Nonparallel Support Vector Machines for Pattern Classification

Yingjie Tian, Zhiquan Qi, XuChan Ju, Yong Shi, Xiaohui Liu

Abstract—We propose a novel nonparallel classifier, named nonparallel support vector machine (NPSVM), for binary classification. Totally different with the existing nonparallel classifiers, such as the generalized eigenvalue proximal support vector machine (GEPSVM) and the twin support vector machine (TWSVM), our NPSVM has several incomparable advantages: (1) Two primal problems are constructed implementing the structural risk minimization principle; (2) The dual problems of these two primal problems have the same advantages as that of the standard SVMs, so that the kernel trick can be applied directly, while existing TWSVMs have to construct another two primal problems for nonlinear cases based on the approximate kernel-generated surfaces, furthermore, their nonlinear problems can not degenerate to the linear case even the linear kernel is used; (3) The dual problems have the same elegant formulation with that of standard SVMs and can certainly be solved efficiently by sequential minimization optimization (SMO) algorithm, while existing GEPSVM or TWSVMs are not suitable for large scale problems; (4) It has the inherent sparseness as standard SVMs; (5) Existing TWSVMs are only the special cases of the NPSVM when the parameters of which are appropriately chosen. Experimental results on lots of data sets show the effectiveness of our method in both sparseness and classification accuracy, and therefore confirm the above conclusion further. In some sense, our NPSVM is a new starting point of nonparallel classifiers.

Index Terms—Support vector machines, nonparallel, structural risk minimization principle, sparseness, classification.

1 Introduction

SUPPORT vector machines (SVMs), which were introduced by Vapnik and his co-workers in the early 1990's[1], [2], [3], are computationally powerful tools for pattern classification and regression and have already successfully applied in a wide variety of fields[4], [5], [6], [7], [8]. There are three essential elements making SVMs so successful: the principle of maximum margin, dual theory, and kernel trick. For the standard support vector classification (SVC), maximizing the margin between two parallel hyperplanes leads to solving a convex quadratic programming problem (QPP), dual theory makes introducing the kernel function possible, then the kernel trick is applied to solve nonlinear cases.

In recent years, some nonparallel hyperplane classifiers, which are different with standard SVC searching for two parallel support hyperplanes, have been proposed[9], [10]. For the twin support vector machine (TWSVM), it seeks two nonparallel proximal hyperplanes such that each hyperplane is closer to one of the two classes and is at least one distance from the other. This strategy results that TWSVM solves two smaller QPPs, whereas SVC solves one larger QPP, which increases the TWSVM training speed by approximately fourfold compared to that of SVC. TWSVMs have been studied extensively[11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25].

However, there are still several drawbacks in existing TWSVMs:

- ♦ Unlike the standard SVMs employing soft-margin loss function for classification and ε-insensitive loss function for regression, TWSVMs lost the sparseness by using two loss functions for each class: a quadratic loss function making the proximal hyperplane close enough to the class itself, and a soft-margin loss function making the hyperplane as far as possible from the other class, which results that almost all the points in this class and some points in the other class contribute to each final decision function. In this paper, we called this phenomenon **Semi-Sparseness**.
- ♦ For the nonlinear case, TWSVMs consider the kernel-generated surfaces instead of hyperplanes and construct extra two different primal problems, which means that they have to solve two problems for linear case and two other problems for nonlinear case separately. However, in the standard SVMs, only one dual problem is solved for both cases with different kernels.
- ♦ Although TWSVMs only solve two smaller QPPs, they have to compute the inverse of matrices, it is in practice intractable or even impossible for a large data set by the classical methods, while in the standard SVMs, large scale problems can be solved efficiently by the well-known SMO algorithm[26].
- ♦ Only the empirical risk is considered in the primal

- *Y. Tian, Z. Qi, X. Ju and Y. Shi are with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China (E-mail: tyj@gucas.ac.cn)*
- *Z. Qi is the corresponding author (E-mail: qizhiquan@gucas.ac.cn)*
- *X. Liu is with the School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge, Middlesex, UK.*

problems of TWSVMs, and it is well known that one significant advantage of SVMs is the implementation of the structural risk minimization (SRM) principle. Although Shao et al.[15] improved TWSVM by introducing a regularization term to make the SRM principle implemented, they explained it a bit far-fetched, especially for the nonlinear case.

In this paper, we propose a novel nonparallel SVM, termed as NPSVM for binary classification. NPSVM has the incomparable advantages that (1) the semi-sparseness is promoted to the whole sparseness; (2) The regularization term is added naturally due to the introduction of ε-insensitive loss function, and two primal problems are constructed implementing the SRM principle; (3) The dual problems of these two primal problems have the same advantages as that of the standard SVMs, i.e., only the inner products appear so that the kernel trick can be applied directly; (4) The dual problems have the same formulation with that of standard SVMs and can certainly be solved efficiently by SMO, we do not need to compute the inverses of the large matrices as TWSVMs usually do; (5) The initial TWSVM or improved TBSVM are the special cases of our models. Our NPSVM degenerates to the initial TWSVM or TBSVM when the parameters of which are appropriately chosen, therefore our models are certainly superior to them theoretically.

The paper is organized as follows. Section 2 briefly dwells on the standard C-SVC and TWSVMs. Section 3 proposes our NPSVM. Section 4 deals with experimental results and Section 5 contains concluding remarks.

2 Background

In this section, we briefly introduce the C-SVC and two variations of TWSVM.

2.1 C-SVC

Consider the binary classification problem with the training set

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\} \tag{1}$$

where $x_i \in \mathcal{R}^n, y_i \in \mathcal{Y} = \{1, -1\}, i = 1, \cdots, l$, standard C-SVC formulates the problem as a convex QPP

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i, \\ \text{s.t.} \quad & y_i((w \cdot x_i) + b) \geqslant 1 - \xi_i \ , \ i = 1, \cdots, l, \\ & \xi_i \geqslant 0 \ , \ i = 1, \cdots, l, \end{aligned} \tag{2}$$

where $\xi = (\xi_1, \cdots, \xi_l)^\top$, and $C > 0$ is a penalty parameter. For this primal problem, C-SVC solves its Lagrangian dual problem

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{l}\alpha_i \ , \\ \text{s.t.} \quad & \sum_{i=1}^{l} y_i\alpha_i = 0, \\ & 0 \leqslant \alpha_i \leqslant C, i = 1, \cdots, l, \end{aligned} \tag{3}$$

where $K(x, x')$ is the kernel function, which is also a convex QPP and then constructs the decision function. The SRM principal is implemented in C-SVC: the confidential interval term $\|w\|^2$ and the empirical risk term $\sum_{i=1}^{l}\xi_i$ are minimized at the same time.

2.2 TWSVM

Consider the binary classification problem with the training set

$$T = \{(x_1, +1), \cdots, (x_p, +1), (x_{p+1}, -1), \cdots, (x_{p+q}, -1)\}, \tag{4}$$

where $x_i \in \mathcal{R}^n, i = 1, \cdots, p+q$. For linear classification problem, TWSVM[10] seeks two nonparallel hyperplanes

$$(w_+ \cdot x) + b_+ = 0 \text{ and } (w_- \cdot x) + b_- = 0 \tag{5}$$

by solving two smaller QPPs

$$\begin{aligned} \min_{w_+,b_+,\xi_-} \quad & \frac{1}{2}\sum_{i=1}^{p}((w_+ \cdot x_i) + b_+)^2 + d_1\sum_{j=p+1}^{p+q}\xi_j, \\ \text{s.t.} \quad & (w_+ \cdot x_j) + b_+ \leqslant -1 + \xi_j, j = p+1, ..., p+q, \\ & \xi_j \geqslant 0, j = p+1, ..., p+q, \end{aligned} \tag{6}$$

and

$$\begin{aligned} \min_{w_-,b_-,\xi_+} \quad & \frac{1}{2}\sum_{i=p+1}^{p+q}((w_- \cdot x_i) + b_-)^2 + d_2\sum_{j=1}^{p}\xi_j, \\ \text{s.t.} \quad & (w_- \cdot x_j) + b_- \geqslant 1 - \xi_j, j = 1, ..., p, \\ & \xi_j \geqslant 0, j = 1, ..., p, \end{aligned} \tag{7}$$

where $d_i, i = 1, 2$ are the penalty parameters. For nonlinear classification problem, two kernel-generated surfaces instead of hyperplanes are considered and two other primal problems are constructed.

