

Polynomial Smooth Twin Support Vector Machines

Shifei Ding^{1,2,*}, Huajuan Huang^{1,2}, Xinzheng Xu¹ and Jian Wang³

¹ School of Computer Science and Technology, China, University of Mining and Technology, Xuzhou 221116, China

² Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Science, Beijing 100190, China

³ College of Geomatics, Shandong University of Science and Technology, Qingdao 266590, China

Received: 3 Sep. 2013, Revised: 6 Dec. 2013, Accepted: 7 Dec. 2013

Published online: 1 Jul. 2014

Abstract: Smoothing functions can transform the unsmooth twin support vector machines (TWSVM) into smooth ones, and thus better classification results can be obtained. It has been one of the key problems to seek a better smoothing function in this field for a long time. In this paper, a novel version for smooth TWSVM, termed polynomial smooth twin support vector machines (PSTWSVM), is proposed. In PSTWSVM, using the series expansion, a new class of polynomial smoothing is proposed, and then their important properties are discussed. It is shown that the approximation accuracy and smoothness rank of polynomial functions can be as high as required. Subsequently, the polynomial functions are adopted to convert the original constrained quadratic programming problems of TWSVM into unconstrained minimization problems, and then are solved by the well-known Newton-Armijo algorithm. The effectiveness of the proposed method is demonstrated via experiments on synthetic and real-word benchmark datasets.

Keywords: Twin support vector machines, Polynomial function, Smoothing, Newton-Armijo.

1 Introduction

Support vector machine (SVM) presented by Vapnik and co-worker [1] is a computationally powerful kernel-based tool for binary data classification and regression. Because the theory of SVM is based on the idea of structural risk minimization principle, SVM has successfully solved the high dimensionality and local minimum problems. Therefore, compared with other machine learning methods, such as artificial neural network [2,3,4], SVM owns better generalization ability. Within a few years after its introduction SVM has played excellent performance on many real-world predictive data mining applications such as text categorization [5], time series prediction [6], pattern recognition [7] and image processing [8], etc.

Although SVM owns better generalization ability compared with many other machine learning methods, however, its computational complexity in training stage is too expensive, i.e., $O(n^3)$, where n is the total size of the training samples. To overcome this problem, so far, many improved algorithms for reducing the computational complexity of SVM have been presented, such as chunking algorithm [9], decomposition algorithm [10]

and sequential minimal optimization (SMO) [11], etc. On the other hand, many researchers have proposed some deformation algorithms based on the standard SVM. For example, in 2006, Mangasarian et al. [12] proposed a nonparallel plane classifier for binary data classification, named the generalized eigenvalue proximal support vector machine (GEPSVM). The essence of GEPSVM is to look for two nonparallel planes, so that data points of each class are proximal to one of them. GEPSVM has good learning speed because it solves two generalized eigenvalue problems of the order of input space dimension, but its classification accuracy is low. In 2007, Jayadeva et al. [13] proposed a new machine learning method called twin support vector machine (TWSVM) for the binary classification in the spirit of GEPSVM. TWSVM would generate two non-parallel planes, such that each plane is closer to one of the two classes and is as far as possible from the other. In TWSVM, a pair of smaller sized quadratic programming problems (QPPs) are solved, instead of solving single large one in SVM, makes the computational speed of TWSVM approximately 4 times faster than the traditional SVM. Because of its excellent performance, TWSVM has been

* Corresponding author e-mail: dingsf@cumt.edu.cn

applied to many areas such as speaker recognition [14], medical detection [15], etc.

Similar to SVM, TWSVM solves its QPPs in the dual space. However, this solving method will be affected by time and memory constraints when dealing with the large datasets, which would make the learning speed of TWSVM low. In order to address this problem in 2008, M. Arun Kumar et al. [16] used the sigmoid function to approach the objective function of TWSVM and then proposed smooth twin support vector machines (STWSVM). STWSVM directly solved QPPs in the original space instead of the dual space. Experimental results showed that STWSVM could make the classifier faster to compute in the classification phase than TWSVM. However, because of the low approximation ability of the sigmoid function, the classification accuracy of STWSVM was unsatisfactory. In order to further improve the classification performance of STWSVM, looking for a new smooth function with better approximation ability is the key problem.

In this paper, using the series expansion, a new class of polynomial smoothing is proposed. We have proved that the proposed smoothing functions have better smooth performance and their approximation accuracy can be as high as required. Subsequently, the polynomial functions are adopted to convert the original constrained quadratic programming problems of TWSVM into unconstrained minimization problems, and then are solved by the well-known Newton-Armijo algorithm. Based on the above idea, a novel version for smooth TWSVM, termed polynomial smooth twin support vector machines (PSTWSVM), is proposed in this paper. The experimental results show that as the smoothness rank of polynomial functions increases, the approximation accuracy and the classification performance are correspondingly improved. Therefore, the new class of polynomial functions provides better performance for smoothing the TWSVM.

