Thai Journal of Mathematics
Volume 19 Number 2 (2021)
Pages 607–622

http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Sparse Pinball Twin Parametric Margin Support Vector Machine

Urairat Deepan[1], Poom Kumam[1,2,*] and Parin Chaipanya[1,2]

[1] *KMUTT-Fixed Point Research Laboratory, Department of Mathematics, Room SCL 802 Fixed Point Laboratory, Science Laboratory Building, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand*
e-mail : urairat.mee@mail.kmutt.ac.th (U. Deepan)

[2] *KMUTT-Fixed Point Theory and Applications Research Group (KMUTT-FPTA), Theoretical and Computational Science Center (TaCS), Science Laboratory Building, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), 126 Pracha-Uthit Road, Bang Mod, Thrung Khru, Bangkok 10140, Thailand*
e-mail : poom.kumam@mail.kmutt.ac.th (P. Kumam); parin.cha@mail.kmutt.ac.th (P. Chaipanya)

Abstract The main purpose of this paper is to construct a twin parametric margin support vector machine combined an $\epsilon-$insensitive loss function for finding a pair of parametric margin hyperplanes that automatically adapts to the parametric noise with arbitrary shape to capture the data structure more accurately. We exhaustively test several UCI datasets demonstrates that our SPTPMSVM is noise insensitive, retains sparsity in most cases. Finally, we present the numerical experiment and compare our model with other models.

MSC: 26B25; 47H05; 47J25; 58A05; 58B20; 90C33

Keywords: Epsilon insensitive loss; noise insensitivity; Twin Parametric Support Vector Machine (TSVM)

Submission date: 11.11.2020 / Acceptance date: 07.12.2020

1. Introduction

The support vector machine (SVM) was proposed by Cortes and Vapnik [1, 2] for classification. The SVM was describe by the maximum margin of hyperplane which the margin means that the minimum of the distance between a point in the training set and hyperplane. We suppose the training sets are defined by

$$T_C = \{(x_i, y_i),\ (i = 1, 2, .., m)\}, \tag{1.1}$$

where the data $x_i \in \mathbb{R}^n$ and labels $y_i \in \{-1, 1\}$ for $i = 1, 2, ..., m$. Let m_1, m_2 are patterns having class label $+1$ and -1 respectively. Thus, the number of data is $m_1 + m_2 = m$. The linear L_1-norm SVM attempts to find a hyperplane of the form

$$w^T x + b = 0, \tag{1.2}$$

*Corresponding author.

Published by The Mathematical Association of Thailand.

where $w \in \mathbb{R}^n,\ b \in \mathbb{R}$. Therefore, the optimization problem of SVM as follows:

$$\begin{aligned} &\min_{(w,b,\xi)} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \\ &\text{subject to} \\ &\quad y_i(w^T x_i + b) + \xi_i \geq 1,\ (i = 1, 2, ..., m), \\ &\quad \xi_i \geq 0,\ (i = 1, 2, ..., m) \end{aligned} \tag{1.3}$$

where $C > 0$ is constant and the term $\sum_{i=1}^{m}\xi_i$ is misclassification error for each $i = 1, 2, ..., m$ [2, 3]. We attempt to find a separating hyperplane $f(x) = w^T x + b = 0,\ w \in \mathbb{R}^n,\ b \in \mathbb{R}$; so as to maximize the margin and minimize the training error by hinge loss function

$$L_{hinge}(x, y, f(x)) = \max(1 - yf(x), 0). \tag{1.4}$$

It is quite easy for finding training error, but sensitive with noise and bad re-sampling [4–9].

In 2014, Huang [3] was introduce the pinball loss function for solved noise insensitive, it follows that

$$L_\tau(u) = \begin{cases} u & , u \geq 0 \\ -\tau u & , u < 0, \end{cases} \tag{1.5}$$

where $u = 1 - yf(x)$ and $\tau \geq 0$ [10–12]. Moreover, Huang was propose ϵ-insensitive zone pinball loss function for solved noise insensitive and sparsity problem, which defined by

$$L_\tau^\epsilon(u) = \begin{cases} u - \epsilon & , \text{ if } u > \epsilon \\ 0 & , \text{ if } -\frac{\epsilon}{\tau} \leq u \leq \epsilon \\ -\tau(u + \frac{\epsilon}{\tau}) & , u < -\frac{\epsilon}{\tau}, \end{cases} \tag{1.6}$$

where $u = 1 - yf(x)$, $\epsilon \geq 0$ and $\tau \geq 0$ are user-defined parameters.

In 2007, Jayadeva [13] proposed twin support vector machine (TWSVM). The concept of TWSVM is to find two non-parallel proximal hyperplanes of other classes for classifier a points in the training set [14–17]. This model is smaller sized QPPs, as compared to single large-sized QPP being solved by SVMs, which TWSVM faster than the model of SVM.

In 2010, Hao improvement of ν-SVM [18] by using $0 \leq \nu \leq 1$ is parametric insensitive for regression and classification. The par$-\nu-$SVM [19] can be useful in many cases, especially when the noise is heteroscedastic.

Later in 2011, Peng [20] introduced the combination models of TWSVM and par$-\nu-$SVM. They determine indirectly the separating hyperplane was solved by two smaller sized support vector machine (SVM), we called that twin parametric-margin support vector machine (TPMSVM). The TPMSVM suitable for the special case, when the data has a heteroscedastic error structure. It clearly that, the speed of TPMSVM more than par$-\nu-$SVM.

For the motivation, we modification of TPMSVM for classification and using the ϵ-insensitive zone pinball loss function. Then, this model can solve noise insensitive in many cases, especially when the data heteroscedastic error structure, that is, the noise strongly depends on the input value.

2. Background

In this section, we present some ideas for modifying our model.

2.1. Par-ν-Support Vector Machine

Suppose two non-parallel separating hyperplanes satisfies $w^Tx+b=\pm(z^Tx+d)$, given $f(x)=w^Tx+b$ and $g(x)=z^Tx+d$ are linear functions, and the classification hyperplane is the bisector of them. The hyperplane $w^Tx+b=0$ separates the data as follows:

$$\begin{aligned} w^Tx+b &\geq (z^Tx+d) \quad \text{for } \forall x \in A \\ w^Tx+b &\leq (z^Tx+d) \quad \text{for } \forall x \in B. \end{aligned}$$

Therefore the QPP as follows:

$$\min_{w,b,z,d,\xi} \frac{1}{2}\|w\|^2 + C(-\nu(\frac{1}{2}\|z\|^2+d)+\frac{1}{m}e^T\xi)$$

$$\text{s.t. } D(Xw+eb) \geq Xz+ed-\xi, \tag{2.1}$$

$$\xi \geq 0, \tag{2.2}$$

where X represents the training data and D is the diagonal matrix with each diagonal element representing the label of the corresponding training point. Here, ξ is the error variable.