2.3 TBSVM

An improved TWSVM, termed as TBSVM, is proposed in [15] whereas the structural risk is claimed to be minimized by adding a regularization term with the idea of maximizing some margin. For linear

classification problem, they solve the following two primal problems

$$\begin{aligned}\min_{w_+,b_+,\xi_-} \quad & \frac{1}{2}(\|w_+\|^2+b_+^2)+\frac{c_1}{2}\sum_{i=1}^{p}((w_+\cdot x_i)+b_+)^2 \\ & +c_2\sum_{j=p+1}^{p+q}\xi_j, \\ \text{s.t.} \quad & (w_+\cdot x_j)+b_+\leqslant -1+\xi_j, j=p+1,...,p+q, \\ & \xi_j\geqslant 0, j=p+1,...,p+q,\end{aligned} \tag{8}$$

and

$$\begin{aligned}\min_{w_-,b_-,\xi_+} \quad & \frac{1}{2}(\|w_-\|^2+b_-^2)+\frac{c_3}{2}\sum_{i=p+1}^{p+q}((w_-\cdot x_i)+b_-)^2 \\ & +c_4\sum_{j=1}^{p}\xi_j, \\ \text{s.t.} \quad & (w_-\cdot x_j)+b_-\geqslant 1-\xi_j, j=1,...,p, \\ & \xi_j\geqslant 0, j=1,...,p.\end{aligned} \tag{9}$$

For nonlinear classification problem, similar with [10] two kernel-generated surfaces instead of hyperplanes are considered and two other regularized primal problems are constructed.

Though TBSVM is claimed a little more rigorous and complete than TWSVM, there are still the drawbacks emphasized in the introduction.

3 NPSVM

In this section, we propose our nonparallel SVM, termed as NPSVM, which has several unexpected and incomparable advantages compared with the existing TWSVMs.

3.1 Linear NPSVM

We seek the two nonparallel hyperplanes (5) by solving two convex QPPs

$$\begin{aligned}\min_{w_+,b_+,\eta_+^{(*)},\xi_-} \quad & \frac{1}{2}\|w_+\|^2+C_1\sum_{i=1}^{p}(\eta_i+\eta_i^*)+C_2\sum_{j=p+1}^{p+q}\xi_j, \\ \text{s.t.} \quad & (w_+\cdot x_i)+b_+\leqslant\varepsilon+\eta_i,\ i=1,\cdots,p, \\ & -(w_+\cdot x_i)-b_+\leqslant\varepsilon+\eta_i^*,\ i=1,\cdots,p, \\ & (w_+\cdot x_j)+b_+\leqslant -1+\xi_j, \\ & \qquad j=p+1,\cdots,p+q, \\ & \eta_i,\eta_i^*\geqslant 0,\ i=1,\cdots,p, \\ & \xi_j\geqslant 0,\ j=p+1,\cdots,p+q,\end{aligned} \tag{10}$$

and

$$\begin{aligned}\min_{w_-,b_-,\eta_-^{(*)},\xi_+} \quad & \frac{1}{2}\|w_-\|^2+C_3\sum_{i=p+1}^{p+q}(\eta_i+\eta_i^*)+C_4\sum_{j=1}^{p}\xi_j, \\ \text{s.t.} \quad & (w_-\cdot x_i)+b_-\leqslant\varepsilon+\eta_i, \\ & \qquad i=p+1,\cdots,p+q, \\ & -(w_-\cdot x_i)-b_-\leqslant\varepsilon+\eta_i^*, \\ & \qquad i=p+1,\cdots,p+q, \\ & (w_-\cdot x_j)+b_-\geqslant 1-\xi_j,\ j=1,\cdots,p, \\ & \eta_i,\eta_i^*\geqslant 0,\ i=p+1,\cdots,p+q, \\ & \xi_j\geqslant 0,\ j=1,\cdots,p,\end{aligned} \tag{11}$$

where $x_i, i=1,\cdots,p$ are positive inputs, and $x_i, i=p+1,\cdots,p+q$ are negative inputs, $C_i\geqslant 0, i=1,\cdots,4$ are penalty parameters, $\xi_+=(\xi_1,\cdots,\xi_p)^\top$, $\xi_-=(\xi_{p+1},\cdots,\xi_{p+q})^\top$, $\eta_+^{(*)}=(\eta_+^\top,\eta_+^{*\top})^\top=(\eta_1,\cdots,\eta_p,\eta_1^*,\cdots,\eta_p^*)^\top$, $\eta_-^{(*)}=(\eta_-^\top,\eta_-^{*\top})^\top=(\eta_{p+1},\cdots,\eta_{p+q},\eta_{p+1}^*,\cdots,\eta_{p+q}^*)^\top$, are slack variables.

Now we discuss the primal problem (10) geometrically in $\mathcal{R}^2$ (see Fig.1). First, we hope that the positive class locate as much as possible in the ε-band between the hyperplanes $(w_+\cdot x)+b_+=\varepsilon$ and $(w_+\cdot x)+b_+=-\varepsilon$ (red thin solid lines), the errors $\eta_i+\eta_i^*, i=1,\cdots,p$ are measured by the ε-insensitive loss function; Second, we hope to maximize the margin between the hyperplanes $(w_+\cdot x)+b_+=\varepsilon$ and $(w_+\cdot x)+b_+=-\varepsilon$, which can be expressed by $\frac{2\varepsilon}{\|w\|}$; Third, similar with the TWSVM, we also need to push the negative class from the hyperplane $(w_+\cdot x)+b_+=-1$ (red thin dotted line) as far as possible, the errors $\xi_i, i=p+1,\cdots,p+q$ are measured by the soft margin loss function.

- Based on the above three considerations, problem (10) is established and the structural risk minimization principle is implemented naturally. Problem (11) is established similarly. When the parameter ε is set to be zero, and the penalty parameters are chosen to be $C_i=\frac{c_i}{2}, i=1,3$ and $C_i=c_i, i=2,4$, problems (10) and (11) of NPSVM degenerate to problems (8) and (9) except that the L_1-loss "$|\eta_i+\eta_i^*|$" is taken instead of the L_2-loss "$(w_\pm\cdot x_i)+b_\pm)^2$", and an additional term $\frac{1}{2}b^2$. Furthermore, if the parameter ε is set to be zero, and $C_i, i=1,\cdots,4$ are chosen large enough and satisfying $\frac{C_2}{C_1}=2d_1, \frac{C_4}{C_3}=2d_2$, problems (10) and (11) degenerate to problems (6) and (7) except that the L_1-loss is taken instead of the L_2-loss.

In order to get the solutions of problems (10) and (11), we need to derive their dual problems. The

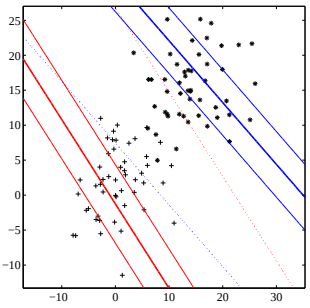


Fig. 1. Geometrical illustration of NPSVM in $\mathcal{R}^2$

Lagrangian of the problem (10) is given by

$$\begin{aligned} &L(w_+, b_+, \eta_+^{(*)}, \xi_-, \alpha_+^{(*)}, \gamma_+^{(*)}, \beta_-, \lambda_-) \\ &= \frac{1}{2}\|w_+\|^2 + C_1 \sum_{i=1}^{p}(\eta_i + \eta_i^*) + C_2 \sum_{j=p+1}^{p+q} \xi_i \\ &\quad + \sum_{i=1}^{p} \alpha_i((w_+ \cdot x_i) + b_+ - \eta_i - \varepsilon) \\ &\quad + \sum_{i=1}^{p} \alpha_i^*(-(w_+ \cdot x_i) - b_+ - \eta_i^* - \varepsilon) \\ &\quad + \sum_{j=p+1}^{p+q} \beta_j((w_+ \cdot x_j) + b_+ + 1 - \xi_j) \\ &\quad - \sum_{i=1}^{p} \gamma_i \eta_i - \sum_{i=1}^{p} \gamma_i^* \eta_i^* - \sum_{j=p+1}^{p+q} \lambda_j \xi_j, \end{aligned} \tag{12}$$

where $\alpha_+^{(*)} = (\alpha_+^\top, \alpha_+^{*\top})^\top = (\alpha_1, \cdots, \alpha_p, \alpha_1^*, \cdots, \alpha_p^*)^\top$, $\gamma_+^{(*)} = (\gamma_+^\top, \gamma_+^{*\top})^\top = (\gamma_1, \cdots, \gamma_p, \gamma_1^*, \cdots, \gamma_p^*)^\top$, $\beta_- = (\beta_{p+1}, \cdots, \beta_{p+q})^\top$, $\lambda_- = (\lambda_{p+1}, \cdots, \beta_{p+q})^\top$ are the Lagrange multiplier vectors. The Karush-Kuhn-Tucker (KKT) conditions[27] for $w_+, b_+, \eta_+^{(*)}, \xi_-$ and $\alpha_+^{(*)}, \gamma_+^{(*)}, \beta_-, \lambda_-$ are given by

$$\nabla_{w_+} L = w_+ + \sum_{i=1}^{p} \alpha_i x_i - \sum_{i=1}^{p} \alpha_i^* x_i + \sum_{j=p+1}^{p+q} \beta_j x_j = 0, \tag{13}$$

$$\nabla_{b_+} L = \sum_{i=1}^{p} \alpha_i - \sum_{i=1}^{p} \alpha_i^* + \sum_{j=p+1}^{p+q} \beta_j = 0, \tag{14}$$