The paper is organized as follows: In section 2, we propose the PSTWSVM model and prove its global convergence. Section 3 deals with experimental results, while section 4 is devoted to concluding remarks.

2 Polynomial Smooth Twin Support Vector Machines

2.1 Twin support vector machines

Consider a binary classification problem of classifying m_1 data points belonging to class +1 and m_2 data points belonging to class -1. Then let matrix A in $R^{m_1 \times n}$ represent the data points of class +1 while matrix B in $R^{m_2 \times n}$ represent the data points of class -1. Two nonparallel hyper-planes of the linear TWSVM can be expressed as follows.

$$x^T w_1 + b_1 = 0, \quad x^T w_2 + b_2 = 0 \quad (1)$$

The target of TWSVM is to generate the above two nonparallel hyper-planes in the n -dimensional real space R^n , such that each plane is closer to one of the two classes and is as far as possible from the other. A new sample point is assigned to class +1 or -1 depending upon its proximity to the two nonparallel hyper-planes. The linear classifiers are obtained by solving the following optimization problems.

$$\begin{aligned} \min_{w^{(1)}, b^{(1)}, \xi^{(2)}} & \frac{1}{2} \|Aw^{(1)} + e_1 b^{(1)}\|^2 + c_1 e_2^T \xi^{(2)} \\ \text{s.t.} & -(Bw^{(1)} + e_2 b^{(1)}) \geq e_2 - \xi^{(2)}, \\ & \xi^{(2)} \geq 0. \end{aligned} \quad (2)$$

$$\begin{aligned} \min_{w^{(2)}, b^{(2)}, \xi^{(1)}} & \frac{1}{2} \|Bw^{(2)} + e_2 b^{(2)}\|^2 + c_2 e_1^T \xi^{(1)} \\ \text{s.t.} & (Aw^{(1)} + e_1 b^{(2)}) \geq e_1 - \xi^{(1)}, \\ & \xi^{(1)} \geq 0. \end{aligned} \quad (3)$$

Where c_1 and c_2 are penalty parameters, $\xi^{(1)}$ and $\xi^{(2)}$ are slack vectors, e_1 and e_2 are the vectors of ones of appropriate dimensions.

In TWSVM, generally, we solve the QPPs in the dual space. However, this solving method will be affected by time and memory constraints when dealing with the big datasets. In order to improve the computational speed, the TWSVM model represented by (2) and (3) would be transformed into two unconstrained nonsmooth optimization problems by using the plus function.

According to the KKT theorem, we can get

$$\xi^{(2)} = \max\{0, e_2 + (Bw^{(1)} + e_2 b^{(1)})\} \quad (4)$$

$$\xi^{(1)} = \max\{0, e_1 - (Aw^{(2)} + e_1 b^{(2)})\} \quad (5)$$

The optimization problems (2) and (3) can be rewritten as

$$\begin{aligned} \min_{w^{(1)}, b^{(1)}, \xi^{(2)}} & \frac{1}{2} \|Aw^{(1)} + e_1 b^{(1)}\|^2 \\ & + c_1 e_2^T \max\{0, (e_2 + Bw^{(1)} + e_2 b^{(1)})\} \end{aligned} \quad (6)$$

$$\begin{aligned} \min_{w^{(2)}, b^{(2)}, \xi^{(1)}} & \frac{1}{2} \|Bw^{(2)} + e_2 b^{(2)}\|^2 \\ & + c_2 e_1^T \max\{0, (e_1 - Aw^{(2)} + e_1 b^{(2)})\} \end{aligned} \quad (7)$$

Let

$$(x_1)_+ = \max\{0, (e_2 + Bw^{(1)} + e_2 b^{(1)})\},$$

$$(x_2)_+ = \max\{0, (e_1 - Aw^{(2)} + e_1 b^{(2)})\},$$

where $(x_1)_+$ and $(x_2)_+$ are the plus functions. The objective functions of the unconstrained optimization problems (6) and (7) are convex and non-smooth which can be proved as follows.

Theorem 1 The unconstrained TWSVM model can be represented as (6) and (7) and the model is continuous but nonsmooth.

Proof Obviously, the differentiability and smoothness of (6) and (7) completely depends on the plus function x_+ .

Because of $x_+ = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$, we can calculate that $\lim_{x \rightarrow 0^+} = 1$ and $\lim_{x \rightarrow 0^-} = 0$. Therefore, x_+ is not differentiable in $x = 0$. Meanwhile, x_+ also can be expressed as $x_+ = \frac{|x|+x}{2}$, so x_+ is the continuous function. Therefore, x_+ is continuous but nonsmooth. End.

Theorem 1 shows that (6) and (7) are nonsmooth, so we can't use the gradient optimization method such as the Newton-Armijo method to solve (6) and (7). In order to address this problem, we will use the polynomial smooth function to approach (6) and (7).

