t.: \sum_{i=1}^{N} y_i \alpha_i = 0 \\ \forall_{i=1}^{N} : 0 \le \alpha_i \le C \end{array}$$

C. The Decision Function

There are two margins which are specified by the equations: $(\mathcal{H}_+) : w^T x - \rho = \frac{2\mu}{\mu+1}$ and $(\mathcal{H}_-) : w^T x - \rho = -\frac{2}{\mu+1}$. To find out the optimal decision hyperplane $(\mathcal{H}) : w^T x - \rho = 0$, we only need to consider the data samples which locate inside the strip formed by the two margins and propose the decision hyperplane such that the empirical error caused by those is minimal. To illuminate it in more details, we enumerate the possible cases:

Case 1. $0 < \alpha_i < C$
- $y_i = 1 : w^T x_i - \rho = \frac{2\mu}{\mu+1}$. Let us call such an index by i_1 and we have $w^T x_{i_1} - \rho = \frac{2\mu}{\mu+1}$.
- $y_i = -1 : w^T x_i - \rho = -\frac{2}{\mu+1}$. Let us call such an index by i_2 and we have $w^T x_{i_2} - \rho = -\frac{2}{\mu+1}$.

Case 2. $\alpha_i = 0$
$y_i(w^T x_i - \rho) \ge \theta_i$ implies that x_i is correctly classified and resides outside the margins. We can safely ignore them.

Case 3. $\alpha_i = C$
$y_i(w^T x_i - \rho) = \theta_i - \xi_i$. We can decide whether x_i locates inside the margins by verifying as follows:
- $y_i = 1 : w^T x_i - \rho \ge -\frac{2}{\mu+1} = w^T x_{i_2} - \rho$ or $w^T x_i \ge w^T x_{i_2}$ (condition 1).
- $y_i = -1 : w^T x_i - \rho \le \frac{2\mu}{\mu+1} = w^T x_{i_1} - \rho$ or $w^T x_i \le w^T x_{i_1}$ (condition 2).

Let's denote J the set of all indices whose data samples locate inside the margins. We can use the conditions 1 and 2 to construct J as follows:

$$\begin{aligned} J = &\left\{i : \alpha_i = C \wedge y_i = 1 \wedge w^T x_i \ge w^T x_{i_2}\right\} \\ &\cup \left\{i : \alpha_i = C \wedge y_i = -1 \wedge w^T x_i \le w^T x_{i_1}\right\} \end{aligned} \quad (24)$$

We propose to choose ρ_{opt} and $\mu_{opt} = -\frac{2}{w^T x_{i_2} - \rho_{opt}} - 1$ such that the empirical error caused by the data samples whose indices are in J is minimal.

$$\rho_{opt} = \underset{\rho}{argmin} \sum_{j\in J} l\left(y_j(w^T x_j - \rho)\right) \quad (25)$$

where $l(o) = \begin{cases} 0 & if\ o \ge 0 \\ 1 & if\ o < 0 \end{cases}$ is the empirical loss function.

In many real world applications, the cost suffered by classifying the negative data samples as the positive ones is different from that of classifying the positive data samples as the negative samples. To handle this problem, we consider *the relative cost* λ, the difference between the former and later, and find ρ_{opt} by:

$$\rho_{opt} = \underset{\rho}{argmin}\left(\underset{\rho}{argmin}\sum_{j\in J}\lambda_i l\left(y_j(w^T x_j - \rho)\right)\right) \quad (26)$$

where $\lambda_i = \begin{cases} 1 & if\ y_i = 1 \\ \lambda & if\ y_i = -1 \end{cases}$.

The algorithm for determining ρ_{opt} is proposed as follows:

Algorithm 1 Determining the optimal decision hyperplane.

Form the set $J = \{j_1, j_2, \ldots, j_k\}$ *as defined in Eq. (24). Arrange the objective values of the data samples whose indices are in* J *in descending order, assume that* $w^T x_{j_1} \ge w^T x_{j_2} \ge \ldots \ge w^T x_{j_k}$.