$$\nabla_{\eta_+} L = C_1 e_+ - \alpha_+ - \gamma_+ = 0, \tag{15}$$

$$\nabla_{\eta_+^*} L = C_1 e_+ - \alpha_+^* - \gamma_+^* = 0, \tag{16}$$

$$\nabla_{\xi_-} L = C_2 e_- - \beta_- - \lambda_- = 0, \tag{17}$$

$$(w_+ \cdot x_i) + b_+ \leqslant \varepsilon + \eta_i, \ i = 1, \cdots, p, \tag{18}$$

$$-(w_+ \cdot x_i) - b_+ \leqslant \varepsilon + \eta_i^*, \ i = 1, \cdots, p, \tag{19}$$

$$(w_+ \cdot x_j) + b_+ \leqslant -1 + \xi_j, \ j = p+1, \cdots, p+q, \tag{20}$$

$$\eta_i, \eta_i^* \geqslant 0, \ i = 1, \cdots, p, \tag{21}$$

$$\xi_j \geqslant 0, \ j = p+1, \cdots, p+q, \tag{22}$$

where $e_+ = (1, \cdots, 1)^\top \in \mathcal{R}^p$, $e_- = (1, \cdots, 1)^\top \in \mathcal{R}^q$. Since $\gamma_+, \gamma_+^* \geqslant 0, \lambda_- \geqslant 0$, from (15), (16) and (17) we have

$$0 \leqslant \alpha_+, \alpha_+^* \leqslant C_1 e_+, \tag{23}$$

$$0 \leqslant \beta_- \leqslant C_2 e_-. \tag{24}$$

And from (13), we have

$$w_+ = \sum_{i=1}^{p} (\alpha_i^* - \alpha_i) x_i - \sum_{j=p+1}^{p+q} \beta_j x_j. \tag{25}$$

Then putting (25) into the Lagrangian (12) and using (13)$\sim$(22), we obtain the dual problem of the problem (10)

$$\begin{aligned} \min_{\alpha_+^{(*)}, , \beta_-} \ & \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(x_i \cdot x_j) \\ & - \sum_{i=1}^{p} \sum_{j=p+1}^{p+q} (\alpha_i^* - \alpha_i)\beta_j (x_i \cdot x_j) \\ & + \frac{1}{2} \sum_{i=p+1}^{p+q} \sum_{j=p+1}^{p+q} \beta_i \beta_j (x_i \cdot x_j) \\ & + \varepsilon \sum_{i=1}^{p} (\alpha_i^* + \alpha_i) - \sum_{i=p+1}^{p+q} \beta_i, \\ \text{s.t.} \ & \sum_{i=1}^{p} (\alpha_i - \alpha_i^*) + \sum_{j=p+1}^{p+q} \beta_j = 0, \\ & 0 \leqslant \alpha_+, \alpha_+^* \leqslant C_1 e_+, \\ & 0 \leqslant \beta_- \leqslant C_2 e_-. \end{aligned} \tag{26}$$

Concisely, this problem can be further formulated as

$$\begin{aligned} \min_{\alpha_+^{(*)}, \beta_-} \ & \frac{1}{2}(\alpha_+^* - \alpha_+)^\top A A^\top (\alpha_+^* - \alpha_+) \\ & - (\alpha_+^* - \alpha_+)^\top A B^\top \beta_- + \frac{1}{2} \beta_-^\top B B^\top \beta_- \\ & + \varepsilon e_+^\top (\alpha^* + \alpha) - e_-^\top \beta_-, \\ \text{s.t.} \ & e_+^\top (\alpha_+ - \alpha_+^*) + e_-^\top \beta_- = 0, \\ & 0 \leqslant \alpha_+, \alpha_+^* \leqslant C_1 e_+, \\ & 0 \leqslant \beta_- \leqslant C_2 e_-, \end{aligned} \tag{27}$$

where $A = (x_1, \cdots, x_p)^\top \in \mathcal{R}^{p \times n}$, $B = (x_{p+1}, \cdots, x_{p+q}) \in \mathcal{R}^{q \times n}$. Furthermore, let

$$\tilde{\pi} = (\alpha_+^{*\top}, \alpha_+^\top, \beta_-^\top)^\top, \tag{28}$$

$$\tilde{\kappa} = (\varepsilon e_+^\top, \varepsilon e_+^\top, -e_-^\top)^\top, \tag{29}$$

$$\tilde{e} = (-e_+^\top, e_+^\top, e_-^\top)^\top, \tag{30}$$

$$\tilde{C} = (C_1 e_+^\top, C_1 e_+^\top, C_2 e_-^\top)^\top \tag{31}$$

and

$$\tilde{\Lambda} = \begin{pmatrix} H_1 & -H_2 \\ -H_2^\top & H_3 \end{pmatrix}, H_1 = \begin{pmatrix} AA^\top & -AA^\top \\ -AA^\top & AA^\top \end{pmatrix}, H_2 = \begin{pmatrix} AB^\top \\ -AB^\top \end{pmatrix}, H_3 = BB^\top, \tag{32}$$

then problem (27) is reformulated as

$$\begin{aligned} \min_{\tilde{\pi}} \quad & \frac{1}{2}\tilde{\pi}^\top \tilde{\Lambda}\tilde{\pi} + \tilde{\kappa}^\top \tilde{\pi}, \\ \text{s.t.} \quad & \tilde{e}^\top \tilde{\pi} = 0, \\ & 0 \leqslant \tilde{\pi} \leqslant \tilde{C}. \end{aligned} \tag{33}$$

- Obviously, problem (33) is a convex QPP and exactly the same elegant formulation as problem (3), the well known SMO can be applied directly with a minor modification.

For the problem (33), applying the KKT conditions we can get the following conclusions without proof which is similar with the conclusions in [3], [28].

Theorem 3.1 *Suppose that* $\tilde{\pi} = (\alpha_+^{*\top}, \alpha_+^\top, \beta_-^\top)^\top$ *is a solution of the problem* (33)*, then for* $i = 1, \cdots, p$*, each pair of* α_i *and* α_i^* *can not be both simultaneously nonzero, i.e.,* $\alpha_i \alpha_i^* = 0, i = 1, \cdots, p$.

Theorem 3.2 *Suppose that* $\tilde{\pi} = (\alpha_+^{*\top}, \alpha_+^\top, \beta_-^\top)^\top$ *is a solution of the problem* (33)*, if there exist components of* $\tilde{\pi}$ *of which value is in the interval* $(0, \tilde{C})$*, then the solution* (w_+, b_+) *of the problem* (10) *can be obtained in the following way:*
Let

$$w_+ = \sum_{i=1}^{p} (\alpha_i^* - \alpha_i) x_i - \sum_{j=p+1}^{p+q} \beta_j x_j, \tag{34}$$

and choose a component of α_+*,* $\alpha_{+j} \in (0, C_1)$*, compute*

$$b_+ = -(w_+ \cdot x_j) + \varepsilon, \tag{35}$$

or choose a component of α_+^**,* $\alpha_{+k}^* \in (0, C_1)$*, compute*

$$b_+ = -(w_+ \cdot x_k) - \varepsilon, \tag{36}$$

or choose a component of β_-*,* $\beta_{-m} \in (0, C_2)$*, compute*

$$b_+ = -(w_+ \cdot x_m) - 1. \tag{37}$$

In the same way, the dual of the problem (11) is obtained

$$\begin{aligned} \min_{\alpha_-^{(*)}, \beta_+} \quad & \frac{1}{2} \sum_{i=p+1}^{p+q} \sum_{j=p+1}^{p+q} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(x_i \cdot x_j) \\ & + \sum_{i=p+1}^{p+q} \sum_{j=1}^{p} (\alpha_i^* - \alpha_i)\beta_j (x_i \cdot x_j) \\ & + \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} \beta_i \beta_j (x_i \cdot x_j) \\ & + \varepsilon \sum_{i=p+1}^{p+q} (\alpha_i^* + \alpha_i) - \sum_{i=1}^{p} \beta_i, \\ \text{s.t.} \quad & \sum_{i=p+1}^{p+q} (\alpha_i - \alpha_i^*) - \sum_{j=1}^{p} \beta_j = 0, \\ & 0 \leqslant \alpha_i, \alpha_i^* \leqslant C_3, i = p+1, \cdots, p+q, \\ & 0 \leqslant \beta_i \leqslant C_4, i = 1, \cdots, p, \end{aligned} \tag{38}$$

where $\alpha_-^{(*)}, \beta_+$ are the Lagrange multiplier vectors. It can also be rewritten as

$$\begin{aligned} \min_{\alpha_-^{(*)}, \beta_+} \quad & \frac{1}{2}(\alpha_-^* - \alpha_-)^\top BB^\top (\alpha_-^* - \alpha_-) \\ & + (\alpha_-^* - \alpha_-)^\top BA^\top \beta_+ + \frac{1}{2}\beta_+^\top AA^\top \beta_+ \\ & + \varepsilon e_-^\top (\alpha^* + \alpha) - e_+^\top \beta_+, \\ \text{s.t.} \quad & e_-^\top (\alpha_- - \alpha_-^*) - e_+^\top \beta_+ = 0, \\ & 0 \leqslant \alpha_-, \alpha_-^* \leqslant C_3 e_-, \\ & 0 \leqslant \beta_+ \leqslant C_4 e_+. \end{aligned} \tag{39}$$

