# Rough Margin Based Core Vector Machine

Gang Niu[1], Bo Dai[2], Lin Shang[1], and Yangsheng Ji[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, P.R.China
{niugang,jiyangsheng,lshang}@ai.nju.edu.cn

[2] NLPR/LIAMA, Institute of Automation, Chinese Academy of Science, Beijing 100190, P.R.China
bdai@nlpr.ia.ac.cn

**Abstract.** [1]The recently proposed rough margin based support vector machine (RMSVM) could tackle the overfitting problem due to outliers effectively with the help of rough margins. However, the standard solvers for them are time consuming and not feasible for large datasets. On the other hand, the core vector machine (CVM) is an optimization technique based on the minimum enclosing ball that can scale up an SVM to handle very large datasets. While the 2-norm error used in the CVM might make it theoretically less robust against outliers, the rough margin could make up this deficiency. Therefore we propose our rough margin based core vector machine algorithms. Experimental results show that our algorithms hold the generalization performance almost as good as the RMSVM on large scale datasets and improve the accuracy of the CVM significantly on extremely noisy datasets, whilst cost much less computational resources and are often faster than the CVM.

## 1 Introduction

People in computer science societies have been questing for faster algorithms since long before. When come to mind the solving techniques of SVMs, there are several approaches ranged from the chunking method [1] to the sequential minimal optimization [2], as well as scaling down the training data and low-rank kernel matrix approximations [3]. Eventually, the *Core Vector Machine* (CVM) algorithms [4, 5, 6, 7] have gone to an extreme that they have linear asymptotic time complexity and constant asymptotic space complexity, since they transform the *quadratic programming* (QP) involved in SVMs to the *minimum enclosing ball* (MEB) problems. In order to perform this transformation the CVM takes the 2-norm error, which may cause it less robust and thus hurt the accuracy. Fortunately the notion of the rough margin in the *Rough Margin based Support Vector Machine* (RMSVM) [8] could make SVMs less sensitive to noises and outliers, and consequently reduce the negative effects of outliers. For this reason we propose our *Rough Margin based Core Vector Machine* (RMCVM), which unites the merits of the two aforementioned methods.

[1] Supported by National Natural Science Foundation of China (No. 60775046) and Natural Science Foundation of Jiangsu Province (No. BK2009233).

M.J. Zaki et al. (Eds.): PAKDD 2010, Part I, LNAI 6118, pp. 134–141, 2010.

After brief introductions to the CVM and the RMSVM in Section 2 and 3, we will first of all define the primal problem of the RMCVM. Next we shall elaborate how to solve an RMCVM through the approximate MEB finding algorithm. We will also investigate the loss functions used by the RMSVM and RMCVM. In the end experimental results are shown in Section 5.

## 2 Core Vector Machine with Minimum Enclosing Ball

Given a set of points $S = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ for some integer $d$, the minimum enclosing ball of $S$ (denoted MEB($S$)) is the smallest ball which contains all the points in $S$ [5]. Formally, let $\varphi$ be a kernel induced feature map,

$$\min_{R,\mathbf{c}} R^2 \quad \text{s.t. } \|\mathbf{c} - \varphi(\mathbf{x}_i)\|^2 \le R^2,\ i = 1, \dots, m. \tag{1}$$

Let $B(\mathbf{c}^*, R^*)$ be the exact MEB($S$). Given an $\epsilon > 0$, a $(1+\epsilon)$-approximation of $B(\mathbf{c}^*, R^*)$ is a ball $B(\mathbf{c}, (1+\epsilon)R)$ such that $R \le R^*$ and $S \subset B(\mathbf{c}, (1+\epsilon)R)$. A subset $Q$ of $S$ is called a core-set, if $S \subset B(\mathbf{c}, (1+\epsilon)R)$ where $B(\mathbf{c}, R)$ is MEB($Q$). The approximate MEB finding algorithm [9] uses a simple iterative scheme: at the $t$-th iteration, $Q_t$ is expanded by including the farthest point of $S$ from $\mathbf{c}_t$, then optimize (1) to get $B(\mathbf{c}_{t+1}, R_{t+1})$; this is repeated until $S \subset B(\mathbf{c}_t, (1+\epsilon)R_t)$. A surprising property is that the number of iterations, and thus the size of the final core-set, depend only on $\epsilon$ but not on $d$ or $m$ [9, 5].