$j_0 = i_1$; //$y_{j_0=1}$
$t = 0$;
$nCorrectMax = \lambda card\left(\{y_{j_t} : y_{j_t} = -1\}\right)$; /*at first all negative data sample in J are correctly classified */
while$(y_{j_t} = 1)$ {t++; $nCorrectMax$ ++;}
$pos = t - 1$;
$nCorrect = nCorrectMax$;
do{
  while$(y_{j_t} = -1)$ { t++; $nCorrect -= \lambda$; }
  while$(y_{j_t} = 1)$ {t++; $nCorrect$ ++;}
  if$(nCorrect > nCorrectMax)$ {
    $nCorrectMax = nCorrect$;
    $pos = t - 1$; }
}**while**$(t \le k)$
$\rho_{opt} = \frac{w^T x_{j_{pos}} + w^T x_{j_{pos+1}}}{2}$;

The idea behind *Algorithm 1* is that we traverse all indices $t\,(1 \le t \le k)$ where $y_{j_t} = 1$ and $y_{j_{t+1}} = -1$, and count the number of correctly classified data samples with the weight λ between the correctly and incorrectly classified negative and positive data samples with the assumption that the hyperplane $w^T x - \rho = 0$, where $\rho = \frac{w^T x_{j_t} + w^T x_{j_{t+1}}}{2}$ is used as the classifier and then choose the case that offers the highest accuracy. The cost of *Algorithm 1* includes the cost to arrange an array of size $k = card(J)$ and the cost to choose ρ_{opt}, which is obviously $O(k)$. Therefore, the total cost of *Algorithm 1* is around $O(klnk + k)$. This cost is not expensive at all because k , i.e., the cardinality of the set J, is always very small as compared to the size of the training set.

VI. The Experiments

A. The Experiments on the Toy datasets

1) The Experiments with the Linear Kernel: To visually explain the behaviors of RSVM, we first conducted the experiments on 2-D toy datasets. The linear kernel was employed and the parameter μ was varied in the grid $\{0, 0.5, 1, 15\}$. When $\mu = 0$ as shown in Figure 1, the margin, which is the distance from the closest negative data sample, is maximized and the optimal hyperplane is pushed as close as possible to the positive region ($m^+ = 0$ and $m^- = m$ is maximized). It is obvious that this setting can be used for one-class classification to classify imbalanced datasets. When $\mu = 1$ as shown in

Figure 2, two margins are equal ($m^+ = m^-$), this setting corresponds to SVM and is used for classifying balanced datasets. In Figure 3, μ was set to 0.5 and the positive margin is a half of the negative margin. This setting is good for imbalanced datasets with a certain level of deviation. In Figure 4, μ was set to 15 and the negative margin is very small.

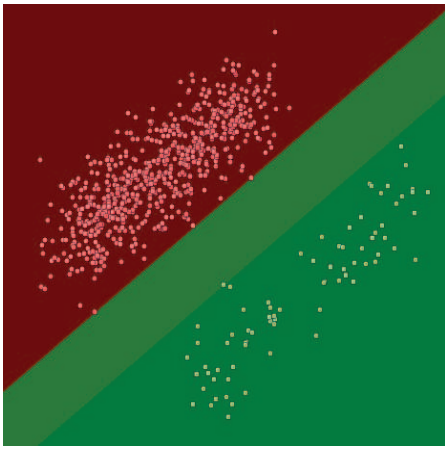

Fig. 1. Experiment with linear kernel and $\mu = 0, C = 30$.

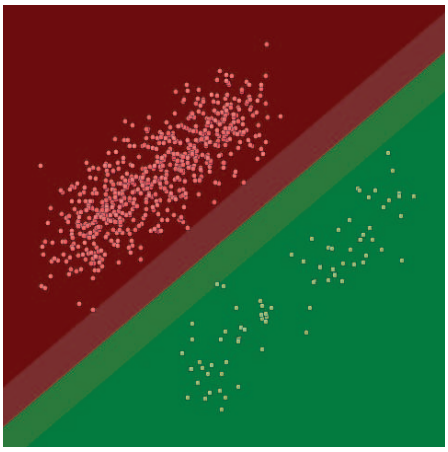

Fig. 2. Experiment with linear kernel and $\mu = 1, C = 30$.