Concisely, it is reformulated as

$$\begin{aligned} \min_{\hat{\pi}} \quad & \frac{1}{2}\hat{\pi}^\top \hat{\Lambda}\hat{\pi} + \hat{\kappa}^\top \hat{\pi}, \\ \text{s.t.} \quad & \hat{e}^\top \hat{\pi} = 0, \\ & 0 \leqslant \hat{\pi} \leqslant \hat{C}, \end{aligned} \tag{40}$$

where

$$\hat{\pi} = (\alpha_-^{*\top}, \alpha_-^\top, \beta_+^\top)^\top, \tag{41}$$

$$\tilde{\kappa} = (\varepsilon e_-^\top, \varepsilon e_-^\top, -e_+^\top)^\top, \tag{42}$$

$$\hat{e} = (-e_-^\top, e_-^\top, -e_+^\top)^\top, \tag{43}$$

$$\hat{C} = (C_3 e_-^\top, C_3 e_-^\top, C_4 e_+^\top)^\top \tag{44}$$

and

$$\hat{\Lambda} = \begin{pmatrix} Q_1 & Q_2 \\ Q_2^\top & Q_3 \end{pmatrix}, Q_1 = \begin{pmatrix} BB^\top & -BB^\top \\ -BB^\top & BB^\top \end{pmatrix}, Q_2 = \begin{pmatrix} BA^\top \\ -BA^\top \end{pmatrix}, Q_3 = AA^\top, \tag{45}$$

For the problem (40), we have the following conclusions corresponding to problem (33).

Theorem 3.3 *Suppose that* $\hat{\pi} = (\alpha_-^{*\top}, \alpha_-^\top, \beta_+^\top)^\top$ *is a solution of the problem* (40)*, then for* $i = p+1, \cdots, p+q$*, each pair of* α_i *and* α_i^* *can not be both simultaneously nonzero, i.e.,* $\alpha_i \alpha_i^* = 0, i = p+1, \cdots, p+q$.

Theorem 3.4 *Suppose that* $\hat{\pi} = (\alpha_-^{*\top}, \alpha_-^\top, \beta_+^\top)^\top$ *is a solution of the problem* (40)*, if there exist components of* $\hat{\pi}$ *of which value is in the interval* $(0, \hat{C})$*, then the solution* (w_-, b_-) *of the problem* (11) *can be obtained in the following way:*
Let

$$w_- = \sum_{i=p+1}^{p+q} (\alpha_i^* - \alpha_i) x_i + \sum_{j=1}^{p} \beta_j x_j, \tag{46}$$

and choose a component of α_+*,* $\alpha_{+j} \in (0, C_3)$*, compute*

$$b_- = -(w_- \cdot x_j) + \varepsilon, \tag{47}$$

or choose a component of α_+^**,* $\alpha_{+k}^* \in (0, C_3)$*, compute*

$$b_- = -(w_- \cdot x_k) - \varepsilon, \tag{48}$$

or choose a component of β_-*,* $\beta_{-m} \in (0, C_4)$*, compute*

$$b_- = -(w_- \cdot x_m) + 1. \tag{49}$$

- From Theorems 3.2 and 3.4, we can see that the inherent semi-sparseness in the existing TWSVMs is improved to the whole sparseness in our linear NPSVM, because of the introduction of ε-insensitive loss function instead of the quadratic loss function for each class itself.

Once the solutions (w_+, b_+) and (w_-, b_-) of the problems (10) and (11) are obtained, a new point $x \in \mathcal{R}^n$ is predicted to the *Class* by

$$Class = \arg \min_{k=-,+} |(w_k \cdot x) + b_k|, \tag{50}$$

where $|\cdot|$ is the perpendicular distance of point x from the planes $(w_k \cdot x) + b_k = 0, k = -, +$.

3.2 Nonlinear NPSVM

Now we extend the linear NPSVM to the nonlinear case.

- Totally different with all the existing TWSVMs, we do not need consider the extra kernel-generated surfaces since only inner products appear in the dual problems (27) and (39), so the kernel functions are applied directly in the problems and the linear NPSVM is easily extended to the nonlinear classifiers.

In detail, introducing the kernel function $K(x, x') = (\Phi(x) \cdot \Phi(x'))$ and the corresponding transformation

$$\mathrm{x} = \Phi(x), \tag{51}$$

where $\mathrm{x} \in \mathcal{H}$, $\mathcal{H}$ is the Hilbert space, we can construct the corresponding problems (10) and (11) in $\mathcal{H}$, the only difference is that the weight vectors w_+ and w_- in $\mathcal{R}^n$ change to be w_+ and w_- respectively. Two dual problems to be solved are

$$\begin{aligned}
\min_{\alpha_+^{(*)}, \beta_-} \quad & \frac{1}{2}(\alpha_+^* - \alpha_+)^\top K(A, A^\top)(\alpha_+^* - \alpha_+) \\
& - (\alpha_+^* - \alpha_+)^\top K(A, B^\top)\beta_- + \frac{1}{2}\beta_-^\top K(B, B^\top)\beta_- \\
& + \varepsilon e_+^\top(\alpha^* + \alpha) - e_-^\top \beta_-, \\
\text{s.t.} \quad & e_+^\top(\alpha_+ - \alpha_+^*) + e_-^\top \beta_- = 0, \\
& 0 \leqslant \alpha_+, \alpha_+^* \leqslant C_1 e_+, \\
& 0 \leqslant \beta_- \leqslant C_2 e_-,
\end{aligned} \tag{52}$$

and

$$\begin{aligned}
\min_{\alpha_-^{(*)}, \beta_+} \quad & \frac{1}{2}(\alpha_-^* - \alpha_-)^\top K(B, B^\top)^\top(\alpha_-^* - \alpha_-) \\
& + (\alpha_-^* - \alpha_-)^\top K(B, A^\top)\beta_+ + \frac{1}{2}\beta_+^\top K(A, A^\top)\beta_+ \\
& + \varepsilon e_-^\top(\alpha^* + \alpha) - e_+^\top \beta_+, \\
\text{s.t.} \quad & e_-^\top(\alpha_- - \alpha_-^*) - e_+^\top \beta_+ = 0, \\
& 0 \leqslant \alpha_-, \alpha_-^* \leqslant C_3 e_-, \\
& 0 \leqslant \beta_+ \leqslant C_4 e_+,
\end{aligned} \tag{53}$$

respectively.

Corresponding Theorems are similar with Theorems 3.1∼3.4 and we only need to take $K(x, x')$ instead of $(x \cdot x')$.

Now we establish the NPSVM as follows:

Algorithm 3.5 (NPSVM)

(1) Input the training set (8);

(2) Choose appropriate kernels $K(x, x')$, appropriate parameters $\varepsilon > 0$, C_1, C_2 for problem (27) , and $C_3, C_4 > 0$ for problem (39);

(3) Construct and solve the two convex QPPs (52) and (53) separately, get the solutions $\alpha^{(*)} = (\alpha_1, \cdots, \alpha_{p+q}, \alpha_1^*, \cdots, \alpha_{p+q}^*)^\top$ and $\beta = (\beta_1, \cdots, \beta_{p+q})^\top$;

(4) Construct the decision functions

$$f_+(x) = \sum_{i=1}^{p}(\alpha_i^* - \alpha_i)K(x_i, x) - \sum_{j=p+1}^{p+q} \beta_j K(x_j, x) + b_+, \tag{54}$$

and

$$f_-(x) = \sum_{i=p+1}^{p+q}(\alpha_i^* - \alpha_i)K(x_i, x) + \sum_{j=1}^{p} \beta_j K(x_j, x) + b_-, \tag{55}$$

separately, where b_-, b_+ are computed by Theorems 3.2 and 3.4 for the kernel cases;

(5) For any new input x, assign it to the class $k(k = -, +)$ by

$$\arg \min_{k=-,+} \frac{|f_k(x)|}{\| \triangle_k \|}, \tag{56}$$

where

$$\triangle_+ = \tilde{\pi}^\top \tilde{\Lambda} \tilde{\pi}, \ \triangle_- = \hat{\pi}^\top \hat{\Lambda} \hat{\pi}. \tag{57}$$

3.3 Advantages of NPSVM

As NPSVM degenerates to TBSVM and TWSVM when parameters are chosen appropriately (See the discussion in Section 3.1), it is theoretically superior to them. Furthermore, it is more flexible and has better generalization ability than typical SVMs since it pursues two nonparallel surfaces for discrimination. Though NPSVM has an additional parameter ε which leads to two larger optimal problems than TBSVM (about 3 times), it still has the following advantages.