The dual of (1) is $\max_{\boldsymbol{\alpha}} \boldsymbol{\alpha}^\top \mathrm{diag}(\mathbf{K}) - \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}$ s.t. $\boldsymbol{\alpha} \ge \mathbf{0}, \boldsymbol{\alpha}^\top \mathbf{1} = 1$, where $\boldsymbol{\alpha}$ is the Lagrange multiplier and $\mathbf{K}$ is the kernel matrix. Conversely, any QP of this form can be regarded as an MEB problem [4]. In particular when

$$k(\mathbf{x}, \mathbf{x}) = \kappa, \tag{2}$$

where $\kappa$ is a constant (this is true for many popular kernels), we can drop the linear term $\boldsymbol{\alpha}^\top \mathrm{diag}(\mathbf{K})$ and obtain a simpler QP,

$$\max_{\boldsymbol{\alpha}} -\boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha} \quad \text{s.t. } \boldsymbol{\alpha} \ge \mathbf{0},\ \boldsymbol{\alpha}^\top \mathbf{1} = 1. \tag{3}$$

**Definition 1 (CVM [4])**

$$\min_{\mathbf{w},b,\boldsymbol{\xi},\rho} \|\mathbf{w}\|^2 + b^2 - 2\rho + C \sum_{i=1}^{m} \xi_i^2 \quad \text{s.t. } y_i(\mathbf{w}^\top \varphi(\mathbf{x}_i) + b) \ge \rho - \xi_i,\ i = 1, \dots m, \tag{4}$$

*where $C$ is a regularization parameter and $\xi_i$ are slack variables.*

The dual of (4) is analogous to the dual (3), in which $\mathbf{K}$ is replaced with $\tilde{\mathbf{K}}$ that $\tilde{\mathbf{K}}_{ij} = \tilde{\varphi}(\mathbf{x}_i, y_i)^\top \tilde{\varphi}(\mathbf{x}_j, y_j)$ where $\tilde{\varphi}(\mathbf{x}_i, y_i) = [y_i \varphi(\mathbf{x}_i)^\top, y_i, \frac{1}{\sqrt{C}} \mathbf{e}_i^\top]^\top$ ($\mathbf{e}_i$ is all 0 except that the $i$-th component is 1). Hence the CVM is an MEB if (2) is true. To deal with the situation that (2) is not satisfied, Tsang et al. extend the MEB to the center-constrained MEB [10], and propose the generalized core vector machine [11] which is applicable for any kernel and can also be applied to kernel methods such as SVR and ranking SVM.

Note that the 2-norm error is used here. It could be less robust in the presence of outliers in theory to some extent [5].

## 3 Rough Margin Based Support Vector Machine

The rough set theory, which is based on the concept of the lower and upper approximation of a set, is a mathematical tool to cope with uncertainty and incompleteness [12]. The rough margins [8] are expressed as a lower margin $\frac{2\rho_l}{\|\mathbf{w}\|}$ and an upper margin $\frac{2\rho_u}{\|\mathbf{w}\|}$ where $0 \leq \rho_l \leq \rho_u$. They are corresponding with the lower and upper approximations of the outlier set, such that the samples in the lower margin are considered as outliers, the samples outside the upper margin are not outliers, and the samples between two rough margins are possibly outliers.