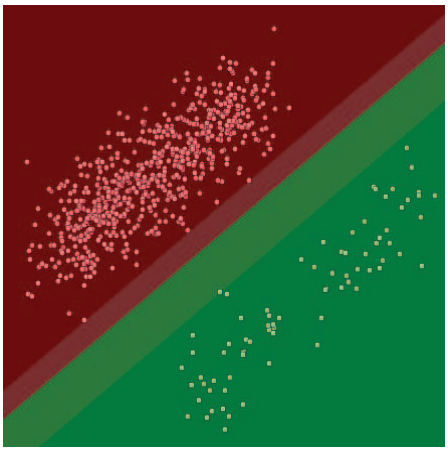

Fig. 3. Experiment with linear kernel and $\mu = 0.5, C = 30$.

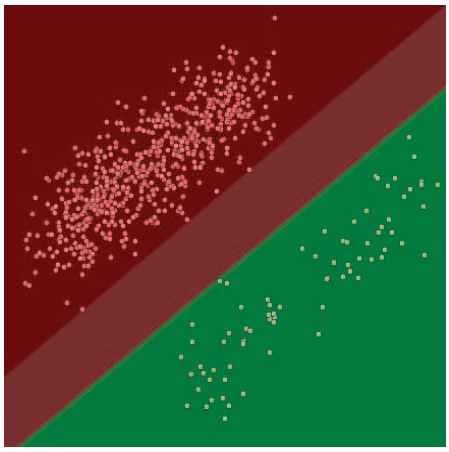

Fig. 4. Experiment with linear kernel and $\mu = 15, C = 30$.

2) The Experiments with RBF Kernel: The experiments were also conducted on the same 2-D datasets. The first dataset is imbalanced and was mainly drawn from the mixture of three Gaussian distributions together with some negative data samples. For the task of learning the data description of the positive class to detect the divergence from normality, μ was set to 0 and 0.5, respectively. As seen in Figure 5, when $\mu = 0$, the classifier can perfectly recognize the mixture of three Gaussian distributions. In Figure 6, $\mu = 0.5$, the positive margin m^+ becomes larger and thereby connect two Gaussian distributions together. The second dataset is fairly balanced and includes some Gaussian distributions inside it. For the task of classifying the positive and negative classes, μ was set to 1 and 15. As shown in Figure 7, the classifier classifies well the dataset when $\mu = 1$. In Figure 8, when $\mu = 15$, the negative margin m^- is very thin. To conclude, it is fair to claim that RSVM can handle well all kinds of datasets. Furthermore, it also offers more insights to fit the real datasets.

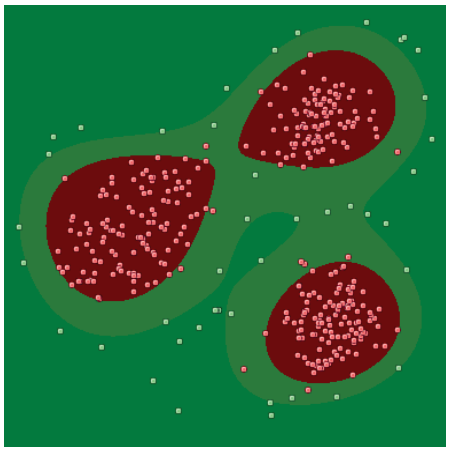

Fig. 5. Experiment with RBF kernel and $\mu = 0, C = 30, \gamma = 2$.

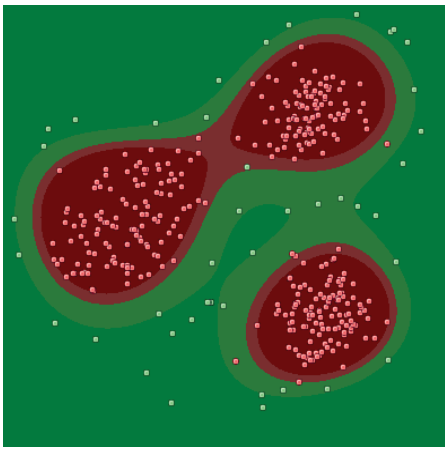

Fig. 6. Experiment with RBF kernel and $\mu = 0.5, C = 30, \gamma = 2$.