2.2 The polynomial smooth function

Weierstrass Theorem [17] Set arbitrary continuous function $f(x)$, $x \in [m, n]$, existing polynomial $P_n(x)$, can make $\lim_{n \rightarrow \infty} \max_{m \leq x \leq n} |f(x) - P_n(x)| = 0$. The weierstrass theorem shows that any continuous real-valued function in closed interval can be arbitrarily approached by the polynomial function. From theorem 1 we can know that the plus function is a continuous function, so we can use the polynomial function to approach it. In this paper, we will give the common formula of the polynomial smooth function by transforming it to an equivalent infinite series.

Lemma 1 [18] Two expansion of $m = \frac{1}{2}$ can be expressed as

$$\begin{aligned} \sqrt{1+x} &= 1 + \frac{1}{2}x - \frac{1}{2 \cdot 4}x^2 + \frac{1 \cdot 3}{2 \cdot 4 \cdot 6}x^3 \\ &\quad - \frac{1 \cdot 3 \cdot 5}{2 \cdot 4 \cdot 6 \cdot 8}x^4 + \dots \\ &= 1 + \frac{1}{2}x - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (-x)^n, \end{aligned} \quad (8)$$

$-1 \leq x \leq 1$

Theorem 2 The plus function x_+ can be transformed to an equivalent infinite series in $[-\frac{1}{k}, \frac{1}{k}]$ as follows.

$$x_+ = \frac{1}{2k} \left(\frac{1+k^2x^2}{2} - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (1-k^2x^2)^n \right) + \frac{x}{2} \quad (9)$$

Proof According to the definition of x_+ , we can get

$$\begin{aligned} x_+ &= \max(0, x) \\ &= \frac{|x|+x}{2} \\ &= \frac{|kx|}{2k} + \frac{x}{2} \\ &= \frac{1}{2k} \sqrt{1+(k^2x^2)} - 1 + \frac{x}{2} \end{aligned} \quad (10)$$

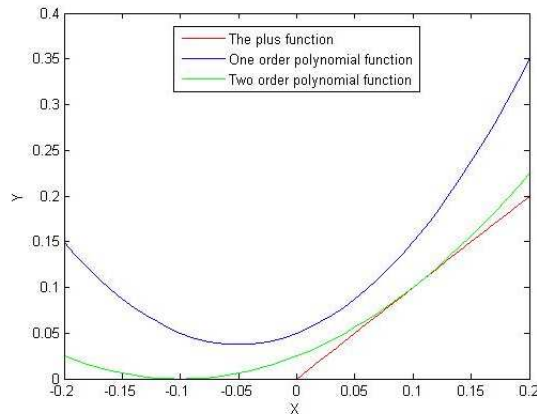


Figure 1 The approximation image of the plus function by the polynomial function

According to lemma 1 and (10), x_+ can be rewritten as

$$x_+ = \frac{1}{2k} \left(\frac{1+k^2x^2}{2} - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (1-k^2x^2)^n \right) + \frac{x}{2} \quad (11)$$

End.

Theorem 3 The polynomial approximation function for x_+ in $[-\frac{1}{k}, \frac{1}{k}]$ is

$$P_n(x, k) = \begin{cases} x, & x \geq \frac{1}{k} \\ \frac{1}{2k} \left(\frac{1+k^2x^2}{2} - \sum_{l=2}^n \frac{(2l-3)!!}{(2l)!!} (1-k^2x^2)^l \right) + \frac{x}{2}, & |x| < \frac{1}{k}, k > 0 \\ 0, & x \leq -\frac{1}{k} \end{cases} \quad (12)$$

where n is a positive integer. The approximation image of the plus function by the polynomial function when $k = 10, n = 1, 2$ is shown as figure 1. From Figure 1, we can see that the approximation accuracy of $P_n(x, k)$ will be higher with n larger.

Theorem 4 $P_n(x, k)$ is defined as (12), it has some characteristics as follows.

- (1) $P_n(x, k)$ has n -order smoothness about x .
- (2) $\lim_{n \rightarrow \infty} \max(P_n(x, k) - x_+) = 0$.

Proof (1) If $P_n(x, k)$ has n -order smoothness about x , it must meet the following conditions.

$$P_n\left(\frac{1}{k}, k\right) = \frac{1}{k}, \quad P_n\left(-\frac{1}{k}, k\right) = 0$$

$$\nabla P_n\left(\frac{1}{k}, k\right) = 1, \quad \nabla P_n\left(-\frac{1}{k}, k\right) = 0$$

$$\nabla^n P_n\left(\frac{1}{k}, k\right) = 0, \quad \nabla^n P_n\left(-\frac{1}{k}, k\right) = 0, n \geq 2$$

According to (12), it can be got

$$P_n(\frac{1}{k}, k) = \frac{1}{k}, \quad P_n(-\frac{1}{k}, k) = 0$$

We find the partial derivative of x , it can be got
When $n \geq 1$,

$$\nabla P_n(x, k) = \begin{cases} 1, & x \geq \frac{1}{k} \\ \frac{kx}{2} (1 + \sum_{l=2}^n \frac{(2l-3)!!}{(2l-2)!!} (1 - k^2 x^2)^{l-1}) + \frac{1}{2}, & |x| < \frac{1}{k}, k > 0 \\ 0, & x \leq -\frac{1}{k} \end{cases}$$