When the training procedure takes place, the RMSVM tries to give major penalty to samples lying within the lower margin, and give minor penalty to other samples [8]. Notice that the *Universum SVM* [13] uses a similar strategy. In practice, the RMSVM introduces slack variables $\zeta_i$ and penalize them $\tau$ times larger than $\xi_i$, since $\zeta_i > 0$ means that $\varphi(\mathbf{x}_i)$ is in the lower margin. Formally,

**Definition 2 (RMSVM [8])**

$$\begin{aligned} \min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\zeta},\rho_l,\rho_u} \quad & \frac{1}{2}\|\mathbf{w}\|^2 - \nu\rho_l - \nu\rho_u + \frac{1}{m}\sum_{i=1}^{m}\xi_i + \frac{\tau}{m}\sum_{i=1}^{m}\zeta_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top\varphi(\mathbf{x}_i) + b) \geq \rho_u - \xi_i - \zeta_i, \\ & 0 \leq \xi_i \leq \rho_u - \rho_l,\ \zeta_i \geq 0,\ \rho_l \geq 0,\ \rho_u \geq 0,\ i = 1, \dots m, \end{aligned} \tag{5}$$

*where* $\nu \in (0, 1), \tau > 1$ *are regularization parameters and* $\xi_i, \zeta_i$ *are slack variables.*

### 3.1 Justification of the Rough Margin

Apparently the RMSVM should encounter severer overfitting problem since it emphasizes more on outliers than the $\nu$-SVM. However, the dual of (5) is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2}\boldsymbol{\alpha}^\top\mathbf{Q}\boldsymbol{\alpha} \quad \text{s.t. } \mathbf{y}^\top\boldsymbol{\alpha} = 0,\ \boldsymbol{\alpha}^\top\mathbf{1} \geq 2\nu,\ 0 \leq \alpha_i \leq \frac{\tau}{m}, i = 1, \dots m, \tag{6}$$

where $\mathbf{Q}_{ij} = y_i y_j \varphi(\mathbf{x}_i)^\top\varphi(\mathbf{x}_j)$. Hence $2\nu/\tau$ is the fraction of samples permitted to lie within the lower margin. For fixed $\nu$, the larger the $\tau$ is, the less overfitting the RMSVM is, and ultimately the RMSVM would become an underfitting classifier.

Likewise the loss function, which was an absent topic in [8], could justify the rough margin well. Let $f(\mathbf{x}_i) = \mathbf{w}^\top\varphi(\mathbf{x}_i) + b$,

**Proposition 1.** *The loss function of the RMSVM is*

$$L_1^{\rho_l,\rho_u}(\mathbf{x}_i, y_i, f) = \begin{cases} \rho_u + (\tau - 1)\rho_l - \tau y_i f(\mathbf{x}_i) & , y_i f(\mathbf{x}_i) \leq \rho_l \\ \rho_u - y_i f(\mathbf{x}_i) & , \rho_l < y_i f(\mathbf{x}_i) \leq \rho_u \\ 0 & , y_i f(\mathbf{x}_i) > \rho_u \end{cases}$$

*Proof.* (Sketch) In fact $\xi_i$ increases before $\zeta_i$, while $\zeta_i$ has to stay at zero until $\xi_i$ arrives at $\rho_u - \rho_l$, since $\xi_i$ suffers less penalty in (5). □

In other words, though $\frac{1}{m} < \alpha_i \le \frac{\tau}{m}$ when $y_i f(\mathbf{x}_i) \le \rho_l$, $\rho_l$ is smaller than $\rho$, the loss $L_1^{\rho_l,\rho_u}(\mathbf{x}_i, y_i, f)$ may still not be large, and the number of samples satisfying $y_i f(\mathbf{x}_i) \le \rho_l$ is usually smaller than the number of samples satisfying $y_i f(\mathbf{x}_i) \le \rho$ in the $\nu$-SVM. Therefore the notion of the rough margin is a tool and technique to avoid the overfitting problem.

There is a side effect that $\rho_u$ is usually larger than $\rho$, which makes the RMSVM generate much more support vectors satisfying $\rho_l < y_i f(\mathbf{x}_i) \le \rho_u$ such that $\varphi(\mathbf{x}_i)$ lies between two rough margins with tiny $\alpha_i$. This phenomenon slows down the speed and increases the storage of the RMSVM.