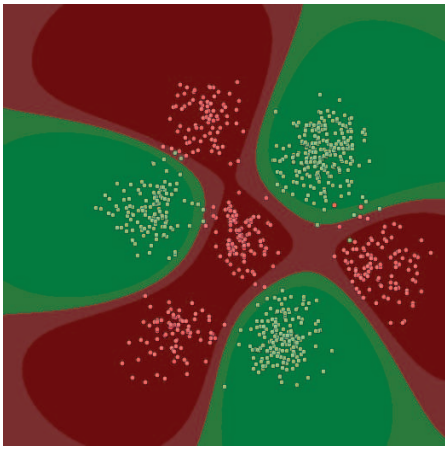

Fig. 7. Experiment with RBF kernel and $\mu = 1, C = 30, \gamma = 2$.

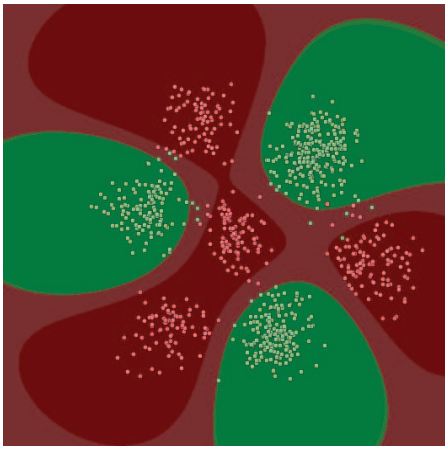

Fig. 8. Experiment with RBF kernel and $\mu = 15, C = 30, \gamma = 2$.

3) The Experiments on Calculating the Adjustment Factor: To demonstrate how to calculate *the optimal adjustment factor*, we conducted the experiments on two other 2-D toy datasets. In these experiments, the linear kernel was employed. In Figure 9, *the relative cost* λ was set to 1, i.e., the positive and negative data samples are identically treated, the margin is adjusted to minimize the empirical lost caused by the data samples inside two margins consequently. In Figure 10, *the relative cost* λ was set to $\frac{\#Positive}{\#Negative}$. Because the currently experimental dataset is imbalanced, where the positive class is the majority, the negative data samples are favored, and as the consequence, the decision hyperplane is pushed closer to the positive region to reduce the empirical error on the inside-margin negative data samples.

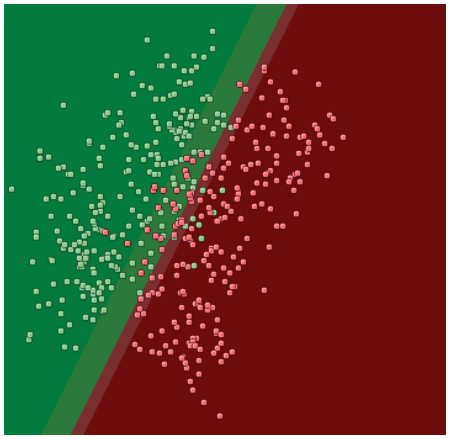

Fig. 9. Experiment with linear kernel and $\lambda = 1, C = 30$.

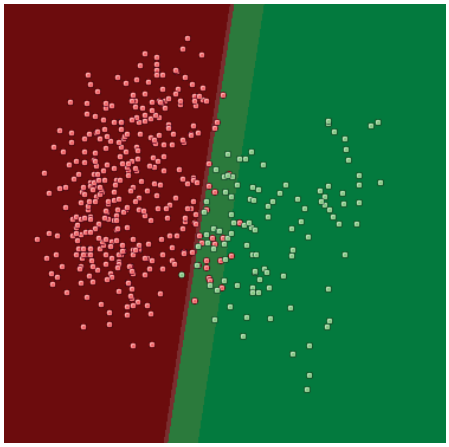

Fig. 10. Experiment with linear kernel and $\lambda = {}^{\#Positive}/_{\#Negative}, C = 30$.

B. The Experiments on Real Datasets

1) The Experimental Settings: To demonstrate the ability of the proposed method, we conducted the experiments on *the benchmark datasets of UCI repository*. The methods in comparison are *SVM, SVDD*, *OCSVM*, and our proposed *RSVM*. We designed the experiments for both two-classes and one-class classifications.