When $n \geq 2$,

$$\nabla^2 P_n(x, k) = \begin{cases} 0, & x \geq \frac{1}{k} \\ \frac{k}{2} (1 + \sum_{l=2}^n \frac{(2l-3)!!}{(2l-2)!!} (1 - k^2 x^2)^{l-1}) & |x| < \frac{1}{k}, k > 0 \\ -\frac{k^3 x^2}{2} \sum_{l=2}^n \frac{(2l-3)!!}{(2l-4)!!} (1 - k^2 x^2)^{l-2}, & |x| < \frac{1}{k}, k > 0 \\ 0, & x \leq -\frac{1}{k} \end{cases}$$

Obviously, $\nabla P_n(x, k)$, $\nabla^2 P_n(x, k)$ and $\nabla^n P_n(x, k)$ ($n > 2$) are existence and continuation in $x = \pm \frac{1}{k}$. So $P_n(x, k)$ has n -order smoothness about x .

(2) According to Weierstrass Theorem, it can be got easily

$$\lim_{n \rightarrow \infty} \max(P_n(x, k) - x_+) = 0.$$

End.

Theorem 4 shows that the polynomial smooth function transformed to an equivalent infinite series can achieve arbitrary precision to approach the plus function when n is large enough.

2.3 The optimal smoothing factor

There is a parameter k called smoothing factor in (12). We give the formula of optimal smoothing factor as follows.

Theorem 5 Give arbitrary precision E , if the smooth function $P_n(x, k)$ meets the condition $|P_n(x, k) - x| \leq E$ when it approaches to x_+ , the smoothing factor k is called the optimal smoothing factor and is denoted as $k_{opt}(n, E)$.

Because the error of $P_n(x, k)$ approaching to x_+ is maximum in $x = 0$, we can get $k_{opt}(n, E)$ when it meets the condition $P_n(x, k) - x_+ \leq E$ in $x = 0$.

Therefore, if $x = 0$, calculate (12), we can get

$$k_{opt}(n, E) \geq \frac{\frac{1}{2} - \sum_{l=2}^n \frac{(2l-3)!!}{(2l)!!}}{2E} \tag{13}$$

2.4 PSTWSVM algorithm

Because $P_n(x, k)$ has n -order smoothness when $n \geq 2$, Newton-Armijo optimization algorithm can be used to

solve the following unconstrained optimization problems.

$$\min_{w^{(1)}, b^{(1)}, \xi^{(2)}} \frac{1}{2} \|Aw^{(1)} + e_1 b^{(1)}\|^2 + c_1 e_2^T P((e_2 + Bw^{(1)} + e_2 b^{(1)}), k) \tag{14}$$

$$\min_{w^{(2)}, b^{(2)}, \xi^{(1)}} \frac{1}{2} \|Bw^{(2)} + e_2 b^{(2)}\|^2 + c_2 e_1^T P((e_1 + Aw^{(2)} + e_1 b^{(2)}), k) \tag{15}$$

Algorithm 1 PSTWSVM based on the Newton-Armijo method

Input: Give the initial value $(w^0, b^0) \in R^{n+1}$, η , let the iteration number $i = 0$, the order of polynomial function n , the arbitrary precision E

Output: The optimal value of the objective function

Step1: calculate $P_n(x, k)$ and $g^i = \nabla P_n(x, k)$.

Step2: If $\|g^i\| \leq \eta$ select $(w^*, b^*) = (w^i, b^i)$, then terminate programs. Otherwise according to $\nabla^2 P_n(x, k) d^i = -g^i$, calculate the down direction d^i .

Step3: (Armijo method) take $\delta \in (0, \frac{1}{2})$,

$$\lambda_i = \max\{1, \frac{1}{2}, \frac{1}{4}, \dots\}$$

let $P_n(x, k) - P_n((w^i, b^i) + \lambda_i d^i, k) \geq -\delta \lambda_i g^i d^i$, then

let $(w^{i+1}, b^{i+1}) = (w^i, b^i) + \lambda_i d^i$

Step4: Let $i \leftarrow i + 1$ turn to Step2.

2.5 The nonlinear PSTWSVM

If the previous conclusions are extended to nonlinear smooth PSTWSVM, it can be used to deal with the nonlinear problem.