2.2. Twin Support Vector Machine (TWSVM)

The linear TWSVM aims to find the two non-parallel hyperplanes given by $w_1^Tx+b_1=0$ and $w_2^Tx+b_2=0$. Let matrix data class $+1$ is A, matrix data class -1 is B of order, $m_1\times n$ and $m_2\times n$ respectively. The TWSVM hyperplanes are obtained by solving the following pair of quadratic programming problems:

$$(TWSVM1) \qquad \begin{aligned} &\min_{w_1,b_1,\xi_2} \frac{1}{2}\|Aw_1+e_1b_1\|^2 + c_1e_2^T\xi_1 \\ &\text{s.t. } -(Bw_1+e_2b_1)+\xi_1 \geq e_2, \\ &\quad \xi_1 \geq 0, \end{aligned} \tag{2.3}$$

and

$$(TWSVM2) \qquad \begin{aligned} &\min_{w_2,b_2,\xi_1} \frac{1}{2}\|Bw_2+e_2b_2\|^2 + c_2e_1^T\xi_2 \\ &\text{s.t. } (Aw_2+e_1b_2)+\xi_2 \geq e_1, \\ &\quad \xi_2 \geq 0, \end{aligned} \tag{2.4}$$

where the constant $c_1 \geq 0$ ($c_2 \geq 0$) is trade-off factor between sum of error vector $\xi_1(\xi_2)$ due to samples of class 1 (class -1) and proximity of hyperplane towards its own class; and e_1 and e_2 are vectors of ones of appropriate dimensions. A new point $x \in \mathbb{R}^n$ is assigned to class $i(i=1$ or $2)$ depending on which of the two aforementioned hyperplanes the point x is closer to, i.e.

$$Class(i) = arg \min_{i=1,2} \frac{|x^Tw_i+b_i|}{\|w_i\|}, \tag{2.5}$$

where $|\cdot|$ is the perpendicular distance of point x from the plane $x^Tw_i+b_i=0, i=1,2$.

2.3. Twin Parametric Support Vector Machine (TPSVM)

Twin Parametric Support Vector Machine [20] considers a pair of parametric-margin hyperplanes $f_1(x) = w_1^T x + b_1 = 0$ and $f_2(x) = w_2^T x + b_2 = 0$ such that each one determines a positive or negative parametric margin. The optimization problems defined by

$$(TPWSVM1) \qquad \begin{aligned} &\min_{w_1,b_1,\xi_1} \frac{1}{2}\|w_1\|^2 + \frac{\nu_1}{m_1}e_1^T(Bw_1 + e_2b_1) + \frac{c_1}{m_1}e_1^T\xi_1 \\ &\text{s.t. } Aw_1 + e_1b_1 \geq 0 - \xi_1, \\ &\quad \xi_1 \geq 0, \end{aligned} \tag{2.6}$$

and

$$(TPWSVM2) \qquad \begin{aligned} &\min_{w_2,b_2,\xi_2} \frac{1}{2}\|w_2\|^2 + \frac{\nu_2}{m_2}e_2^T(Aw_2 + e_1b_2) + \frac{c_2}{m_2}e_2^T\xi_2 \\ &\text{s.t. } Bw_2 + e_2b_2 \geq 0 - \xi_2, \\ &\quad \xi_2 \geq 0, \end{aligned} \tag{2.7}$$

where $c_1 \geq 0(c_2 \geq 0)$ and $\nu_1 \geq 0(\nu_2 \geq 0)$ are the regularization parameters which determine the penalty weights, and ξ_1, ξ_2 represents the error variables.

After optimizing Eqs. (2.6) and (2.7), the classifier for TPSVM is given as follows:

$$f(x) = sign[(\widehat{w}_1 + \widehat{w}_2)^T x + (\widehat{b}_1 + \widehat{b}_2)], \tag{2.8}$$

where $\widehat{w}_1 = \frac{w_i}{\|w_i\|}$ and $\widehat{b}_1 = \frac{b_i}{\|w_i\|}$ for $i = 1, 2$.

3. Sparse Pinball Twin Parametric Margin Support Vector Machine (SPTPMSVM)

In this section, we introduce a spars pinball twin parametric margin support vector machine in linear and nonlinear case. Moreover, calculated the dual problems of two case by lagrangian and Karush-Kuhn-Tucker(KKT) optimal conditions.

3.1. Linear Case

For the linear case, the SPTPMSVM finds two hyperplanes in $\mathbb{R}^m$:

$$f_1(x) = w_1^T x + b_1 = 0 \text{ and } f_2(x) = w_2^T x + b_2 = 0, \tag{3.1}$$

$$\min_{w_1,b_1,\xi_1} \frac{1}{2}[\|w_1\|^2 + b_1^2] + \frac{\nu_1}{m_2}e_2^T(Bw_1 + e_2b_1) + \frac{c_1}{m_1}e_1^T\xi_1 \tag{3.2}$$

$$\begin{aligned} \text{subject to } Aw_1 + b_1e_1 &\geq 0 - (\xi_1 + \epsilon_1 e_1) \\ Aw_1 + b_1e_1 &\leq 0 + \frac{1}{\tau_1}(\xi_1 + \epsilon_1 e_1) \\ \xi_1 &\geq 0, \end{aligned}$$

and

$$\min_{w_2,b_2,\xi_2} \frac{1}{2}[\|w_2\|^2 + b_2^2] - \frac{\nu_2}{m_1} e_1^T (Aw_2 + e_1 b_2) + \frac{c_2}{m_2} e_2^T \xi_2 \tag{3.3}$$
$$\begin{aligned} \text{subject to } Bw_2 + b_2 e_2 &\leq 0 + (\xi_2 + \epsilon_2 e_2) \\ Bw_2 + b_2 e_2 &\geq 0 - \frac{1}{\tau_2}(\xi_2 + \epsilon_2 e_2) \\ \xi_2 &\geq 0, \end{aligned}$$

where ξ_1, ξ_2 are slack vectors and $c_1, c_2 > 0$, $\nu_1, \nu_2 > 0$, and $\tau_1, \tau_2 > 0$ are parameters chosen in advance.