## 4 Rough Margin Based Core Vector Machine

For the sake of using the approximate MEB finding algorithm to solve the rough margin based SVM, we use 2-norm error because it allows a soft-margin L2-SVM to be transformed to a hard-margin one. Subsequently we have

**Definition 3 (RMCVM)**

$$
\begin{aligned}
&\min_{\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\zeta},\rho_l,\rho_u} \ \|\mathbf{w}\|^2 + b^2 - 2\rho_l - 2\rho_u + C\sum_{i=1}^{m}\xi_i^2 + \tau C\sum_{i=1}^{m}\zeta_i^2 \\
&\text{s.t.} \quad y_i(\mathbf{w}^\top\varphi(\mathbf{x}_i) + b) \ge \rho_u - \xi_i - \zeta_i,\ \xi_i \le \rho_u - \rho_l,\ i = 1, \dots m,
\end{aligned}
\tag{7}
$$

*where $C > 0, \tau > 1$ are regularization parameters and $\xi_i, \zeta_i$ are slack variables. The dual of (7) is*

$$
\begin{aligned}
&\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} \ -\boldsymbol{\alpha}^\top\left(\mathbf{K}\circ\mathbf{y}\mathbf{y}^\top + \mathbf{y}\mathbf{y}^\top + \frac{1}{\tau C}\mathbf{I}\right)\boldsymbol{\alpha} - \frac{1}{C}\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|^2 \\
&\text{s.t.}\ \boldsymbol{\alpha}^\top\mathbf{1} = 2,\ \boldsymbol{\beta}^\top\mathbf{1} = 1,\ \boldsymbol{\alpha} \ge \mathbf{0},\ \boldsymbol{\beta} \ge \mathbf{0},
\end{aligned}
\tag{8}
$$

*where $\mathbf{y} = [y_1, \dots, y_m]^\top$ and the operator $\circ$ denotes the Hadamard product.*

*Remark 1.* We omit the constraint $\zeta_i \ge 0$ since it is dispensable for L2-SVMs. We omit the constraints $\rho_l, \rho_u \ge 0$ based on the fact that certain inequality constraints in the dual problem can be replace by the corresponding equality constraints [14, 15]. Finally we omit $\xi_i \ge 0$, otherwise there will be $\frac{1}{C}\|\boldsymbol{\alpha}-\boldsymbol{\beta}+\boldsymbol{\gamma}\|^2$ in the objective and additional constraint $\boldsymbol{\gamma} \ge \mathbf{0}$, and the optimal $\boldsymbol{\gamma}^* = \mathbf{0}$ obviously. The constraint $\rho_u \ge \rho_l$ is indeed implicated by (7) already.

*Remark 2.* Note that the regularization parameter of the original SVM [16] and the $\nu$-SVM [17] is $C$ and $\nu$ respectively. In the CVM it is $C$ through which we control the trading off between the flatness and training errors. Since the order of $\|\mathbf{w}\|$ and $\xi_i$ are equal in (4), their coefficients would change simultaneously under scaling, which means that the coefficient of $\rho$ does not influence very much.

*Remark 3.* It is obvious that there is only $\nu$-RMSVM insofar as it applies 1-norm error. Similarly the RMSVM using 2-norm error would be $C$-RMSVM inherently.

Therefore we demonstrate that *Definition 3* is proper. Furthermore,

**Proposition 2.** *The loss function of the RMCVM is*

$$L_2^{\rho_l,\rho_u}(\mathbf{x}_i, y_i, f) = \begin{cases} (\rho_u - \rho_l)^2 + \tau(\rho_l - y_i f(\mathbf{x}_i))^2 & , y_i f(\mathbf{x}_i) \le \frac{\tau+1}{\tau}\rho_l - \frac{1}{\tau}\rho_u \\ \frac{\tau}{\tau+1}(\rho_u - y_i f(\mathbf{x}_i))^2 & , \frac{\tau+1}{\tau}\rho_l - \frac{1}{\tau}\rho_u < y_i f(\mathbf{x}_i) \le \rho_u \\ 0 & , y_i f(\mathbf{x}_i) > \rho_u \end{cases}$$

*Proof.* (Sketch) We have $\xi_i = \tau\zeta_i$ when $0 < \xi_i < \rho_u - \rho_l$, by equalling the partial derivatives of the objective function of (7) w.r.t. $\xi_i$ and $\zeta_i$ respectively. □

### 4.1 Solving Rough Margin Based CVM

From now on we will proof that (7) can be solved approximately by the CVM.