In order to obtain the nonlinear classifiers we consider the following kernel generated surfaces

$$K(x^T, C^T)u_1 + b_1 = 0, \quad K(x^T, C^T)u_2 + b_2 = 0, \tag{16}$$

where $C^T = [A \ B]^T$, $(u_{(i)}, b_{(i)}) \in (R^m \times R)$ ($i = 1, 2$) and K is an chosen kernel. The nonlinear TWSVM are obtained by solving the following optimization problems.

$$\min_{w^{(1)}, b^{(1)}, \xi^{(2)}} \frac{1}{2} \|K(A, C^T)w^{(1)} + e_1 b^{(1)}\|^2 + c_1 e_2^T \xi^{(2)}$$

$$s.t. \quad -(K(B, C^T)w^{(1)} + e_2 b^{(1)}) \geq e_2 - \xi^{(2)}, \tag{17}$$

$$\xi^{(2)} \geq 0.$$

$$\min_{w^{(1)}, b^{(1)}, \xi^{(2)}} \frac{1}{2} \|K(B, C^T)w^{(2)} + e_2 b^{(1)}\|^2 + c_2 e_1^T \xi^{(1)}$$

$$s.t. \quad (K(A, C^T)w^{(2)} + e_1 b^{(2)}) \geq e_1 - \xi^{(1)}, \tag{18}$$

$$\xi^{(1)} \geq 0.$$

Introducing the plus function, (16) and (17) can be transformed into the following optimization problems without constraint.

$$\min_{w^{(1)}, b^{(1)}, \xi^{(2)}} \frac{1}{2} \left\| K(A, C^T)w^{(1)} + e_1 b^{(1)} \right\|^2 + c_1 e_2^T (e_2 + K(B, C^T)w^{(1)} + e_2 b^{(1)})_+ \quad (19)$$

$$\min_{w^{(2)}, b^{(2)}, \xi^{(1)}} \frac{1}{2} \left\| K(B, C^T)w^{(2)} + e_2 b^{(2)} \right\|^2 + c_2 e_1^T (e_1 - K(A, C^T)w^{(2)} + e_1 b^{(2)})_+ \quad (20)$$

We can get the nonlinear PSTWSVMs-NA model using the polynomial smooth function.

$$\min_{w^{(1)}, b^{(1)}, \xi^{(2)}} \frac{1}{2} \left\| K(A, C^T)w^{(1)} + e_1 b^{(1)} \right\|^2 + c_1 e_2^T P((e_2 + K(B, C^T)w^{(1)} + e_2 b^{(1)}), k) \quad (21)$$

$$\min_{w^{(2)}, b^{(2)}, \xi^{(1)}} \frac{1}{2} \left\| K(B, C^T)w^{(2)} + e_2 b^{(2)} \right\|^2 + c_2 e_1^T P((e_1 - K(A, C^T)w^{(2)} + e_1 b^{(2)}), k) \quad (22)$$

The previous conclusions and theorems are also applicable to the nonlinear PSTWSVM model.

3 Numerical experiments and analysis

In this section, in order to show the performance of PSTWSVM, we conduct experiments on one synthetic dataset and ten benchmark datasets using four algorithms, that is, GEPSVM, TWSVM, STWSVM and PSTWSVM. Furthermore, in order to test the ability of these algorithms to deal with the large samples, we make experiment on NDC dataset. The environments of all algorithms are implemented in Intel (R) Core (TM) 2Duo CUP E4500, 2G memory and MATLAB 7.11.0. The dual QPPs arising in TWSVM are solved using mosek optimization toolbox for MATLAB [19] which implements fast interior point based algorithms. Classification accuracy of each algorithm is measured by standard tenfold cross-validation methodology.

3.1 Experiment on synthetic dataset

This synthetic dataset is 2-dimensional and it is mainly used to intuitively test the classification performance of linear PSTWSVM and nonlinear PSTWSVM. The performance of these algorithms depends heavily on the choices of parameters. In this experiment, the optimal parameters of these algorithms are searched from set $\{2^i | i = -6, -4, -2, 0, 1, 2, 4, 6\}$. In PSTWSVM, the parameter of Newton-Armijo method is set $\varepsilon_1 = 1.0E - 3$,

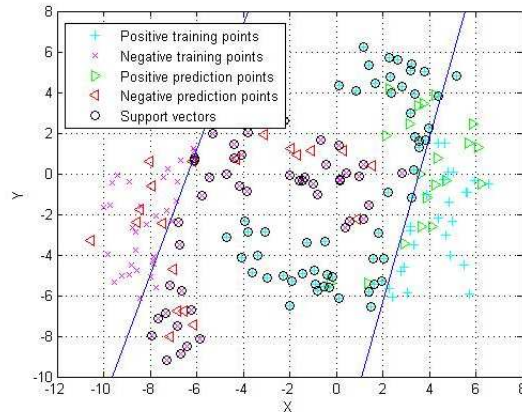


Figure 2 The classification result of linear PSTWSVM

the approximation accuracy of smooth function is set $\varepsilon_2 = 1.0E - 3$. For the nonlinear case, we only consider the Gaussian kernel function. The optimal value of Gaussian kernel parameter is selected over the range $\{2^i | i = -6, -4, -2, 0, 1, 2, 4, 6\}$. The order of polynomial is set $n = 5$.