- Although TWSVM and TBSVM solve smaller QPPs in which successive overrelaxation (SOR) technique or coordinate descent method can be applied[15], [18], they have to compute the inverse matrices before training which is in practice intractable or even impossible for a large data set. More detailed, suppose the size of the training set is l, and the size of negative training set is roughly equal to the size of positive set, i.e. $p \approx q \approx 0.5l$, the computational complexity of TWSVM or TBSVM solved by SOR is estimated as

$$O(l^3) + \sharp iteration \times O(0.5l), \tag{58}$$

where $O(l^3)$ is the complexity of computing $l \times l$ inverse matrix, and $\sharp iteration \times O(0.5l)$ is of SOR for $0.5l$ sized problem($\sharp iteration$ is the number of the iterations, experiments in [29] has shown that $\sharp iteration$ is almost linear scaling with the size l). While NPSVM dose not require the inverse matrices and can be solved efficiently by the SMO-type technique, [30] has proved that for the two convex QPPs (52) and (53), an SMO-type decomposition method [31] implemented in LIBSVM has the complexity

$$\sharp iterations \times O(1.5l) \tag{59}$$

if most columns of the kernel matrix are cached throughout iterations ([30] also pointed out that there is no theoretical result yet on LIBSVM's number of iterations. Empirically, it is known that the number of iterations may be higher than linear to the number of training data). Comparing equations (58) and (59), obviously NPSVM is faster than TWSVMs.

- Though TBSVM improved TWSVM by introducing the regularization terms $(\|w_+\|^2 + b_+^2)$ (for example in problem (8), another regularization term, $\|w_+\|^2$, can be found in [18] and [20]) to make the SRM principle implemented, it can only be explained for the linear case that $\frac{1}{\sqrt{\|w_+\|^2 + b_+^2}}$ is the margin of two parallel hyperplanes $(w_+ \cdot x) + b_+ = 0$ (the proximal hyperplane) and $(w_+ \cdot x) + b_+ = -1$ (the bounding hyperplane) in $\mathcal{R}^{n+1}$ space . However, for the nonlinear case, it is not a "real" kernel method like the standard SVMs usually do, it considers the kernel-generated surfaces, and apply the regularization terms for example $(\|u_+\|^2 + b_+^2)$ [15]. This term can not be explained clearly, since it is only an approximation of the term $(\|\mathrm{w}_+\|^2 + b_+^2)$ in Hilbert space. NPSVM introduces the regularization terms $\|w_+\|^2$ (for example in (10)) for linear case and $\|\mathrm{w}_\pm\|^2$ for nonlinear case naturally and reasonably, since $\frac{2}{\|w_\pm\|}$ is the margin of two parallel hyperplanes $(w_\pm \cdot x) + b_\pm = \varepsilon$ and $(w_\pm \cdot x) + b_\pm = -\varepsilon$ in $\mathcal{R}^n$ space, while $\frac{2}{\|\mathrm{w}_\pm\|}$ is the margin of two parallel hyperplanes $(\mathrm{w}_\pm \cdot \mathrm{x}) + b_\pm = \varepsilon$ and $(\mathrm{w}_\pm \cdot \mathrm{x}) + b_\pm = -\varepsilon$ in Hilbert space.
- For the nonlinear case, TWSVMs have to consider the kernel-generated surfaces instead of the hyperplanes in the Hilbert space, they are still parametric methods. NPSVM constructs two primal problems for both cases via using different kernels, which is the marrow of the standard SVMs.

4 Experimental Results

In this section, in order to validate the performance of our NPSVM, we compare it with C-SVC, TWSVM, TBSVM on different types of datasets. All methods are implemented in MATLAB 2010[32] on a PC with an Intel Core I5 processor and 2 GB RAM. TBSVM and TWSVM are solved by the optimization toolbox, C-SVC are solved by the SMO algorithm, and NPSVM are solved by a modified SMO technique.

4.1 Illustrated Iris Dataset

First, we apply NPSVM to the iris data set[33], which is an established data set used for demonstrating the performance of classification algorithms. It contains three classes (Setosa, Versilcolor, Viginica) and four attributes for an iris, and the goal is to classify the class of iris based on these four attributes. Here we restrict ourselves to the two classes (Versilcolor, Viginica), and the two features that contain the most information about the class, namely the petal length and the petal width. The distribution of the data is illustrated in Fig.2, where "+"s and "∗"s represent classes Versilcolor and Viginica respectively.

Linear and RBF kernel $K(x, x') = \exp(\frac{-\|x-x'\|^2}{\sigma})$ are used in which the parameter σ is fixed to be 4.0, and set $C = 10$, ε varies in $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$. Experiment results are shown in Fig.2, where two proximal lines $f_+(x) = 0$ and $f_-(x) = 0$, four ε-bounded lines $f_+(x) = \pm\varepsilon$ and $f_-(x) = \pm\varepsilon$, two margin lines $f_+(x) = -1$ and $f_-(x) = 1$ are depicted, support vectors are marked by "$\circ$" for different ε. Fig.3 records the varying percentage of support vectors corresponding to problems (52) and (53), respectively, we can see that with the increasing ε, the number of support vectors decreases therefore the semi-sparseness ($\varepsilon = 0$) is improved and the sparseness increases for both linear and nonlinear cases.

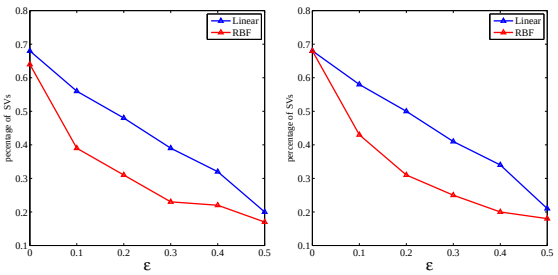
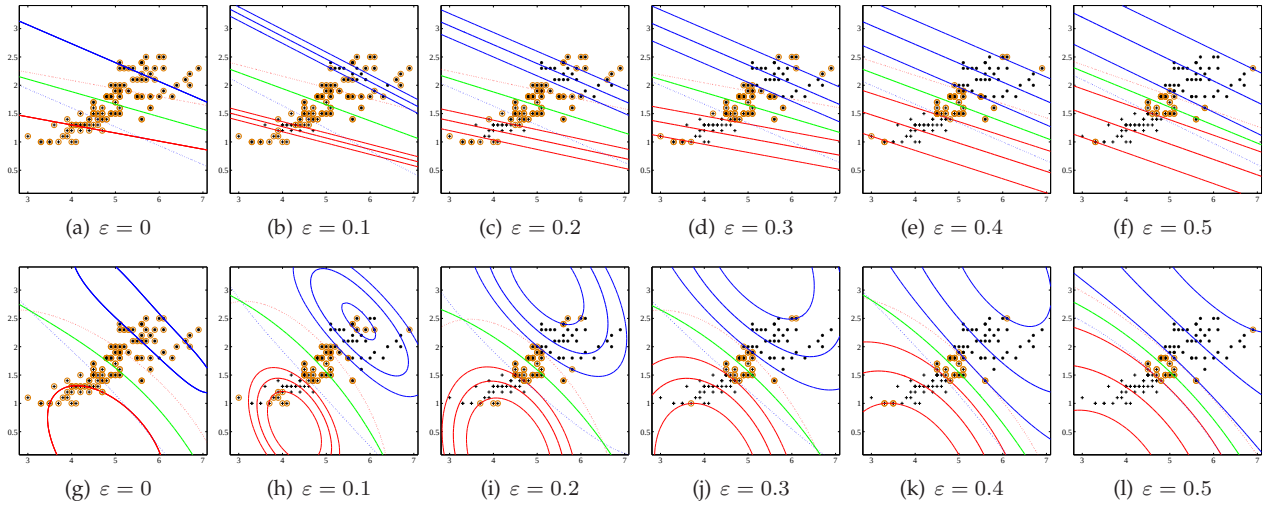


Fig. 3. Sparseness increases with the increasing ε: (a) for problem (52); (b) for problem (53).

4.2 UCI and NDC datasets

Second, we perform these methods on several publicly available benchmark datasets [33], some of which



(a) $\varepsilon = 0$ (b) $\varepsilon = 0.1$ (c) $\varepsilon = 0.2$ (d) $\varepsilon = 0.3$ (e) $\varepsilon = 0.4$ (f) $\varepsilon = 0.5$

(g) $\varepsilon = 0$ (h) $\varepsilon = 0.1$ (i) $\varepsilon = 0.2$ (j) $\varepsilon = 0.3$ (k) $\varepsilon = 0.4$ (l) $\varepsilon = 0.5$

Fig. 2. Linear cases: (a)∼(f); Nonlinear cases: (g)∼(i). Positive proximal line $f_+(x) = 0$(red thick solid line), negative proximal line $f_-(x) = 0$ (blue thick solid line), positive ε-bounded lines $f_+(x) = \pm\varepsilon$ (red thin solid lines), negative ε-bounded lines $f_-(x) = \pm\varepsilon$ (blue thin solid lines), two margin lines $f_+(x) = -1$ (red thin dotted line) and $f_-(x) = 1$ (blue thin dotted line), support vectors (marked by orange "$\circ$"), the decision boundary (green thick solid line).

are used in [10][15]. All samples were scaled such that the features locate in $[0, 1]$ before training.