To solve problem (3.2) and (3.3). Consider (3.2) by using Lagrangian function as follow

$$\begin{aligned} L_{(w_1,b_1,\xi_1,\alpha,\beta,\gamma)} &= \frac{1}{2}[\|w_1\|^2 + b_1^2] + \frac{\nu_1}{m_2} e_2^T (Bw_1 + e_2 b_1) + \frac{c_1}{m_1} e_1^T \xi_1 \\ &\quad -\alpha^T ((Aw_1 + b_1 e_1) + \xi_1 + \epsilon_1 e_1) \\ &\quad -\beta^T (-(Aw_1 + b_1 e_1) + \frac{1}{\tau_1}\xi_1 + \frac{1}{\tau_1}\epsilon_1 e_1) - \gamma^T \xi_1 \end{aligned} \tag{3.4}$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_{m_1}), \beta = (\beta_1, \beta_2, ..., \beta_{m_1}), \gamma = (\gamma_1, \gamma_2, ..., \gamma_{m_1}) \geq 0$.

From Karush-Kuhn-Tucker(KKT) optimality condition, we get

$$\frac{\partial L}{\partial w_1} = w_1 + \frac{\nu_1}{m_2} B^T e_2 - A^T \alpha + A^T \beta = 0, \tag{3.5}$$

$$\frac{\partial L}{\partial b_1} = b_1 + \frac{\nu_1}{m_2} e_2^T e_2 - e_1^T \alpha + e_1^T \beta = 0, \tag{3.6}$$

$$\frac{\partial L}{\partial \xi_1} = \frac{c_1}{m_1} e_1 - \alpha - \frac{1}{\tau_1}\beta - \gamma = 0. \tag{3.7}$$

$$Aw_1 + b_1 e_1 + \xi_1 + \epsilon_1 e_1 \geq 0 \qquad , \xi_1 \geq 0 \tag{3.8}$$
$$\alpha^T ((Aw_1 + b_1 e_1) + \xi_1 + \epsilon_1 e_1) = 0 \qquad , \alpha \geq 0 \tag{3.9}$$
$$\beta^T (-(Aw_1 + b_1 e_1) + \frac{1}{\tau_1}\xi_1 + \frac{1}{\tau_1}\epsilon_1 e_1) = 0 \qquad , \beta \geq 0 \tag{3.10}$$
$$\gamma^T \xi_1 = 0 \qquad , \gamma \geq 0. \tag{3.11}$$

From (3.7) we have,

$$\alpha = \frac{c_1}{m_1} e_1 - \frac{1}{\tau_1}\beta - \gamma \tag{3.12}$$

and

$$\beta = \frac{c_1 \tau_1}{m_1} e_1 - \tau_1 \alpha - \tau_1 \gamma. \tag{3.13}$$

Since $\alpha, \beta, \gamma \geq 0$ so, we get $0 \leq \alpha \leq \frac{c_1}{m_1} e_1$ and $0 \leq \beta \leq \frac{c_1 \tau_1}{m_1} e_1$.

Let $u = \begin{bmatrix} w_1 \\ b_1 \end{bmatrix}$, $H = [A \;\; e_1]$, $G = [B \;\; e_2]$ and we combine equation (3.5) and (3.6) to get,

$$\begin{bmatrix} w_1 \\ b_1 \end{bmatrix} + \frac{\nu_1}{m_2} e_2 \begin{bmatrix} B^T \\ e_2^T \end{bmatrix} - \begin{bmatrix} A^T \\ e_1^T \end{bmatrix} \alpha + \begin{bmatrix} A^T \\ e_1^T \end{bmatrix} \beta = 0 \tag{3.14}$$

from equation (3.14) can be rewritten as

$$u + \frac{\nu_1}{m_2} G^T e_2 - H^T(\alpha - \beta) = 0. \tag{3.15}$$

Set $\lambda = \alpha - \beta$, we get that

$$u = H^T \lambda - \frac{\nu_1}{m_2} G^T e_2. \tag{3.16}$$

Then, the dual problem of the problem (3.2) is

$$\begin{aligned} &\max_{\lambda,\alpha,\beta} -\frac{1}{2}\lambda^T H H^T \lambda + \frac{\nu_1}{m_2}\lambda^T H G^T e_2 - \epsilon_1 \alpha^T e_1 - \frac{\epsilon_1}{\tau_1}\beta^T e_1 \\ &\text{subject to} \quad 0 \le \alpha \le \frac{c_1}{m_1} e_1, \ 0 \le \beta \le \frac{c_1 \tau_1}{m_1} e_1, \ \lambda = \alpha - \beta. \end{aligned} \tag{3.17}$$

Similarly, let $v = \begin{bmatrix} w_2 \\ b_2 \end{bmatrix}$, $P = [A \ \ e_1]$, $Q = [B \ \ e_2]$ such that $v = \frac{\nu_2}{m_1} P^T e_1 - Q^T \mu$ where $\mu = \omega - \sigma$. The dual problem of (3.3) get that,

$$\begin{aligned} &\max_{\mu,\omega,\sigma} -\frac{1}{2}\mu^T Q Q^T \mu + \frac{\nu_2}{m_1}\mu^T P Q^T e_1 - \epsilon_2 \omega^T e_2 - \frac{\epsilon_2}{\tau_2}\sigma^T e_2 \\ &\text{subject to} \quad 0 \le \omega \le \frac{c_2}{m_2} e_2, \ 0 \le \sigma \le \frac{c_2 \tau_2}{m_2} e_2, \ \mu = \omega - \sigma. \end{aligned} \tag{3.18}$$

3.2. Non-Linear Case

In this case, we extend SPTPMSVM to the non-linear case [13]. Suppose that

$$\begin{aligned} K(x^T, C^T) z_1 + b_1 &= 0, \\ \text{and } K(x^T, C^T) z_2 + b_2 &= 0, \end{aligned} \tag{3.19}$$

where $C = \begin{bmatrix} A_{m_1 \times n} \\ B_{m_2 \times n} \end{bmatrix}$.

Where, K is the kernel function which can be chosen according to the specific task at hand and $z_1, z_2 \in \mathbb{R}^{m_1+m_2}$. For instance, $K(x^T, C^T) = x^T C^T$ and define $C^T z_1 = w_1$ and $C^T z_2 = w_2$, then we get the linear planes.