RBF kernel given by $K(x, x') = e^{-\gamma \|x - x'\|^2}$ was used in the experiments. The *width of kernel* γ was varied in the grid $\left\{2^{-15}, 2^{-13}, \ldots, 2^3, 2^5\right\}$. The *trade-off parameter* C was searched in the grid $\left\{2^{-15}, 2^{-13}, \ldots, 2^3, 2^5\right\}$. For RSVM, *the adjustment factor* μ was searched in the grid

$\{0, 0.2, 0.4, 0.6, 0.8, 1, 1.2, 20\}$. To investigate how good the procedure for calculating *the adjustment parameter* μ is, we computed the optimal values two ways by setting *the relative cost* λ by 1 and $\frac{\#Positve}{\#Negative}$, respectively. We name two these specific cases by *Optimal RSVM 1* (OR1) and *Optimal RSVM 2* (OR2), respectively. The cross-validation with five folds was used. The accuracy was measured by $acc = \frac{acc^+ + acc^-}{2}$ where $acc^+ = \%TP$ and $acc^- = \%TN$ are the accuracies on the positive and negative classes, respectively. This measure is appropriate since it insures the high accuracies for both positive and negative classes to ensure the high accuracy for the entire dataset.

In the experiments for one-class classification, we preprocessed the datasets by: 1) appointing one class as the normal class; 2) recommending the rest as the abnormal class; 3) randomly selecting the data samples in the abnormal class such that the ratio of the data in the normal and abnormal classes is 10:1.

In the experiments for two-classes classification, we preprocessed the datasets by: 1) choosing one class as the positive class; 2) choosing one another class as the negative class.

2) The Experiments on Imbalanced Datasets: The experimental results on the imbalanced are shown in Table I and Figure 11. To increase the readability, we emphasized in bold the methods that result in the highest accuracy for each dataset. As shown in Table I and Figure 11, our proposed methods *RSVM* and its variations *OR1* and *OR2* always surpass other methods on all experimental datasets. To examine how good the procedure of calculating *the adjustment factor* μ is, we focus on the proposed methods *RSVM, OR1*, and *OR2*. It appears that *OR1* wins *RSVM* on the three datasets whereas *OR2* wins *RSVM* on the five datasets. On the remaining datasets, both *OR1* and *OR2* are slightly lower than RSVM. In the comparison of *OR1* and *OR2*, it can be seen that *OR2* is better than *OR1*. It is is not unexpected because the *relative cost* λ of *OR2* is 10 and thus it favors the negative data samples and thereby encouraging the magnitude of the negative accuracy acc^-.

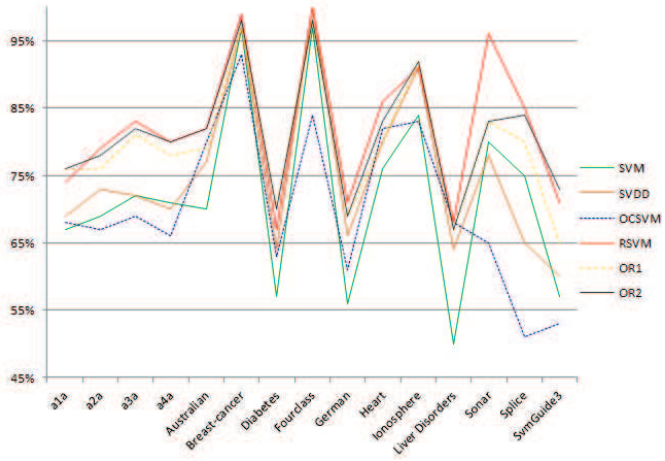

Fig. 11. The experimental results on the imbalanced datasets.

3) The Experiments on Balanced Datasets: The experimental results on balanced datasets are shown in Table II and Figure 12. As shown in II and Figure 12, our proposed *RSVM*, *OR1*, and *OR2* win over the others on all experimental datasets. *RSVM* is the best on 11 out of 15 datasets. *OR1* wins on 4 datasets whereas *OR2* wins on 9 datasets. Although, in *OR1* and *OR2*, we scan only one value for *the adjustment factor* μ, they are still comparable with *RSVM*. According to the experimental results, *OR2* performs better than *OR1*.

TABLE I
THE EXPERIMENTAL RESULTS ON THE IMBALANCED DATASETS (*OC means OCSVM*).