**Proposition 3.** *The RMSVM cannot be transformed to an MEB problem unless we drop the group of constraints $\alpha_i \le \frac{\tau}{m}$ in (6).*

**Lemma 1.** *Given a non negative vector $\boldsymbol{\alpha} \in \mathbb{R}^m$, the optimal value of*

$$\min_{\boldsymbol{\beta}} \|\boldsymbol{\alpha} - \boldsymbol{\beta}\|^2 \quad \text{s.t. } \boldsymbol{\beta}^\top \mathbf{1} = \tfrac{1}{2}\boldsymbol{\alpha}^\top \mathbf{1},\ \boldsymbol{\beta} \ge \mathbf{0} \tag{9}$$

*is between $\frac{1}{4m}(\boldsymbol{\alpha}^\top \mathbf{1})^2$ and $\frac{1}{4}\|\boldsymbol{\alpha}\|^2$.*

*Proof.* (Sketch) The upper bound is quite straightforward by setting $\boldsymbol{\beta} = \frac{1}{2}\boldsymbol{\alpha}$. Let $V = \{\boldsymbol{\beta} : \boldsymbol{\beta}^\top \mathbf{1} = \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{1}\}$, then $V$ consists of a hyperplane in $\mathbb{R}^m$. The lower bound is given by $\min_{\boldsymbol{\beta} \in V} \|\boldsymbol{\alpha} - \boldsymbol{\beta}\| = \frac{|\boldsymbol{\alpha}^\top \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^\top \mathbf{1}|}{\sqrt{\mathbf{1}^\top \mathbf{1}}} = \frac{|\boldsymbol{\alpha}^\top \mathbf{1}|}{2\sqrt{m}}$. □

Actually $\beta_i = 0$ iff $\cot(\boldsymbol{\alpha}, \mathbf{e}_i) \ge \sqrt{2} + 1$. In other words, $\boldsymbol{\beta}$ is even sparser than $\boldsymbol{\alpha}$, which is consistent with that there are less outliers than support vectors.

**Theorem 1.** *The optimum of* (8) *is bounded by*

$$\max_{\boldsymbol{\alpha}} -\boldsymbol{\alpha}^\top \left(\mathbf{K} \circ \mathbf{y}\mathbf{y}^\top + \mathbf{y}\mathbf{y}^\top + \frac{\tau+4}{4\tau C}\mathbf{I}\right)\boldsymbol{\alpha} \quad \text{s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 2,\ \boldsymbol{\alpha} \ge \mathbf{0}, \tag{10}$$

$$\max_{\boldsymbol{\alpha}} -\boldsymbol{\alpha}^\top \left(\mathbf{K} \circ \mathbf{y}\mathbf{y}^\top + \mathbf{y}\mathbf{y}^\top + \frac{\mathbf{1}\mathbf{1}^\top}{4\mathbf{1}^\top \mathbf{1} C} + \frac{1}{\tau C}\mathbf{I}\right)\boldsymbol{\alpha} \quad \text{s.t. } \boldsymbol{\alpha}^\top \mathbf{1} = 2,\ \boldsymbol{\alpha} \ge \mathbf{0}. \tag{11}$$

*Proof.* (Sketch) Denote the objective function of (8) as $\max_{\boldsymbol{\alpha},\boldsymbol{\beta}} -h(\boldsymbol{\alpha}, \boldsymbol{\beta}) = -\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} h_1(\boldsymbol{\alpha}) + h_2(\boldsymbol{\alpha}, \boldsymbol{\beta})$ where $h_1(\boldsymbol{\alpha}) = \boldsymbol{\alpha}^\top \left(\mathbf{K} \circ \mathbf{y}\mathbf{y}^\top + \mathbf{y}\mathbf{y}^\top + \frac{1}{\tau C}\mathbf{I}\right)\boldsymbol{\alpha}$ and $h_2(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{C}\|\boldsymbol{\alpha} - \boldsymbol{\beta}\|^2$. Then