From (3.2) and (3.3), we formulate in case of non-linear case:

$$\begin{aligned} \min_{z_1,b_1,\xi_1} \ & \frac{1}{2}[\|K(A, C^T) z_1\|^2 + b_1^2] + \frac{\nu_1}{m_2} e_2^T (K(B, C^T) z_1 + e_2 b_1) + \frac{c_1}{m_1} e_1^T \xi_1 \\ & \text{subject to } K(A, C^T) z_1 + b_1 e_1 \ge 0 - (\xi_1 + \epsilon_1 e_1) \\ & K(A, C^T) z_1 + b_1 e_1 \le 0 + \frac{1}{\tau_1}(\xi_1 + \epsilon_1 e_1) \\ & \xi_1 \ge 0, \end{aligned} \tag{3.20}$$

and

$$\begin{aligned} \min_{z_2,b_2,\xi_2} \ & \frac{1}{2}[\|K(B, C^T) z_2\|^2 + b_2^2] - \frac{\nu_2}{m_1} e_1^T (K(A, C^T) z_2 + e_1 b_2) + \frac{c_2}{m_2} e_2^T \xi_2 \\ & \text{subject to } K(B, C^T) z_2 + b_2 e_2 \le 0 + (\xi_2 + \epsilon_2 e_2) \\ & K(B, C^T) z_2 + b_2 e_2 \ge 0 - \frac{1}{\tau_2}(\xi_2 + \epsilon_2 e_2) \\ & \xi_2 \ge 0, \end{aligned} \tag{3.21}$$

where ξ_1, ξ_2 are slack vectors and $c_1, c_2 > 0$, $\nu_1, \nu_2 > 0$ and $\tau_1, \tau_2 > 0$ are parameters chosen in advance.

We can write the dual problem of problem (3.20), it's follows

$$\begin{aligned} \max_{\lambda,\alpha,\beta} & -\frac{1}{2}\lambda^T SS^T\lambda + \frac{\nu_1}{m_2}\lambda^T SR^T e_2 - \epsilon_1\alpha^T e_1 - \frac{\epsilon_1}{\tau_1}\beta^T e_1 \\ \text{subject to} & \quad 0 \le \alpha \le \frac{c_1}{m_1}e_1,\ 0 \le \beta \le \frac{c_1\tau_1}{m_1}e_1,\ \lambda = \alpha - \beta. \end{aligned} \tag{3.22}$$

Where, $S = [K(A, C^T)\ e_1]$ and $R = [K(B, C^T)\ e_2]$. The augmented vector $u = \begin{bmatrix} z_1 \\ b_1 \end{bmatrix}$ and get,

$$u = S^T\lambda - \frac{\nu_1}{m_2}R^T e_2. \tag{3.23}$$

We note that we apply well-conditioning, when required, in the same manner as in (3.16). Similarly, the dual problem for (3.21) is:

$$\begin{aligned} \max_{\mu,\omega,\sigma} & -\frac{1}{2}\mu^T NN^T\mu + \frac{\nu_2}{m_1}\mu^T LN^T e_1 - \epsilon_2\omega^T e_2 - \frac{\epsilon_2}{\tau_2}\sigma^T e_2 \\ \text{subject to} & \quad 0 \le \omega \le \frac{c_2}{m_2}e_2,\ 0 \le \sigma \le \frac{c_2\tau_2}{m_2}e_2,\ \mu = \omega - \sigma. \end{aligned} \tag{3.24}$$

Here, $L = [K(A, C^T)\ e_1]$ and $N = [K(B, C^T)\ e_2]$. Further, the augmented vector $v = \begin{bmatrix} z_2 \\ b_2 \end{bmatrix}$

is calculated by the relation:

$$v = \frac{\nu_2}{m_1}L^T e_1 - N^T\mu. \tag{3.25}$$

Once we obtain the required parameters from problems (3.22) and (3.24), we use the decision function to predict the class of a new sample $x \in \mathbb{R}^n$ by assigning it to class l, $(l = 1, 2)$ in a manner similar to the linear case.

4. Noise Insensitivity

Here we explain, from an analytical perspective, how incorporating the ϵ-insensitive pinball function leads to noise insensitivity.

For the sake of brevity, we consider SPTPMSVM (3.2) for the linear case. Consider the generalized sign function, $sgn_\tau^\epsilon(u)$ corresponding to (1.6):

$$sgn_\tau^\epsilon(u) = \begin{cases} \{1\} & \text{, if } u > \epsilon, \\ [0, 1] & \text{, if } u = \epsilon, \\ \{0\} & , -\frac{\epsilon}{\tau} < u < \epsilon, \\ [-\tau,\ 0] & \text{, if } u = -\frac{\epsilon}{\tau} \\ \{-\tau\} & \text{, if } u < -\frac{\epsilon}{\tau}, \end{cases} \tag{4.1}$$

$sgn_\tau^\epsilon(u)$ is the subgradient of the $\epsilon-$insensitive pinball loss function and, hence, the optimality condition for (3.2) can be written as:

$$0 \in w_1 + \frac{\nu_1}{m_2}\sum_{j=1}^{m_2} x_j + \frac{c_1}{m_1}\sum_{i=1}^{m_1} sgn_{\tau_1}^{\epsilon_1}(0 - (w_1^T x_i + b_1)x_i), \tag{4.2}$$

where 0 is the vector which has all its components equal to zero and $x_i \in A,\ x_j \in B$.

For a give w_1, b_1, we partition the index set into five sets

$$
\begin{aligned}
S_0^{w_1,b_1} &= \{i : w_1^T x_i + b_1 < -\epsilon\} \\
S_1^{w_1,b_1} &= \{i : w_1^T x_i + b_1 = -\epsilon\} \\
S_2^{w_1,b_1} &= \{i : -\epsilon < w_1^T x_i + b_1 < \frac{\epsilon_1}{\tau_1}\} \\
S_3^{w_1,b_1} &= \{i : w_1^T x_i + b_1 = \frac{\epsilon_1}{\tau_1}\} \\
S_4^{w_1,b_1} &= \{i : w_1^T x_i + b_1 > \frac{\epsilon_1}{\tau_1}\}.
\end{aligned}
$$