Table 1 shows the comparison of classification accuracy and CPU time for PSTWSVM with GEPSVM, TWSVM and STWSVM for linear kernel on the synthetic dataset. Table 2 shows the comparison of classification performance for nonlinear extensions of PSTWSVM with GEPSVM, TWSVM and STWSVM. Figure 2 and figure 3 are the classification map of linear PSTWSVM and nonlinear PSTWSVM.

Table 1 Comparison for linear kernel

Algorithms	Train accuracy (%)	Test accuracy (%)	Time (s)
GEPSVM	85.5±6.35	82.7±1.49	0.0679
TWSVM	87.8±3.21	83.0±5.21	0.1141
STWSVM	86.7±5.47	81.6±6.54	0.0910
PSTWSVM	87.3±0.21	84.5±1.25	0.0772

Table 2 Comparison for Gaussian kernel

Algorithms	Train accuracy (%)	Test accuracy (%)	Time (s)
GEPSVM	91.7±3.26	86.2±2.12	0.1954
TWSVM	94.2±3.07	87.1±2.25	0.1278
STWSVM	92.8±2.23	86.5±1.58	0.1023
PSTWSVM	94.3±2.21	87.9±1.29	0.0954

Figure 2. The classification result of linear PSTWSVM
Figure 3. The classification result of nonlinear PSTWSVM

Table 1 and Table 2 reveal that the accuracy of PSTWSVM is significantly better than STWSVM also. It

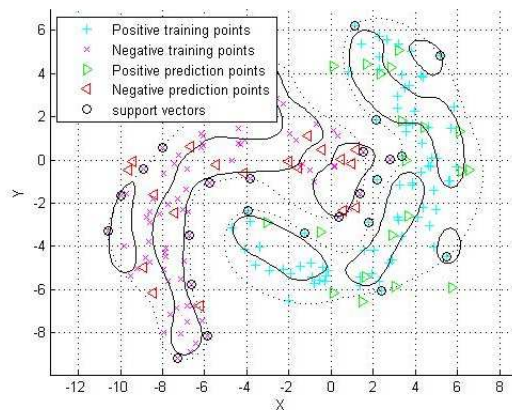


Figure 3 The classification result of nonlinear PSTWSVM

can also be observed that the accuracy of PSTWSVM is almost same as that of TWSVM. Compared with linear PSTWSVM, the classification accuracy of nonlinear PSTWSVM is improved, but its training time has increased. This is because that it needs to calculate the nuclear matrix computation time for nonlinear PSTWSVM. From Figure 2 and 3, we can intuitively see that this synthetic dataset is nonlinear, so the classification performance of nonlinear PSTWSVM is better than the linear PSTWSVM.

3.2 Experiment on UCI machine learning datasets

In order to further demonstrate the performance of our algorithm, we conduct experiments on ten benchmark datasets from the UCI. Table 3 shows the comparison of classification accuracy for PSTWSVM with GEPSVM, TWSVM and STWSVM for linear kernel on 10 UCI datasets. Table 4 shows the comparison of classification performance for nonlinear extensions of PSTWSVM with GEPSVM, TWSVM and STWSVM. The parameters are selected the same as 3.1.

Table 3 and 4 show that the accuracy performance of PSTWSVM is better than STWSVM. It can also be observed that the accuracy of PSTWSVM is almost same as that of TWSVM. In fact, accuracy of PSTWSVM is slightly better than TWSVM on some datasets.

3.3 Experiment on NDC datasets

In order to test the ability of our algorithm for dealing with the large datasets, we conducted experiment on NDC datasets which are generated by David Musicant' NDC Data Generator [20]. For experiment on NDC datasets, we fix penalty parameters of all algorithms to be one (i.e.

Table 3 Comparison for linear kernel

Dataset	PSTWSVM	STWSVM	TWSVM	GEPSVM
Hepatitis (155×19)	78.05±4.31	77.39±2.15	78.08±2.16	77.28±2.78
Housing (506×13)	86.21±2.39	84.42±3.87	85.42±4.53	74.81±2.85
Wdbc (596×30)	96.10±6.32	94.89±4.31	96.22±6.67	92.81±2.54
Glass6 (214×10)	96.52±4.56	95.70±6.05	96.55±2.40	96.21±2.72
Votes (435×16)	95.50±1.23	94.96±4.24	95.85±2.24	95.63±2.74
Pima (768×8)	75.95±2.31	74.52±3.37	75.40±9.68	74.48±4.67
Spect (267×22)	80.85±8.36	80.50±9.85	81.98±2.17	55.98±2.17

Table 4 Comparison for Gaussian kernel

Dataset	PSTWSVM	STWSVM	TWSVM	GEPSVM
Wdbc (194×32)	77.01±6.56	75.89±7.68	77.27±8.62	75.81±6.35
Ionosphere (351×34)	94.01±1.76	92.84±2.09	93.68±2.74	86.62±6.78
Heart-statlog (270×14)	78.23±8.67	74.89±9.17	79.26±6.34	63.35±12.56
Hepatitis (155×19)	78.88±10.16	78.85±8.99	78.81±9.18	78.54±10.17

$C = C_1 = C_2 = 1$). We use Gaussian kernel with $\sigma=2^{-17}$. Other parameters are set as same as the section 3.1. Table 5 gives a description of NDC datasets. Table 6 shows the comparison of computing time and accuracy for three algorithms with linear kernel. On the other hand, table 7 shows the comparison of classification performance for these algorithms with Gaussian kernel.