For all the methods, the RBF kernel $K(x, x') = \exp(\frac{-\|x-x'\|^2}{\sigma})$ is applied, the optimal parameters $d_i, i = 1, 2$ in TWSVM, $c_i = 1, \cdots, 4$ in TBSVM, $C_i, i = 1, \cdots, 4$ in NPSVM along with σ are tuned for best classification accuracy in the range 2^{-8} to 2^{12}, the optimal parameter ε in NPSVM is obtained in the range $[0, 0.5]$ with the step 0.05.

For each dataset, we randomly select the same number of samples from different classes to compose a balanced training set, therefore based on this set to verify the above methods. This procedure is repeated 5 times and Table 1 lists the average tenfold cross-validation results of these methods in terms of accuracy and the percentage of SVs. Since the TWSVM and TBSVM are the special cases of NPSVM with some fixed parameters, theoretically NPSVM will perform better than them and in fact the results also indicate that NPSVM obtained enhanced test accuracies and sparseness when compared to them for all of the datasets. For example, for Australian, the accuracy of our NPSVM is 86.84%, and much better than 75.47% and 76.43% of TBSVM and TWSVM respectively. The reason behind this interesting phenomenon is that both TWSVM and TBSVM with kernel can not degenerate to the linear case even the linear kernel is applied. Therefore the reported best results of TWSVM in [10] is 85.80% and 85.94% in [15] for linear case, while reported 75.8% for RBF kernel in [15] and [13]. However, as we all know, RBF kernel performs approximately like linear kernel when the parameter σ is chosen large enough, they should get the similar best results with linear case after parameters tuning. While our NPSVM fixed this problem and got the best results 86.84%.

In addition, NPSVM is better than C-SVC for almost all of the datasets, and at the same time more sparse than it because of the additional sparse parameter ε, the semi-sparseness of TWSVM and TBSVM are not necessarily recorded in Table 1. Fig. 4 shows two relationships for several datasets, one relation is between the cross-validation accuracy and the parameter ε of NPSVM, the other is between the percentage of SVs and the parameter ε. These results imply NPSVM obtains a sparse classifier with good generalization.

We further compare NPSVM , TWSVM and TBSVM with the two-dimensional scatter plots that are obtained from the part test data points for the Australian, BUPA-liver, Heart-Statlog and Image. These datasets are randomly comprised of 200 points: 100 positive and 100 negative respectively. The plots are obtained by plotting points with coordinates: perpendicular distance of a test input x from hyperplane (54) and the distance from hyperlane (55). Figs. 5 describe the comparisons of the three methods on the four data sets. Obviously NPSVM obtained better clustered points and separated classes than TBSVM and TWSVM.

In order to further observe the computing time of the methods scaling w.r.t. the number of data points, we also performed experiments on large datasets, generated using David Musicant's NDC Data Generator[34]. Table 2 gives a description of NDC datasets. We used RBF kernel with $\sigma = 1$ and fixed penalty parameters of all methods: $c_1 = c_2 = 1$ in TWSVM and TBSVM, $C_i = 1, i = 1, \cdots, 4$ in NPSVM.

Table 3 shows the comparison results in terms of training time and accuracy for the NPSVM, TWSVM,

TABLE 1
Average results of the benchmark datasets

Datasets	TWSVM Accuracy % SVs %	TBSVM Accuracy % SVs %	C-SVC Accuracy % SVs %	NPSVM Accuracy % SVs %
Australian (383+307) × 14	75.47± 4.79 –	76.43±4.16 –	85.79±4.85 61.76±2.31	86.84±4.13 55.47±1.93
BUPA liver (145+200) × 6	74.26± 5.85 –	75.36±5.22 –	74.86±4.53 61.52±2.59	77.12±4.60 56.65±2.71
CMC (333+511) × 9	72.02± 2.47 –	73.16±3.09 –	70.42±4.62 57.67±4.03	74.19±2.25 51.80±3.67
Credit (383+307) × 19	86.12± 3.53 –	87.23±3.16 –	85.86±3.25 32.18±4.16	87.44±3.71 28.75±3.28
Diabetis (468+300)× 8	75.54± 3.62 –	77.13±3.14 –	76.47±2.61 57.91±2.57	78.78±2.72 45.39±3.06
Flare-Solar (666+400)× 9	66.25± 3.17 –	67.18±2.93 –	67.45±2.69 75.75±3.48	68.74±2.87 68.74±2.79
German (300+700)× 20	72.36± 3.55 –	73.09±2.86 –	71.45±2.69 53.27±3.49	74.71±3.13 48.81±3.83
Heart-Statlog (120+150) × 14	84.15± 5.09 –	85.22±5.96 –	83.36±6.02 48.30±1.06	86.72±5.13 42.26±2.53
Hepatitis (123+32) × 19	83.20± 5.23 –	84.16±6.52 –	83.17±4.33 38.36±2.37	85.68±4.19 32.53±2.22
Image (1300+1010) × 18	93.13± 1.98 –	94.31±2.07 –	93.54±2.16 6.23±1.49	95.32±2.01 4.17±1.08
Ionosphere (126+225) × 34	87.46± 3.34 –	87.78±3.47 –	89.20±3.45 30.07±3.03	90.15±3.27 25.74±2.81
Pima-Indian (500+268) × 8	75.08± 4.10 –	76.11±3.45 –	77.49±5.18 47.26±2.77	79.01±3.21 42.83±3.03
Sonar (97+111) × 60	90.09± 4.85 –	90.92±4.51 –	89.59±4.57 41.83±2.59	92.62±3.86 36.43±2.17
Spect (55+212) × 44	78.14± 3.57 –	78.50±4.11 –	76.92±3.18 51.33±2.91	79.76±3.09 47.34±2.32
Splice (1000+2175) × 60	90.75± 2.31 –	91.18±2.29 –	89.46±2.40 58.89±2.44	91.11±2.18 51.57±3.73
Titanic (150+2050) × 3	76.57± 2.46 –	77.02±2.31 –	77.15±2.34 47.46±3.51	77.83±2.56 40.28±3.84
Twonorm (400+7000) × 20	97.04± 1.57 –	97.35±1.33 –	97.38±1.59 10.23±2.02	97.74±1.15 7.57±1.88
Votes (168+267) × 16	95.04± 2.34 –	96.22±3.17 –	95.18±2.18 32.46±3.06	96.37±2.16 27.91±3.21
Waveform (400+4600) × 21	91.25± 2.23 –	91.67±2.45 –	91.37±3.06 18.41±3.25	92.13±2.19 14.76±2.77
WPBC (46+148) × 34	83.57± 5.62 –	84.16±4.15 –	83.28±4.59 63.57±3.42	85.13±4.11 57.74±2.44

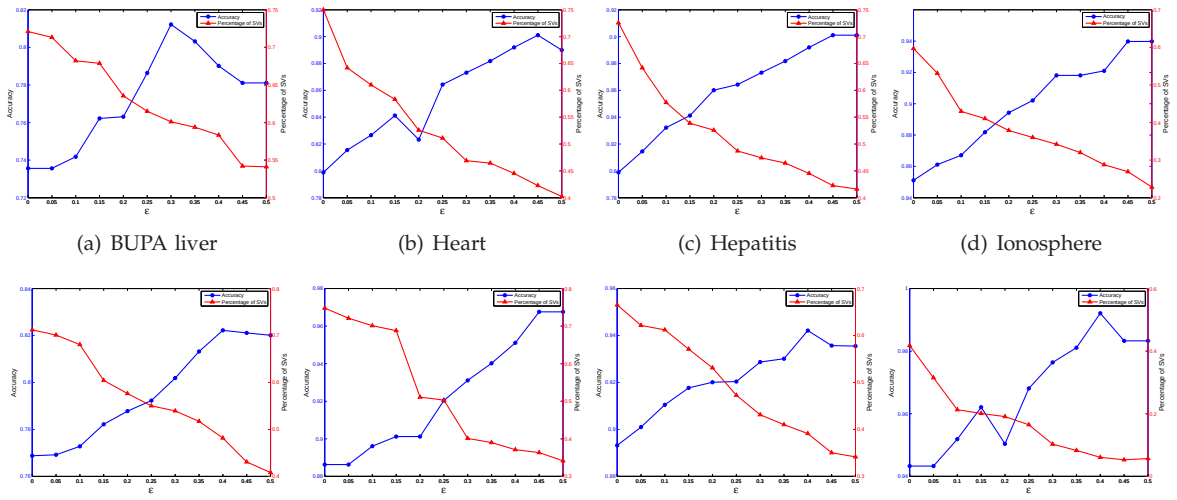
TABLE 2
Description of NDC datasets

Dataset	♯Training data	♯Testing data	♯Features
NDC-500	500	50	32
NDC-700	700	70	32
NDC-900	900	90	32
NDC-1k	1000	100	32
NDC-2k	2000	200	32
NDC-3k	3000	300	32
NDC-4k	4000	400	32
NDC-5k	5000	500	32

TBSVM and C-SVC on several NDC datasets. For NDC-2k, NDC-3k and NDC-5k datasets, we used rectangular kernel[35] using 10% of total data points since TWSVM and TBSVM have to precompute and store the inverse of matrices before training, which will make the experiments run out of memory. However, our NPSVM can be efficiently solved by the SMO technique similar with C-SVC and thus avoid such difficult situation. The results demonstrate that NPSVM performs better than TWSVM, TBSVM and C-SVC in terms of generalization, and NPSVM with SMO technique are more suitable than TWSVM and TBSVM for large-scale problems.