Here, $i = 1, 2, ..., m_1$. The data samples in $S_2^{w_1,b_1}$ do not contribute to w_1 since the sub-gradient at these data samples is zero, as is evident from (4.1). Thus, $S_2^{w_1,b_1}$ directly affects sparsity of the model. Set $S_2^{w_1,b_1}$ is dependent on the value of ϵ. As ϵ approaches 0 sparsity is lost whereas if $\epsilon \to \infty$, more samples lie in $S_2^{w_1,b_1}$ and, as a result, we gain sparsity. With the above notations and the existence of $\psi_1 \in [0, 1]$ and $\theta_1 \in [-\tau_1, 0]$ Eq. (4.2) can be rewritten as:

$$\frac{m_1}{c_1} w_1 + \frac{m_1}{c_1} \frac{\nu_1}{m_2} \sum_{j=1}^{m_2} x_j + \sum_{i \in S_0^{w_1,b_1}} x_i + \sum_{i \in S_1^{w_1,b_1}} \psi x_i - \sum_{i \in S_3^{w_1,b_1}} \theta x_i - \tau \sum_{i \in S_4^{w_1,b_1}} x_i = 0. \quad (4.3)$$

The above condition shows that when the value of ϵ is fixed, τ controls the number of samples in the sets $S_0^{w_1,b_1}, S_1^{w_1,b_1}, S_2^{w_1,b_1}, S_3^{w_1,b_1}$, and $S_4^{w_1,b_1}$. However, since the number of data samples in $S_1^{w_1,b_1}$ and $S_3^{w_1,b_1}$ are much fewer than in the other sets, we are primarily concerned with sets $S_0^{w_1,b_1}, S_2^{w_1,b_1}$ and $S_4^{w_1,b_1}$. When τ is small, the number of samples in $S_4^{w_1,b_1}$ is quite large while the other sets have fewer data samples, thus making the result sensitive to feature noise in the samples. On the contrary, having a larger value imparts many data samples to all the five sets and the result is less sensitive to zero mean feature noise.

Proposition 4.1. *If the optimization problem* (3.17) *or* (3.22) *has a solution, parameters must satisfy the condition* $\nu_1 + b_1 \leq c_1$*.*

Proposition 4.2. *Let* p_0 *denote the number of positive samples* $x_i (i = 1, 2, ..., m_1)$ *in* $S_0^{w_1,b_1}$*. We have*

$$\frac{p_0}{m_1} \leq 1 - \frac{1 - \frac{b_1 + \nu_1}{c_1}}{(1 + \tau_1)}. \quad (4.4)$$

Proof. Consider an arbitrary sample $x_{i_0} \in S_0^{w_1,b_1}$, $(1 \leq i_0 \leq m_1)$. From the KKT condition (3.10) and (3.11), we have $\beta_{i_0} = \gamma_{i_0} = 0$. From (3.7), we have $\alpha_{i_0} = \frac{c_1}{m_1}$ such that $\lambda_{i_0} = \alpha_{i_0} - \beta_{i_0} = \frac{c_1}{m_1}$. Also, from the KKT condition (3.6), we have

$$b_1 + \nu_1 - (\sum_{i \in S_0^{w_1,b_1}} \lambda_i + \sum_{i \notin S_0^{w_1,b_1}} \lambda_i) = 0 \quad (4.5)$$

and then,

$$b_1 + \nu_1 - (\frac{p_0 c_1}{m_1} + \sum_{i \notin S_0^{w_1,b_1}} \lambda_i) = 0. \tag{4.6}$$

Since $\alpha_i \geq 0,\ \beta_i \geq 0$ and (3.7), we will consider

Case I

$$\begin{aligned}
\frac{c_1}{m_1} - \alpha_i - \frac{1}{\tau_1}\beta_i - \gamma_i &= 0 \\
\frac{c_1}{m_1} - \alpha_i - \frac{1}{\tau_1}(\beta_i + \alpha_i - \alpha_i) - \gamma_i &= 0 \\
\frac{c_1}{m_1} - \alpha_i - \frac{1}{\tau_1}(\alpha_i - (\alpha_i - \beta_i)) - \gamma_i &= 0 \\
\frac{c_1}{m_1} - \alpha_i - \gamma_i + \frac{1}{\tau_1}(\lambda_i - \alpha_i) &= 0 \\
\frac{c_1}{m_1} - \gamma_i + \frac{1}{\tau_1}\lambda_i + (-1 - \frac{1}{\tau_1})\alpha_i &= 0 \\
\frac{c_1}{m_1} + \frac{1}{\tau_1}\lambda_i + (-1 - \frac{1}{\tau_1})\alpha_i &= \gamma_i.
\end{aligned}$$

Since $\gamma \geq 0$, we have

$$-\frac{\tau_1 c_1}{m_1} \leq \lambda_i. \tag{4.7}$$

Case II

$$\begin{aligned}
\frac{c_1}{m_1} - \alpha_i - \frac{1}{\tau_1}\beta_i - \gamma_i &= 0 \\
\frac{c_1}{m_1} - (\alpha_i - \beta_i + \beta_i) - \frac{1}{\tau_1}\beta_i - \gamma_i &= 0 \\
\frac{c_1}{m_1} - (\lambda_i + \beta_i) - \frac{1}{\tau_1}\beta_i - \gamma_i &= 0 \\
\frac{c_1}{m_1} - \gamma_i - \lambda_i + (-1 - \frac{1}{\tau_1})\beta_i &= 0 \\
\frac{c_1}{m_1} - \lambda_i + (-1 - \frac{1}{\tau_1})\beta_i &= \gamma_i.
\end{aligned}$$

Since $\gamma_i \geq 0$, we have

$$\lambda_i \leq \frac{c_1}{m_1}. \tag{4.8}$$

From equation (4.7) and (4.8), we get that

$$-\frac{\tau_1 c_1}{m_1} \leq \lambda_i \leq \frac{c_1}{m_1}. \tag{4.9}$$

Since p_0 denotes the number of samples in $S_0^{w_1,b_1}$, by (4.9), we have

$$\begin{aligned}
-\frac{\tau_1 c_1}{m_1}(m_1 - p_0) \leq &\quad \textstyle\sum_{i \notin S_0^{w_1,b_1}} \lambda_i &&\leq \frac{c_1}{m_1}(m_1 - p_0) \\
-\frac{\tau_1 c_1}{m_1}(m_1 - p_0) \leq &\quad b_1 + \nu_1 - \frac{p_0 c_1}{m_1} &&\leq \frac{c_1}{m_1}(m_1 - p_0) \\
-b_1 - \nu_1 - \frac{\tau_1 c_1}{m_1}(m_1 - p_0) \leq &\quad -\frac{p_0 c_1}{m_1} &&\leq -b_1 - \nu_1 \frac{c_1}{m_1}(m_1 - p_0) \\
b_1 + \nu_1 - \frac{c_1}{m_1}(m_1 - p_0) \leq &\quad \frac{p_0 c_1}{m_1} &&\leq b_1 + \nu_1 + \frac{\tau_1 c_1}{m_1}(m_1 - p_0) \\
\frac{m_1(b_1 + \nu_1 - \frac{c_1}{m_1}(m_1 - p_0))}{c_1} \leq &\quad p_0 &&\leq \frac{m_1(b_1 + \nu_1 + \frac{\tau_1 c_1}{m_1}(m_1 - p_0))}{c_1}
\end{aligned}$$