$$\min_{\boldsymbol{\alpha},\boldsymbol{\beta}} h(\boldsymbol{\alpha}, \boldsymbol{\beta}) \iff \min_{\boldsymbol{\alpha}} \left(h_1(\boldsymbol{\alpha}) + \min_{\boldsymbol{\beta}} h_2(\boldsymbol{\alpha}, \boldsymbol{\beta})\right)$$

since $h(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is convex. According to *Lemma 1*, $\min_{\boldsymbol{\beta}} h_2(\boldsymbol{\alpha}, \boldsymbol{\beta})$ for given $\boldsymbol{\alpha}$ is bounded by $\frac{1}{4mC}(\boldsymbol{\alpha}^\top \mathbf{1})^2$ and $\frac{1}{4C}\boldsymbol{\alpha}^\top \boldsymbol{\alpha}$. □

**Corollary 1.** *The RMCVM can be solved approximately using the approximate MEB finding algorithm.*

*Proof.* Let $\mathbf{Q}^{(1)}, \varphi_1, \mathbf{Q}^{(2)}, \varphi_2$ be

$$\mathbf{Q}_{ij}^{(1)} = \varphi_1(\mathbf{x}_i, y_i)^\top \varphi_1(\mathbf{x}_j, y_j),\ \varphi_1(\mathbf{x}_i, y_i) = \left[ y_i \varphi(\mathbf{x}_i)^\top, y_i, \sqrt{\frac{\tau+4}{4\tau C}} \mathbf{e}_i^\top \right]^\top,$$

$$\mathbf{Q}_{ij}^{(2)} = \varphi_2(\mathbf{x}_i, y_i)^\top \varphi_2(\mathbf{x}_j, y_j),\ \varphi_2(\mathbf{x}_i, y_i) = \left[ y_i \varphi(\mathbf{x}_i)^\top, y_i, \frac{1}{2\sqrt{mC}}, \frac{1}{\sqrt{\tau C}} \mathbf{e}_i^\top \right]^\top.$$

Then (10) and (11) can be regarded as MEB problems with parameters

$$R_t = \sqrt{\frac{1}{2} \boldsymbol{\alpha}^\top \mathrm{diag}(\mathbf{Q}^{(t)}) - \frac{1}{4} \boldsymbol{\alpha}^\top \mathbf{Q}^{(t)} \boldsymbol{\alpha}}, \quad \mathbf{c}_t = \frac{1}{2} \sum_{i=1}^{m} \alpha_i \varphi_t(\mathbf{x}_i, y_i), \quad t = 1, 2. \qquad \square$$

The convergence of (10) and (11) are as same as the CVM but we omit the proof here. Hence the approximate RMCVM algorithms based on (10) and (11) have linear asymptotic time complexity and constant asymptotic space complexity. Recall that the RMSVM is slower than the $\nu$-SVM since it generates more support vectors. Surprisingly the RMCVM most often generates less core vectors and are faster than the CVM, even though we solve two MEB problems.

## 5 Experiments

We implement the RMSVM using `LibSVM` [18] and the RMCVM using `LibCVM`. The parameters are fixed to $\tau = 5$ as [8] suggested, and $\epsilon = 10^{-6}$ as [5] recommended. We tune $\nu \in \{0.1, 0.2, \ldots, 0.9\}$ providing it is feasible. The kernel is Gaussian and its width is the better one computed by the default methods of `LibSVM` and `LibCVM`. The computation method of kernel width is kept unchanged over one dataset.

The notation $\mathrm{RC}_l$ stands for the solution $f_l(\mathbf{x}) = \sum \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + b$ from (10) and $\mathrm{RC}_u$ for $f_u(\mathbf{x})$ from (11), where $b = \mathbf{y}^\top \boldsymbol{\alpha}$ respectively. We denote $\mathrm{RC}_{avg}$ as the average solution of them. For multi-class tasks the default one-versus-one strategy is used. The datasets[2] are listed in *Table 1*.