Table 5 Description of NDC datasets

Dataset	# Training data	# Test data	# Feature
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
NDC-10k	10,000	1000	32
NDC-1l	100,000	10,000	32
NDC-3l	300,000	30,000	32
NDC-5l	500,000	50,000	32
NDC-1m	1,000,000	100,000	32

Table 6 Comparison for linear kernel

Dataset	PSTWSVM	STWSVM	TWSVM
	Train (%)	Train (%)	Train (%)
	Test (%)	Test (%)	Test (%)
	Time (s)	Time (s)	Time (s)
NDC-3k	80.05	79.85	79.93
	77.64	77.51	77.66
	0.0469	0.0545	27.08
NDC-4k	79.82	79.69	79.80
	73.77	73.75	73.75
	0.0605	0.0663	60.94
NDC-5k	79.12	78.95	79.15
	80.22	80.20	80.23
	0.0734	0.0761	114.24
NDC-10k	86.58	86.42	86.45
	87.29	87.24	87.38
	0.1178	0.1239	1092.07
NDC-11	87.59	86.35	-
	86.78	86.24	-
	0.996	1.014	-
NDC-31	81.07	78.79	-
	78.89	78.75	-
	2.899	3.103	-
NDC-51	79.36	78.85	-
	80.08	79.12	-
	5.1312	5.3505	-

“-” We stop experiment as computing time was very high

Table 7 Comparison for Gaussian kernel

Dataset	PSTWSVM	STWSVM	TWSVM
	Train (%)	Train (%)	Train (%)
	Test (%)	Test (%)	Test (%)
	Time (s)	Time (s)	Time (s)
NDC-500	100.00	100.00	99.25
	80.18	80.15	80.18
	0.5621	0.5687	0.789
NDC-700	99.28	99.25	99.27
	84.25	83.14	84.29
	1.3119	1.3119	1.7322
NDC-900	99.56	99.56	99.56
	81.89	81.34	80.58
	2.5014	2.5328	3.4675
NDC-1k	98.83	98.47	98.85
	85.32	84.12	83.85
	3.5012	3.5198	4.1176
NDC-2k	100.00	100.00	99.68
	88.26	88.24	88.27
	21.07	21.23	25.8958
NDC-3k	100.00	100.00	99.53
	90.38	90.12	90.45
	66.298	66.604	85.445

From table 6, we can see that in view of the high computing time, TWSVM can't work when the training samples reach 100000. However, PSTWSM and STWSVM can get reasonable accuracy in the relatively short time when the training samples reach 500000, which indicates that PSTWSVM and STWSVM have the

advantage on dealing with large dataset comparing with TWSVM. Furthermore, from table 6 and 7, we also see that the classification accuracy of PSTWSVM and the computing time are better than STWSVM. The accuracy of PSTWSVM is almost same as that of TWSVM. In fact, accuracy of PSTWSVM is slightly better than TWSVM on some datasets. Therefore, PSTWSVM is suitable for dealing with the large dataset.

4 Conclusion

In order to improve the performance of STWSVM, seeking a better smoothing function is the key problem. In this paper, a novel version for smooth TWSVM, called polynomial smooth twin support vector machines (PSTWSVM), is proposed. Firstly, using the series expansion, a new class of polynomial smoothing is proposed, and then we prove their important properties. Subsequently, the polynomial functions are adopted to convert the original constrained QPPs of TWSVM into unconstrained minimization problems, and then are solved by the well-known Newton-Armijo algorithm. The experiments on synthetic and real-word benchmark datasets show that the proposed algorithm can obtain better classification than STWSVM. At last, we enhance our algorithm to deal with large datasets, the results indicate that PSTWSVM is a good method to deal with large datasets.

Acknowledgements

This work is supported by the National Key Basic Research Program of China (No.2013CB329502), the National Natural Science Foundation of China (Nos.61379101,51104157), the Natural Science Foundation of Jiangsu Province (No.BK20130209), and the Excellent Youth Scientist Promotive foundation of Shandong Province (No.BS2010NJ014).