which gives us

$$\frac{m_1(b_1 + \nu_1 - \frac{c_1}{m_1}(m_1 - p_0))}{c_1} \leq p_0$$

such that

$$\frac{b_1 + \nu_1}{c_1} \leq 1$$

that is

$$b_1 + \nu_1 \leq c_1. \tag{4.10}$$

The second condition gives us

$$p_0 \leq \frac{m_1(b_1 + \nu_1 + \frac{\tau_1 c_1}{m_1}(m_1 - p_0))}{c_1}$$

such that

$$\frac{p_0}{m_1} \leq \frac{b_1 + \nu_1 + \tau_1 c_1}{c_1(1 + \tau_1)}$$

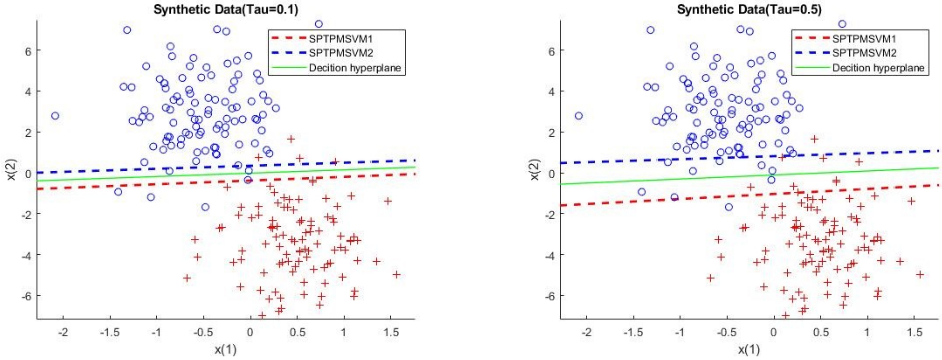
that is

$$\frac{p_0}{m_1} \leq 1 - \frac{1 - \frac{b_1 + \nu_1}{c_1}}{(1 + \tau_1)}. \tag{4.11}$$

■

As is evident, the above proposition places an upper bound on the number of samples in $S_0^{w_1,b_1}$, when τ becomes small, p_0 gets smaller and the result becomes more sensitive to feature noise since a lot fewer data samples are distributed in sets other than $S_4^{w_1,b_1}$. As a result, feature noise around the decision boundary significantly affects the classification results. A similar analysis holds for SPTPSVM2 problems (3.3) and (3.21).

The SPTPMSVM gives penalties on the misclassified points (i.e., $\xi_1 > 0$ or $\xi_2 > 0$) and no weight on the correctly classified points (i.e., $\xi_1 = 0$ or $\xi_2 = 0$). By contrast, the SPTPMSVM gives weights on both the correctly classified and misclassified points by the pinball loss. For the SPTPMSVM, with the increase of parameter τ , the weights on the correctly classified points become great. Thus, the margin between the positive hyperplane and negative hyperplane becomes large.



(A) (B)

FIGURE 1. Illustrations of (a) SPTPMSVM as $\tau = 0.1$, (b) SPTPMSVM as $\tau = 0.5$ on a $2 - D$ artificial data set.

5. Numerical Experiment

In this section, we proposed the above figures represent the noise insensitive properties compared between SPTPMSVM and C-SVM and Bar graph of the accuracies for four algorithms on the $2 - D$ artificial data. Moreover, presented the numerical experiment of data from UCI datasets.

5.1. Synthetic Dataset

The objective of our SPTPMSVM is to be able to deal with noise around the decision boundary while retaining sparsity. To illustrate the noise insensitivity performance consider Figure 2, we suppose a two dimensional synthetic dataset with equal number of samples from two Gaussian distributions: $x_i,\ i \in \{i : y_i = 1\} \sim \mathrm{N}(\mu_1, \sum_1)$ and $x_i,\ i \in \{i : y_i = -1\} \sim \mathrm{N}(\mu_2, \sum_2)$ where $\mu_1 = [0.5, -3]^T, \mu_2 = [-0.5, 3]^T$ and $\sum_1 = \sum_2 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}$. The Bayes classifier for the given Gaussian distribution is $f_c(x) = 2.5x(1) - x(2)$.

We now add noise to the dataset, with each noise sample drawn from the Gaussian distribution $\mathrm{N}(\mu_n, \sum_n)$ where $\mu_n = [0, 0]^T$ and $\sum_n = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}$.

From Figure 2 we have varying number of noise samples, from $r = 0$ (noise free) to $r = 0.5$. Where, r is the ratio of total number of noisy samples to the total number of samples originally in the dataset (including both classes). We see that, if the amount of noise increase (from $r = 0$ to $r = 0.5$) then, the hyperplanes of C-SVM and TPMSVM start deviating from the ideal slope of 2.5 whereas the deviation in the slopes of hyperplanes (with fixed values of $\tau = 0.5$ and $\epsilon = 0.05$) is significantly lesser in our SPTPMSVM. This implies the sensitivity of the C-SVM models to noise around the boundary.