4.3 Text Categorization

In this subsection we further investigate the NPSVM for text categorization (TC) applications and perform experiments on 3 well-known datasets in TC research. The first dataset is gathered from the top 10 largest categories of the mode Apte split of the Reuters-



(a) BUPA liver (b) Heart (c) Hepatitis (d) Ionosphere

(e) Pima-Indian (f) Sonar (g) Splice (h) Twonorm

Fig. 4. Relationships between the cross-validation accuracy and the parameter ε (blue curves), Relationships between the percentage of SVs and ε (red curves).

TABLE 3
Comparison on NDC datasets with RBF kernel

Dataset	TWSVM Train % Test % Time (s) %	TBSVM Train % Test % Time (s) %	C-SVC Train % Test % Time (s) %	NPSVM Train % Test % Time (s)%
NDC-500	93.24 82.36 18.3	94.43 84.75 19.0	92.11 85.45 11.6	95.76 90.17 12.2
NDC-1k	98.37 84.28 36.37	99.76 85.83 37.02	100 94.56 22.8	100 95.69 23.6
NDC-2k[a]	95.83 81.02 8.21	96.17 82.21 8.23	94.24 85.46 4.54	96.25 86.38 4.78
NDC-3k[a]	84.28 77.3 12.81	85.21 78.62 12.16	82.09 78.0 6.35	86.15 81.49 6.49
NDC-5k[a]	87.33 84.53 21.10	89.16 86.81 22.16	89.65 87.07 13.17	90.52 87.74 13.46

[a] A rectangular kernel using 10% of total data points was used.

21578[36], after preprocessing, 9,990 news stories have been partitioned into a training set of 7,199 documents and a test set of 2,791 documents. The 20 Newsgroups (20NG) collection[37] which has about 20,000 newsgroup documents evenly distributed across 20 categories is used as the second dataset. We partition it into ten subsets in equal size and randomly selecting three subsets for training and the remaining seven subsets for testing. The third dataset is the Ohsumed collection[38], where 6,286 documents and 7,643 documents retained for training and testing respectively after removing the duplicate issues. For all the three datasets, stemming, stop word removal, and omitting the words that occur less than 3 times or is shorter than 2 in length are executed in the preprocessing.

Furthermore, since documents have to be transformed into a representation suitable for the classification algorithms, and an effective text representation scheme dominates the performance of TC system, we adopt an efficient schemes[39], the weighted co-contributions of different terms corresponding to the class tendency, to achieve improvements on text representation.

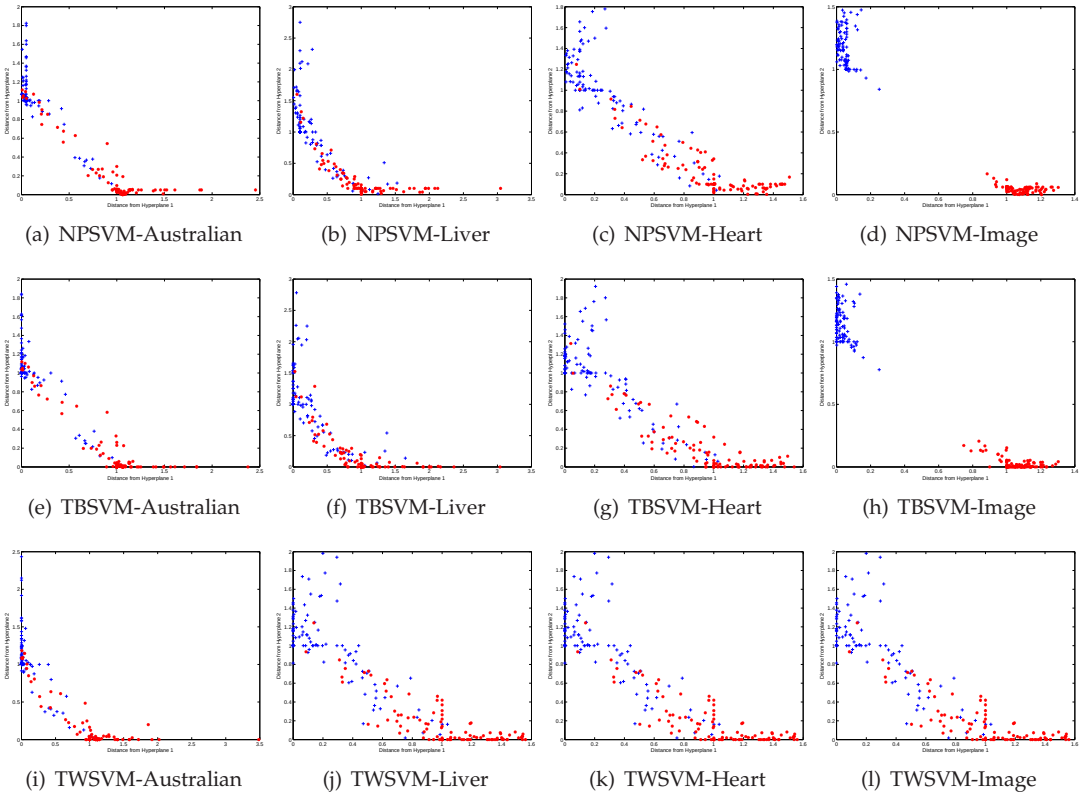
Usually, the precision (P), recall (R) and F_1 are the popular performance metrics used in TC to measure its effectiveness. Since neither precision nor recall is meaningful in isolation of the other, we prefer to use F_1 measure to compute the averaged performance in two ways: micro-averaging (miF_1) and macro-averaging (maF_1), where miF_1 is defined in terms of the micro-averaged values of precision P and recall R, and maF_1 is computed as the mean of category-specific measure F_1^M over all the M target categories:

$$miF_1 = \frac{2PR}{P+R},\ maF_1 = \frac{1}{M}\sum_{i=1}^{M} F_1^M, \qquad (60)$$

We did not conduct experiments using TWSVM and TBSVM as they run out the memory or cost high computing time for these three large scale datasets. The experiment results of NPSVM and C-SVC are given in Table 4. Thus NPSVM achieves improved performance on all the three text corpuses considered in terms of maF_1 and miF_1 performance measures.

5 Concluding Remarks

In this paper, we have proposed a novel nonparallel classifier, termed NPSVM. By introducing the ε-insensitive loss function instead of the quadratic loss function into the two primal problems in TWSVM,



(a) NPSVM-Australian (b) NPSVM-Liver (c) NPSVM-Heart (d) NPSVM-Image

(e) TBSVM-Australian (f) TBSVM-Liver (g) TBSVM-Heart (h) TBSVM-Image

(i) TWSVM-Australian (j) TWSVM-Liver (k) TWSVM-Heart (l) TWSVM-Image

Fig. 5. Two-dimensional projections of NPSVM, TWSVM and TBSVM for 200 test points from the four data sets. "$+$": scatter plot of the positive point; "$*$": scatter plot of the negative point.

TABLE 4
F_1 performance of NPSVM and C-SVC

	Reuters-21578		20NG		Ohsumed	
	miF_1	maF_1	miF_1	maF_1	miF_1	maF_1
NPSVM	0.8615	0.7132	0.8347	0.8178	0.7106	0.5853
C-SVC	0.8524	0.7059	0.8217	0.8125	0.6951	0.5664

NPSVM has several unexpected and incomparable advantages: (1) Two primal problems are constructed implementing the structural risk minimization principle; (2) The dual problems of these two primal problems have the same advantages as that of the standard SVMs, so that the kernel trick can be applied directly, while existing TWSVMs have to construct another two primal problems for nonlinear cases based on the approximate kernel-generated surfaces, furthermore, their nonlinear problems can not degenerate to the linear case even the linear kernel is used; (3) The dual problems have the same elegant formulation with that of standard SVMs and can certainly be solved efficiently by sequential minimization optimization (SMO) algorithm, while existing GEPSVM or TWSVMs are not suitable for large scale problems; (4) It has the inherent sparseness as standard SVMs, the semi-sparseness resulted from TWSVMs is improved to the whole sparseness; (5) Existing TWSVMs are only the special cases of the NPSVM when the parameters of which are appropriately chosen. Our NPSVM degenerates to the initial TWSVM or TBSVM when the parameters of which are appropriately chosen, therefore our models are certainly superior to them theoretically.