To begin with, we conduct experiments on a small dataset shown in *Table 2*. Our accuracy is almost always higher than the CVM. We find that $\mathrm{RC}_{avg}$ is not always the best and $\mathrm{RC}_u$ is usually better than $\mathrm{RC}_l$. The next are results on large datasets in the top of *Table 3*, where the time for reading input and writing output files is excluded from the training time. Note that the CVM and the RMCVM are very fast on `extended-usps`, on which all the RMSVM fail to give a solution in 24 hours. Moveover, the RMCVM is usually better than the CVM and even beat the RMSVM once on `web`. At last we display results

[2] Download from `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/` and `http://www.cse.ust.hk/~ivor/cvm.html`

**Table 1.** Data sets

| name | #class | #training | #testing | #feature |
|---|---|---|---|---|
| satimage | 6 | 4,435 | 2,000 | 36 |
| web | 2 | 49,749 | 14,951 | 300 |
| ex-usps | 2 | 266,079 | 75,383 | 676 |
| intrusion | 2 | 4,898,431 | 311,029 | 127 |
| SensIT Vehicle: acoustic | 3 | 78,823 | 19,705 | 50 |
| SensIT Vehicle: seismic | 3 | 78,823 | 19,705 | 50 |

**Table 2.** Experimental results on satimage (accuracy (%))

| | $\epsilon = 10^{-5}$ | | | | $\epsilon = 10^{-6}$ | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $C$ | CVM | RC$_l$ | RC$_u$ | RC$_{avg}$ | CVM | RC$_l$ | RC$_u$ | RC$_{avg}$ | $\nu$ | $\nu$-SVM | RMSVM |
| 0.1 | 82.5 | 83.5 | 84.8 | 84.1 | 82.5 | 83.5 | 84.8 | 84.1 | 0.5 | 81.9 | 88.8 |
| 1 | 85.7 | 86.9 | 87.7 | 86.8 | 85.6 | 86.9 | 87.8 | 86.9 | 0.4 | 84.8 | 89.5 |
| 10 | 89.0 | 88.3 | 88.8 | 88.3 | 88.6 | 89.2 | 89.4 | 89.3 | 0.3 | 87.2 | 89.8 |
| 100 | 89.3 | 88.3 | 84.9 | 88.4 | 89.6 | 89.8 | 90.0 | 90.1 | 0.2 | 88.9 | 89.6 |

**Table 3.** Experimental results on large datasets

| | accuracy (%) | | | | | training time (seconds) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $C$ | CVM | RC$_l$ | RC$_u$ | RC$_{avg}$ | RMSVM | CVM | RC$_l$ | RC$_u$ | RC$_{avg}$ | RMSVM |
| **web** | | | | | | | | | | |
| 10 | 98.6 | 98.7 | 98.8 | 98.7 | 98.9 | 423 | 218 | 79 | 295 | 134 |
| 100 | 98.9 | 99.0 | 98.9 | 99.1 | | 39 | 29 | 23 | 53 | |
| 1,000 | 96.2 | 97.1 | 97.3 | 98.2 | | 20.7 | 11.6 | 12.9 | 24.6 | |
| 10,000 | 95.9 | 96.1 | 93.0 | 97.4 | | 8.88 | 7.75 | 4.53 | 12.22 | |
| **extended-usps** | | | | | | | | | | |
| 100 | 99.46 | 99.50 | 99.49 | 99.49 | - | 166 | 121 | 109 | 229 | > 1 day |
| 1,000 | 99.45 | 99.47 | 99.46 | 99.47 | | 81 | 92 | 108 | 201 | |
| **intrusion** | | | | | | | | | | |
| 100 | 91.8 | 92.0 | 92.1 | 92.1 | - | 209 | 36 | 13 | 49 | not enough memory |
| 10,000 | 91.8 | 91.7 | 91.8 | 92.1 | | 3.61 | 3.45 | 3.58 | 6.77 | |
| 1,000,000 | 91.8 | 92.0 | 88.3 | 92.3 | | 2.41 | 1.65 | 1.61 | 2.84 | |
| **SensIT Vehicle: acoustic** | | | | | | | | | | |
| 100 | 50.0 | 26.7 | 31.9 | 41.9 | 66.1 | 3923 | 706 | 138 | 832 | 79434 |
| 1,000 | 39.4 | 37.1 | 52.4 | 58.4 | | 68.5 | 23.6 | 15.1 | 38.2 | |
| 10,000 | 40.9 | 42.1 | 41.0 | 46.6 | | 9.47 | 7.46 | 6.98 | 14.1 | |
| 1,000,000 | 40.6 | 50.6 | 42.4 | 47.7 | | 4.90 | 5.45 | 4.37 | 9.64 | |
| **SensIT Vehicle: seismic** | | | | | | | | | | |
| 100 | 50.1 | 23.2 | 23.5 | 26.2 | 64.6 | 3882 | 796 | 164 | 951 | 35194 |
| 1,000 | 28.1 | 46.5 | 59.1 | 57.3 | | 55.0 | 21.5 | 11.6 | 33.3 | |
| 10,000 | 51.2 | 41.6 | 50.3 | 55.0 | | 8.50 | 7.17 | 6.61 | 13.7 | |
| 1,000,000 | 34.6 | 46.7 | 41.1 | 45.9 | | 4.60 | 4.25 | 4.67 | 8.70 | |