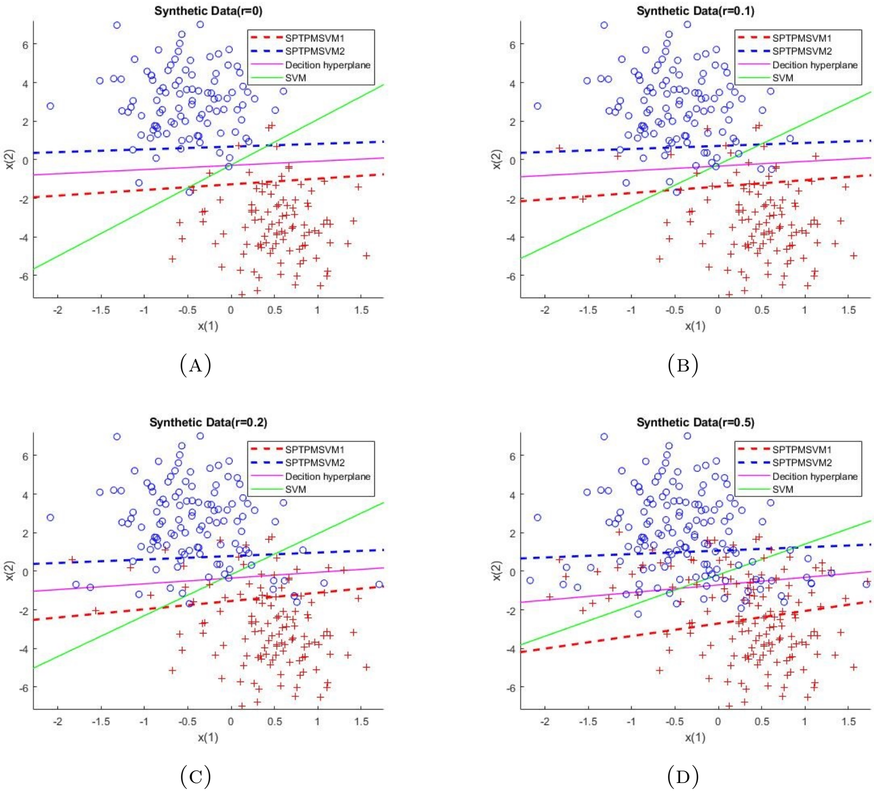


FIGURE 2. The above four figures demonstrate the noise insensitive properties possessed by our SPTPMSVM as compared to C-SVM.

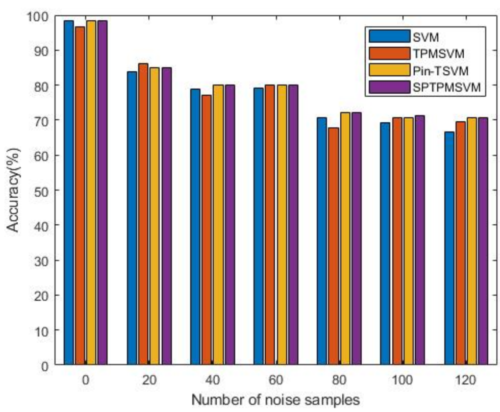


FIGURE 3. Bar graph of the accuracies for four algorithms on the $2-D$ artificial data. x-axis: number of noise points. y-axis: testing accuracy.

5.2. UCI Datasets

We using our SPTPMSVM model with 5 UCI datasets [21] to exhibit the accuracy, noise insensitivity and sparsity of our model. All of the experiments have been performed on MATLAB R2020a on a Windows 10 machine with an Intel i3 Processor (2.0 GHz) with 16 GB RAM. To solve our SPTPMSVM model with lower computational complexity. Suppose that $c_1 = c_2 = c, \tau_1 = \tau_2 = \tau$, and $\epsilon_1 = \epsilon_2 = \epsilon$. In all our experiments, c is chosen from the set $\{10^i : i = -3, -2, -1, ..., +1, +2, +3\}$, τ is set $\{0.01, 0.1, 0.2, 0.5, 1\}$ and ϵ is set $\{0, 0.05, 0.1, 0.5\}$ [22–26]. We calculated accuracy by :

$$\text{Acc} \ = \frac{\text{Number of datas test set} - \ \text{Number of datas missclass}}{\text{Number of datas test set}} \times 100. \qquad (5.1)$$

Table 1. Performance comparisons of four algorithms on UCI data sets.

Data sets	Noise(r)	SVM	TPMSVM	pin-TPMSVM	SPTPMSVM			
		Accuracy Time	Accuracy Time	Accuracy Time	Accuracy Time			
					$\epsilon = 0$	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.5$
Heart (270x13)	r = 0	**86.25** 0.01	**86.25** **0.008**	85.00 0.05	**86.25** 0.06	**86.25** 0.05	85.00 0.04	**86.25** 0.06
	r = 0.1	83.75 0.02	86.25 **0.01**	86.25 0.08	85.00 0.05	**87.50** 0.1	85.00 0.05	**87.50** 0.09
	r = 0.5	85.00 0.30	85.00 **0.02**	86.25 0.36	86.25 0.37	**87.50** 0.34	**87.50** 0.36	**87.50** 0.39
Monk3 (432x6)	r = 0	78.37 0.04	83.17 **0.02**	83.77 0.27	83.77 0.19	**84.38** 0.47	**84.38** 0.51	**84.38** 0.49
	r = 0.1	78.37 0.05	**85.58** **0.02**	84.38 0.41	84.38 0.34	**85.58** 0.34	84.98 0.39	**85.58** 0.56
	r = 0.5	80.77 0.20	84.98 **0.04**	**85.58** 1.24	**85.58** 1.17	**85.58** 1.40	**85.58** 1.46	**85.58** 1.69
Fertility (100x9)	r = 0	**87.50** 0.12	71.88 **0.01**	75.00 0.02	75.00 0.02	75.00 0.02	84.38 0.02	75.00 0.02
	r = 0.1	**87.50** 0.02	78.13 **0.01**	81.25 2.00	81.25 0.02	78.13 0.02	84.38 0.03	75.00 0.02
	r = 0.5	**87.50** **0.01**	84.38 **0.01**	78.13 0.05	84.38 0.04	78.13 0.04	**87.50** 0.05	78.13 0.04
Breast cancer (699x10)	r = 0	**98.09** **0.12**	**98.09** 0.15	**98.09** 1.17	**98.09** 0.76	**98.09** 0.72	**98.09** 0.63	**98.09** 0.66
	r = 0.1	**98.09** 1.53	**98.09** **0.17**	**98.09** 1.79	**98.09** 1.20	**98.09** 0.88	**98.09** 0.91	**98.09** 0.88
	r = 0.5	**87.50** **0.01**	84.38 0.01	78.13 0.05	84.38 0.04	78.13 0.04	**87.50** 0.05	78.13 0.04
Banknote (1,372x5)	r = 0	**98.31** 0.71	84.96 **0.12**	94.92 3.80	95.13 2.53	95.34 2.71	95.76 2.87	95.34 2.53
	r = 0.1	**98.09** 0.55	85.38 **0.09**	94.70 3.40	94.70 5.23	94.92 4.36	95.13 4.25	94.92 4.26
	r = 0.5	**87.50** **0.01**	84.38 0.01	78.13 0.05	84.38 0.04	78.13 0.04	**87.50** 0.05	78.13 0.04

From Table 1, we can learn that our proposed SPTPMSVM yields the best prediction accuracy on 3 of 5 data sets. Among the 15 cases, our proposed SPTPMSVM achieves the best prediction accuracy in the 9 cases. Followed by the SVM, TPMSVM and pin-TPMSVM with comparable performance.