References

- [1] S. F. Ding and B. J. Qi, Research of granular support vector machine, *Artificial Intelligence Review*, **38**, 1-7 (2012).
- [2] Lippmann, R.P., An introduction to computing with neural nets, *ASSP Magazine, IEEE*, **4**, 4-22 (1987).
- [3] X. J. Jing and L. Cheng, An Optimal PID Control Algorithm for Training Feedforward Neural Networks, *IEEE Transactions on Industrial Electronics*, **60**, 2273-2283 (2013).
- [4] A. Mellit, H. Eleuch, M. Benhanem, C. Elaoun and A. Massi Pavan *Energy Conversion and Management*, **51**, 771-782 (2010).
- [5] D. Morariu, R. Cretulescu and L. Vintan, Improving a SVM Meta-classifier for Text Documents by using Naïve Bayes, *International Journal of Computers Communications & Control*, **5**, 351-361 (2010).

- [6] Z. Y. Chen and Z. P. Zhi. Distributed customer behavior prediction using multiplex data: A collaborative MK-SVM approach, *Knowledge-Based Systems*, **35**, 111-119 (2012).
- [7] Haralick, R.M. ; Shanmugam, K. ; Dinstein, Its'Hak, Textural Features for Image Classification, *Systems, Man and Cybernetics, IEEE Transactions on*, **SMC-3**, 610-621 (1973)
- [8] C. S. Lo and C. M. Wang, Support vector machine for breast MR image classification. *Computers & Mathematics with Applications*, **64**, 1153-1162 (2012).
- [9] C. Cortes and V. N. Vapnik, Support vector networks. *Machine Learning*, **20**, 273-297 (1995).
- [10] E. Osuna and R. F. Girosi, An improved training algorithm for support vector machines. *Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing*. New York: IEEE Press, 276-285 (1997).
- [11] J. C. Platt, Using analytic QP and sparseness to speed training of support vector machines. In M. Kearns, S.olla and D.Cohn, *Advances in Neural Information Processing Systems 11*. Cambridge, MA: MIT Press, 557-563 (1999).
- [12] O. L. Mangasarian, Multisurface proximal support vector machine classification via generalized eigenvalues, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **28**, 69-74 (2006).
- [13] K. R. Jayadeva and C. D. Suresh. Twin support vector machines for pattern classification, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **29**, 905-910 (2007).
- [14] H. H. Cong, C. F. Yang and X. R. Pu, Efficient Speaker Recognition based on Multi-class Twin Support Vector Machines and GMMs. *2008 IEEE Conference on Robotics, Automation and Mechatronics*, 348-352 (2008).
- [15] X. S. Zhang and X. B. Gao and Y. Wang, Twin Support Vector Machine For MCs Detection. *Journal of Electronics (China)*, **26**, 318-325 (2009).
- [16] M. Arun Kumar and M. Gopal, Application of smoothing technique on twin support vector machines, *Pattern Recognition Letters*, **28**, 1842-1848 (2008).
- [17] Y. Q. Liu, S. Y. Liu and M. T. Gu, Self-training Polynomial Support Smooth Semi-supervised Support Vector Machines. *Journal of System Simulation*, **21**, 5740-5743 (2009).
- [18] R. Bin and L. L. Cheng. Polynomial smoothing support vector regression. *Control Theory & Applications*, **28**, 261-265 (2011).
- [19] Mosek, <http://www.mosek.com>, (2007).
- [20] O. L. Mangasarian and D. R. Musicant, Lagrangian support vector machines, *J. Machine Learn. Res.*, **1**, 161-177 (2001).



Shifei Ding

received his B.Sc., and M.Sc. degree in mathematics and computer science in 1987, 1998 respectively from Qufu Normal University, received Ph.D. degree in computer science from Shandong University of Science and Technology in 2004, and received post Ph.D. degree in computer science from Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, and Chinese Academy of Sciences in 2006. He is a professor and Ph.D supervisor at China University of Mining and Technology. His research interests include intelligent information processing, pattern recognition, machine learning, data mining, and granular computing et al. He has published 3 books, and more than 150 research papers in journals and international conferences.



Huajuan Huang

received her B.Sc. degree and M.Sc. degree in applied computer Technology from Guangxi University for Nationalities, Guangxi, China, in 2006 and 2009 respectively. Since 2011, she has been a Ph.D. degree candidate in applied computer Technology from the China University of Mining and Technology, Xuzhou, China. Her current research interests include data mining, machine learning and computational intelligence.



Xinzheng Xu

received the B.S. degree from College of Information and Electrical Engineering, Shandong University of Science and Technology, Qingdao, and M.S. degree from Computer Science Department of Xiamen University, Xiamen, China in 2002 and 2005, respectively. And, he received Ph.D. degree from School of Computer Science and Technology, China University of Mining and Technology under the supervision of Prof. Shifei Ding. His research interests include intelligent information processing, neural networks, machine learning, et al.



Jian Wang works in Shandong University of Science and Technology, associate professor. She received B.S. degree from surveying department of Shandong Institute of Mining and Technology, Taian, M.S. degree from Geoinformation Science & Engineering

College, Shandong University of Science and Technology, and Ph.D. of Geomatics College, Shandong University of Science and Technology, China, in 1997, 2002 and 2006. Her research interests include modern data-processing theory, deformation monitoring and LiDAR technology.