on the extremely noisy `SensIT Vehicle` in the bottom of *Table 3*. Perhaps the RMCVM could always choose the right core vector and have less iteration before convergence as a result. When $C$ is too small the RMCVM is inferior to the CVM since it is underfitting.

## 6 Conclusion

Motivated by the rough margin, we propose the rough margin based core vector machine and demonstrate that it can be solved efficiently. Experimental results show that the derived algorithms can handle very large datasets and the accuracy is almost comparable to the rough margin based SVM.

## References


[1] Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
[2] Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods, pp. 185–208. MIT Press, Cambridge (1999)
[3] Smola, A.J., Schökopf, B.: Sparse greedy matrix approximation for machine learning. In: 17th ICML, pp. 911–918 (2000)
[4] Tsang, I.W., Kwok, J.T., Cheung, P.M.: Very large svm training using core vector machines. In: 20th AISTATS (2005)
[5] Tsang, I.W., Kwok, J.T., Cheung, P.M.: Core vector machines: Fast svm training on very large data sets. JMLR 6, 363–392 (2005)
[6] Tsang, I.W., Kocsor, A., Kwok, J.T.: Simpler core vector machines with enclosing balls. In: 24th ICML, pp. 911–918 (2007)
[7] Asharaf, S., Murty, M.N., Shevade, S.K.: Multiclass core vector machine. In: 24th ICML, pp. 41–48 (2007)
[8] Zhang, J., Wang, Y.: A rough margin based support vector machine. Information Sciences 178, 2204–2214 (2008)
[9] Bădoiu, M., Clarkson, K.L.: Optimal core-sets for balls. In: DIMACS Workshop on Computational Geometry (2002)
[10] Tsang, I.W., Kwok, J.T., Lai, K.T.: Core vector regression for very large regression problems. In: 22nd ICML, pp. 912–919 (2005)
[11] Tsang, I.W., Kwok, J.T., Zurada, J.M.: Generalized core vector machines. IEEE Transactions on Neural Networks 17, 1126–1140 (2006)
[12] Pawlak, Z.: Rough sets. International Journal of Computer and Information Sciences 11, 341–356 (1982)
[13] Weston, J., Collobert, R., Sinz, F., Bottou, L., Vapnik, V.: Inference with the universum. In: 23rd ICML, pp. 1009–1016 (2006)
[14] Crisp, D.J., Burges, C.J.C.: A geometric interpretation of $\nu$-svm classifiers. In: NIPS, vol. 12, pp. 244–250 (1999)
[15] Chang, C.C., Lin, C.J.: Training $\nu$-support vector classifiers: Theory and algorithms. Neural Computation 13, 2119–2147 (2001)
[16] Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20, 273–297 (1995)
[17] Schölkopf, B., Smola, A.J., Williamson, R.C., Bartlett, P.L.: New support vector algorithms. Neural Computation 12, 1207–1245 (2000)
[18] Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), `http://www.csie.ntu.edu.tw/~cjlin/libsvm`