Datasets	SVM	SVDD	OC	RSVM	OR1	OR2
a1a	67%	69%	68%	74%	**76%**	**76%**
a2a	69%	73%	67%	**79%**	76%	78%
a3a	72%	72%	69%	**83%**	81%	82%
a4a	71%	70%	66%	**80%**	78%	**80%**
Australian	70%	77%	80%	**82%**	79%	**82%**
BC	97%	97%	93%	**99%**	97%	98%
Diabetes	57%	64%	63%	67%	**70%**	**70%**
Fourclass	97%	**100%**	84%	**100%**	98%	98%
German	56%	66%	61%	**71%**	69%	69%
Heart	76%	80%	82%	**86%**	80%	83%
Ionosphere	84%	91%	83%	91%	**92%**	**92%**
Liver Disorders	50%	64%	**68%**	**68%**	67%	67%
Sonar	80%	78%	65%	**96%**	83%	83%
Splice	75%	65%	51%	**85%**	80%	84%
SvmGuide3	57%	60%	53%	71%	65%	**73%**

TABLE II
THE EXPERIMENTAL RESULTS ON THE IMBALANCED DATASETS.

Datasets	SVM	SVDD	OC	RSVM	OR1	OR2
a1a	74%	72%	55%	79%	78%	**81%**
a2a	74%	71%	67%	**79%**	76%	78%
a3a	79%	69%	69%	**83%**	81%	82%
a4a	78%	70%	66%	**80%**	78%	**80%**
Australian	86%	81%	78%	86%	**88%**	**88%**
BC	97%	97%	89%	**98%**	**98%**	**98%**
Diabetes	72%	67%	60%	**76%**	74%	74%
Fourclass	**100%**	97%	66%	**100%**	**100%**	**100%**
German	68%	64%	53%	**72%**	71%	**72%**
Heart	84%	76%	61%	**85%**	84%	84%
Ionosphere	**96%**	92%	80%	**96%**	95%	95%
LD	65%	65%	58%	65%	66%	**69%**
Sonar	67%	64%	55%	67%	68%	**72%**
Splice	**87%**	71%	52%	**87%**	**87%**	**87%**
SG3	83%	79%	79%	**84%**	83%	82%

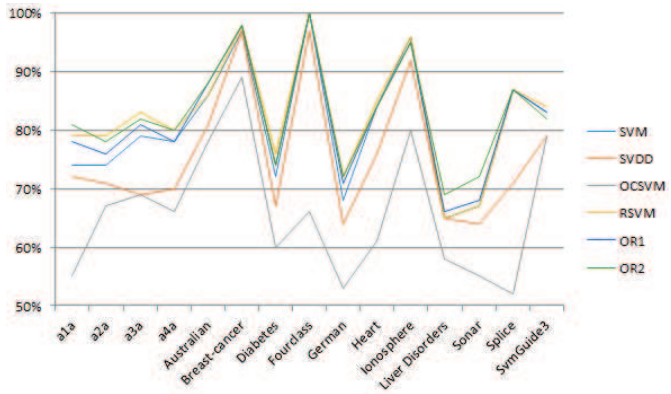

Fig. 12. The experimental results on the imbalanced datasets.

VII. Conclusion

In this paper, we propose Robust Support Vector Machine (RSVM), where the optimal hyperplanes can be adjusted to fit the profiles of the datasets. In RSVM, two margins are adjustable by a certain ratio μ, called *the adjustment factor*. By setting the appropriate values for μ, RSVM can adjust well to both the balanced and imbalanced datasets. We also suggest, in this paper, the procedure to calculate the optimal value for μ. The experiments conducted on 15 benchmark datasets of UCI repository demonstrate the superiority of our proposed method.

References

[1] C. Cortes and V. Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[2] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[3] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[4] ——, *The Nature of Statistical Learning Theory*, 2nd ed. Springer, 1999.

[5] M. I. Jordan, *An Introduction to Probabilistic Graphical Models*. In preparation, 2003.

[6] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[7] D. Tax and R. P. W. Duin, "Support vector data description," *Journal of Machine Learning Research*, vol. 54, no. 1, pp. 45–66, 2004.

[8] D. Tax and R. Duin, "Support vector domain description," *Pattern Recognition Letters*, vol. 20, pp. 1191–1199, 1999.