The parameters $C_i, i = 1, 2, 3, 4$ introduced are the weights between the regularization term and the empirical risk, ε is the parameter controlling the sparseness. All the parameters can be chosen flexibly, improving the existing TWSVMs in many ways. Computational comparisons between our NPSVM and other methods including TWSVM, TBSVM and C-SVC have been made on lots of datasets, indicating that our NPSVM is not only more sparse, but also more robust and shows better generalization.

Though there are five parameters in our NPSVM, however, for each model we only have an extra parameter ε than TBSVM. The parameter selection seems a difficult problem, we think that the existing efficient methods, such as minimizing the leave one out (LOO) error bound[40], [41] can be applied since the dual problems of our NPSVM has the same formulation with standard SVMs. Besides, for each class, different sparseness can be obtained by using different parameter ε, i.e., ε_+ in problem (52) and ε_- in problem (53). Furthermore, extensions to multi-class classification, regression, semisupervised learning[42], knowledge-based learning[43] are also interesting and

under our consideration.

ACKNOWLEDGMENTS

This work has been partially supported by grants from National Natural Science Foundation of China(NO.11271361, NO.70921061), the CAS/SAFEA International Partnership Program for Creative Research Teams, Major International(Ragional) Joint Research Project(NO.71110107026).

REFERENCES

[1] C. Cortes and V.N. Vapnik, "Support-vector networks, " Mach. Learn., vol. 20, no. 3, pp. 273-297, 1995.
[2] V.N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 1996.
[3] V.N. Vapnik, *Statistical Learning Theory*, New York: John Wiley and Sons, 1998.
[4] T.B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proc. IEEE-INNSENNS Int. Joint Conf. Neural Netw.*, vol. 6. Como, Italy, pp. 348-353, Jul. 2000.
[5] W.S. Noble, "Support vector machine applications in computational biology," in *Kernel Methods in Computational Biology*, B. Schökopf, K. Tsuda, and J.-P. Vert, Eds. Cambridge, MA: MIT Press, 2004.
[6] K.S. Goh, E.Y. Chang and B.T. Li, "Using One-Class and Two-Class SVMs for Multiclass Image Annotation", IEEE Trans. Knowledge and Data Engineering, vol. 17, no. 10, pp. 1333-1346, Oct. 2005.
[7] D. Isa, L.H. Lee, V.P. Kallimani and R. RajKumar, "Text Document Preprocessing with the Bayes Formula for Classification Using the Support Vector Machine", IEEE Trans. Knowledge and Data Engineering, vol. 20, no. 9, pp. 1264-1272, Sep. 2008.
[8] M.B. Karsten, "Kernel Methods in Bioinformatics", *Handbook of Statistical Bioinformatics*, Part 3, pp. 317–334, 2011.
[9] O.L. Mangasarian and E.W. Wild, "Multisurface Proximal Support Vector Classification via Generalized Eigenvalues," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 1, pp. 69-74, Jan. 2006.
[10] R.K. Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 5, pp. 905-910, May 2007.
[11] M.A. Kumar and M. Gopal, "Application of smoothing technique on twin support vector machines," Pattern Recognit. Lett., vol. 29, no. 13, pp. 1842-1848, Oct. 2008.
[12] R. Khemchandani, R.K. Jayadeva, and S. Chandra, "Optimal kernel selection in twin support vector machines," Optim. Lett., vol. 3, no. 1, pp. 77-88, 2009.
[13] M.A. Kumar and M. Gopal, "Least squares twin support vector machines for pattern classification," Expert Syst. Appl., vol. 36, no. 4, pp. 7535-7543, May 2009.
[14] M.A. Kumar, R. Khemchandani, M. Gopal, S. Chandra, "Knowledge based Least Squares Twin support vector machines", Information Sciences, vol. 180, no. 23, pp. 4606-4618, 2010.
[15] Y.H. Shao, C.H. Zhang, X.B. Wang, and N.Y. Deng, "Improvements on twin support vector machines," IEEE Trans. Neural Netw., vol. 22, no. 6, June 2011.
[16] Y.Shao, Z.Wang, W.Chen, N.Deng, "A regularization for the projection twin support vector machine", Knowledge-Based Systems, vol. 37, pp. 203-210, 2013.
[17] Y.Shao, N.Deng, Z.Yang, "Least squares recursive projection twin support vector machine for classification", Pattern Recognition, vol. 45, pp. 2299-2307, 2012.
[18] Y.H. Shao and N.Y. Deng, "A coordinate descent margin based-twin support vector machine for classification", Neural Networks, vol. 25, pp. 114-121, 2012.
[19] X. Peng, "TSVR: An efficient twin support vector machine for regression", Neural Networks, vol. 23, no. 3, pp. 365-372, 2010.
[20] X. Peng, "TPMSVM: A novel twin parametric-margin support vector for pattern recognition", Pattern Recognition, vol. 44, pp. 2678-2692, 2011.
[21] Z.Q. Qi, Y.J. Tian, Y. Shi, "Robust twin support vector machine for pattern classification", Pattern Recognition, vol.46, no. 1, pp. 305-316, 2013.
[22] Z.Q. Qi, Y.J. Tian, Y. Shi, "Laplacian twin support vector machine for semi-supervised classification", Neural Networks, vol. 35, pp. 46-53, 2012.
[23] Z.Q. Qi, Y.J. Tian, Y. Shi, "Twin support vector machine with Universum data", Neural Networks, vol. 36, pp. 112-119, 2012.
[24] Q.L. Ye, C.X. Zhao, N. Ye, X.B. Chen, "Localized twin SVM via convex minimization", Neurocomputing, vol. 74, no. 4, pp. 580-587, 2011.
[25] S. Ghorai, A. Mukherjee, P.K. Dutta, "Nonparallel plane proximal classifier", Signal Processing, vol.89, no.4, pp.510-522, 2009.
[26] J. Platt, "Fast training of support vector machines using sequential minimal optimization". In *Advances in Kernel Methods ł Support Vector Learning*, B. Schölkopf, C.J.C. Burges, and A.J. Smola, Eds. Cambridge, MA: MIT Press, Cambridge, 2000.
[27] O.L. Mangasarian, *Nonlinear Programming*. Philadelphia, PA: SIAM, 1994.
[28] B. Schölkopf and A.J. Smola. *Learning with Kernels*, MIT Press, Cambridge, MA, 2002.
[29] O.L. Mangasarian and D.R. Musicant, "Successive overrelaxation for support vector machines", IEEE Trans. Neural Netw., vol.10, no. 5, pp. 1032-1037, 1999.
[30] C.C. Chang and C.J. Lin, "LIBSVM : a library for support vector machines", ACM Transactions on Intelligent Systems and Technology, vol. 2, no. 3, 27:1-27:27, 2011.
[31] R.E. Fan, P.H. Chen, and C.J. Lin, "Working set selection using second order information for training SVM", Journal of Machine Learning Research, vol. 6, pp. 1889-1918, 2005. URL http://www.csie.ntu.edu.tw/ cjlin/papers/quadworkset.pdf.
[32] MATLAB. 2010. The MathWorks, Inc. http://www.mathworks.com.
[33] C.L. Blake and C.J. Merz, *UCI Repository for Machine Learning Databases*. Dept. Inf. Comput. Sci., Univ. California, Irvine [Online], 1998. Available: http://www.ics.uci.edu/~mlearn/MLRepository.html
[34] D. R. Musicant, NDC: Normally distributed clustered datasets, 1998. Available: http://www.cs.wisc.edu/~musicant/data/ndc.
[35] G. Fung, O. L. Mangasarian, "Proximal support vector machine classifiers", In *Proc. Int. Conf. Knowledge and Data Discovery*, pp. 77-86, 2001.
[36] Reuters-21578, 2007. Available: http://www.daviddlewis.com/resources/testcollections/reuters21578/.
[37] 20 Newsgroups, 2004. Availabel: http://kdd.ics.uci.edu/databases/20newsgroups-/20newsgroups.htm.
[38] Ohsumed, 2007. Available: ftp://medir.ohsu.edu/pub/ohsumed.
[39] Y. Ping, Y. J. Zhou, C. Xue, Y. X. Yang, "Efficient representation of text with multiple perspectives", The Journal of China Universities of Posts and Telecommunications, vol.15, no. 5, pp. 1-12, Sep. 2011.
[40] T. Joachims, "Estimating the generalization performance of an SVM efficientily, " in *Proc. Int. Conf. Machine Learning*, San Franscisco, California, Morgan Kaufmann, pp. 431-438, 2000.
[41] V. N. Vapnik, O. Chapelle, "Bounds on error expectation for SVM". In *Advances in Large-Margin Classifiers, Neural Information Processing*, MIT press, pp. 261-280, 2000.
[42] M.M. Adankon, M. Cheriet, and A. Biem, "Semisupervised least squares support vector machine," IEEE Trans. Neural Netw., vol. 20, no. 12, pp. 1858-1870, Dec. 2009.
[43] K.R. Muller, S. Mika, G. Ratsch, K. Tsuda, B. Schölkopf, "An introduction to kernel-based learning algorithms", IEEE Trans. Neural Netw., vol. 12, no. 2, pp. 181-201, Aug. 2002.