TABLE 2. Sparsity on UCI data sets with linear kernel.

Datasets	ϵ	pin-TPMSVM		SPTPMSVM	
		$\tau=0$		$\tau=0$	
Heart	0	60	63	60	81
(270x13)	0.1			57	59
	0.2			51	54
	0.3			48	48
	0.4			40	41
	0.5			36	39
Monk	0	197	208	197	200
(432x6)	0.05			185	194
	0.1			171	179
	0.2			127	132
	0.3			119	118
	0.5			106	109
Banknote	0	360	353	360	353
(1372x5)	0.05			306	312
	0.1			259	272
	0.2			225	243
	0.3			201	216
	0.5			181	187
Fertility	0	69	11	69	11
(100x9)	0.05			65	10
	0.1			60	10
	0.2			55	10
	0.3			53	10
	0.5			51	8
Breast cancer	0	19	32	19	32
(699x10)	0.05			18	30
	0.1			18	27
	0.2			17	21
	0.3			17	15
				17	15

In Table 2, the sparsity of our proposed SPTPMSVM is analyzed as compared to the original pin-TPMSVM for the linear. In tables, the two columns under each model show the number of non-zero dual variables corresponding to each of the two separating hyperplanes. From both the tables it is evident that our SPTPMSVM is more sparse as compared to the original pin-TPMSVM.

CONCLUSION

Finally, we have completed constructing a Twin Parametric Margin Support Vector Machine combined with $\epsilon-$insensitive loss function for finding a pair of parametric margin hyperplanes that automatically adapts to the parametric noise. Compared to the SVM, our proposed SPTPMSVM is noise insensitive and sparse at the same time. The validity of our proposed SPTPMSVM is demonstrated by numerical experiments performed on several UCI benchmark and synthetic datasets for linear cases. Numerical experiments clearly show that the classification accuracy of our SPTPMSVM better than the accuracy of pin-TPMSVM and SVM in most of the cases, and simultaneously maintain sparsity and insensitivity to outliers. Several parameters need to be regularized in our SPTPMSVM.

Acknowledgements

The authors acknowledge the financial support provided by National Research Council of Thailand and Mathematical Association of Thailand and King Mongkut's University of Technology Thonburi through the KMUTT 55^{th} Anniversary Commemorative Fund. The first author was supported by the Petch Pra Jom Klao Master's Degree Research Scholarship from King Mongkut's University of Technology Thonburi Grant No.50/2561. This project is supported by the Theoretical and Computational Science (TaCS) Center under Computational and Applied Science for Smart Innovation research Cluster (CLASSIC), Faculty of Science, KMUTT. Dr. Parin Chaipunya is supported by Thailand Science Research and Innovation (TSRI) Basic Research Fund: Fiscal year 2021 under project number 64A306000005. The guidance of Prof. Dr. Rabian Wangkeeree is gratefully acknowledged.

References

[1] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273–297.

[2] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[3] X. Huang, L. Shi, J.A.K. Suykens, Support vector machine classifier with pinball loss, IEEE Transactions on Pattern Analysis 36 (5) (2014) 984–997.

[4] J. Bi, T. Zhang, Support vector classification with input data uncertainty, Vancouver (2005) 161–168.

[5] I. Guyon, N. Matic, V. Vapnik, Discovering informative patterns and data cleaning, Advances in Knowledge Discovery and Data Mining (1996) 181–203.

[6] R. Herbrich, J. Weston, Adaptive margin support vector machines for classification, Ninth International Conference on Artificial Neural Networks ICANN99 2 (470) (1999) 880–885.

[7] Q. Song, W. Hu, W. Xie, Robust support vector machine with bullet hole image classification, IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews) 32 (4) (2002) 440–448.

[8] P.K. Shivaswamy, C. Bhattacharyya, A.J. Smola, Second order cone programming approaches for handling missing and uncertain data, Journal of Machine Learning Research 7 (2006) 1283–1314.

[9] J. Zhang, Y. Wang, A rough margin based support vector machine, Information Sciences 178 (9) (2008) 2204–2214.

[10] I. Steinwart, A. Christmann, How SVMs can estimate quantiles and the median, NIPS, Vancouver (2007) 305–312.

[11] V. Jumutc, X. Huang, J.A.K. Suykens, Fixed-size Pegasos for hinge and pinball loss SVM, The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX (2013) 1122–1128.

[12] I. Steinwart, A. Christmann, Estimating conditional quantiles with the help of the pinball loss, Statistics Theory (Math. ST) 17 (1) (2011) 211–225.

[13] Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 29

(5) (2007) 905–910.

[14] G. Fung, O.L. Mangasarian, Proximal support vector machine classifiers, Discovery Data Mining (2001) 77–86.

[15] S. Ghorai, A. Mukherjee, P.K. Dutta, Nonparallel plane proximal classifier, Signal Process 89 (4) (2009) 510–522.

[16] G.M. Fung, O.L. Mangasarian, Multicategory proximal support vector machine classifiers, Machine Learning 59 (2005) 77–97.

[17] Jayadeva, R. Khemchandani, S. Chandra, Fuzzy multi-category proximal support vector classification via generalized eigenvalues, Soft Comput 11 (7) (2007) 679–685.

[18] B. Scholkopf, A.J. Smola, R.C. Williamson, P.L. Bartlett, New support vector algorithms, Neural Computation 12 (5) (2000) 1–39.

[19] P.Y. Hao, New support vector algorithms with parametric insensitive/margin model, Neural Networks23 (1) (2010) 60–73.

[20] X. Peng, TPMSVM: A novel twin parametric-margin support vector machine for pattern recognition, Pattern Recognition 44 (2011) 2678–2692.

[21] K. Bache, M. Lichman, UCI Machine Learning Repository, [Online] (2013) http://archive.ics.uci.edu/ml(1990).

[22] I. Steinwart, A. Christmann, Support Vector Machines, Information Science and Statistics, Springer-Verlag, 2008.

[23] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge, 2000.

[24] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge, 2004.

[25] R. Khemchandani, Jayadeva, S. Chandra, Optimal kernel selection in twin support vector machines, Optimization Letters 3 (1) (2009) 77–88.

[26] X. Huang, L. Shi, J.A.K. Suykens, Sequential minimal optimization for SVM with pinball loss, Neurocomputing 149 (2015) 1596–1603.