Polynomial Smooth Twin Support Vector Machines Based on Invasive Weed Optimization Algorithm

Shifei Ding\textsuperscript{1,2}, Huajuan Huang\textsuperscript{1,2}, Junzhao Yu\textsuperscript{1,2}, Fu\textsuperscript{lin} Wu\textsuperscript{1,2}

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China, 221116
Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, 100190
Email: dingsf@cumt.edu.cn

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 used 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. Meanwhile, in order to find the suitable parameters of PSTWSVM, Invasive Weed Optimization (IWO) algorithm is used to optimize the proposed algorithm. Then we propose an algorithm called polynomial smooth twin support vector machines based on invasive weed optimization algorithm (PSTWSVM-IWO). Finally, the effectiveness of the proposed method is demonstrated via experiments on synthetic and UCI benchmark datasets.

Index Terms—Polynomial function, Newton-Armijo, Invasive weed optimization algorithm, Parameter optimization, Twin support vector machines

I. INTRODUCTION

Support vector machine (SVM) proposed 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], SVM owns better generalization ability.

Within a few years after its introduction SVM has played excellent performance in many real-world predictive data mining applications such as text categorization [4], time series prediction [5], pattern recognition [6] and image processing [7], etc.

Although SVM owns better generalization ability compared with many other machine learning methods, however, its computational complexity in training stage is too expensive. To address this problem, so far, many improved algorithms have been presented, such as chunking algorithm [8], decomposition algorithm [9] and sequential minimal optimization (SMO) [10], etc. However, these algorithms are too complex. On the other hand, many researchers have proposed some deformation algorithms based on standard SVM. For example, in 2006, Mangasarian et al. [11] proposed a nonparallel plane classifier for binary data classification, named 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, but its classification accuracy is low. In 2007, Jayadeva et al. [12] 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 a 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 applied to many areas such as speaker recognition [13], medical detection [14], 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. [15] used the sigmoid function to approach the objective function of TWSVM and then proposed smooth twin support vector machines

This work is supported by the National Natural Science Foundation of China (Nos.61373910), the National Key Basic Research Program of China (No. 2013CB329502), and the Natural Science Foundation of Jiangsu Province (No.BK20130290).
Corresponding author, Shifei Ding, E-mail: dingsf@cumt.edu.cn.

© 2014 ACADEMY PUBLISHER
doi:10.4304/jcp.9.5.1226-1233
(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 used 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. Besides, in order to overcome PSTWSVM parameters selection problem, we use Invasive Weed Optimization (IWO) algorithm [16] which has fast global searching ability to select PSTWSVM parameters, so that we would obtain the optimal parameters combination. Finally, the experimental results show the effectiveness and stability of the proposed method.

The paper is organized as follows: In section 2, we propose the PSTWSVM model and prove its global convergence. In section 3, PSTWSVM-IWO is detailed introduced and analyzed. Computational comparisons on synthetic and UCI datasets are done in section 4 and section 5 gives concluding remarks.

II. POLYNOMIAL SMOOTH TWIN SUPPORT VECTOR MACHINES

A. 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 \( \mathbb{R}^{m_1 \times n} \) represent the data points of class +1 while matrix \( B \) in \( \mathbb{R}^{m_2 \times n} \) represent the data points of class -1. Two nonparallel hyper-planes of the linear TSVMs can be expressed as follows.

\[
x^T w_1 + b_1 = 0 \quad \text{and} \quad x^T w_2 + b_2 = 0 \tag{1}
\]

The target of TSVMs is to generate the above two nonparallel hyper-planes in the \( n \)-dimensional real space \( \mathbb{R}^n \), 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.

\[
\min_{w_1^{(i)}, b_1^{(i)}, w_2^{(i)}, b_2^{(i)}, \xi} \frac{1}{2} \|A w_1^{(i)} + b_1^{(i)}\|^2 + c_1 \xi_1 + \xi_2

\text{s.t.} \quad -(B w_1^{(i)} + b_1^{(i)}) \geq \xi_2 - \xi_1 + \xi_2^{(2)}, \tag{2}
\]

where \( \xi_1 \) and \( \xi_2 \) are slack vectors, \( c_1 \) and \( c_2 \) are penalty parameters, \( \xi_1 \) and \( \xi_2 \) are slack vectors, \( c_1 \) and \( c_2 \) are 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 non-smooth optimization problems by using the plus function.

According to the KKT theorem, we can get

\[
\xi_1^{(2)} = \max\{0, c_1 + (B w_1^{(i)} + b_1^{(i)})\} \tag{4}
\]

\[
\xi_2^{(1)} = \max\{0, c_2 - (A w_1^{(i)} + b_1^{(i)})\} \tag{5}
\]

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

\[
\min_{w_1^{(i)}, b_1^{(i)}, \xi_1^{(i)}, \xi_2^{(i)}} \frac{1}{2} \|A w_1^{(i)} + b_1^{(i)}\|^2 + c_1 \xi_1^{(i)} + \xi_2^{(i)} \max\{0, (c_1 + (B w_1^{(i)} + b_1^{(i)})\} \tag{6}
\]

\[
\min_{w_1^{(i)}, b_1^{(i)}, \xi_1^{(i)}, \xi_2^{(i)}} \frac{1}{2} \|B w_1^{(i)} + b_1^{(i)}\|^2 + c_2 \xi_2^{(i)} \max\{0, (c_2 - (A w_1^{(i)} + b_1^{(i)})\} \tag{7}
\]

Let \( (x_1) = \max\{0, (c_1 + (B w_1^{(i)} + b_1^{(i)})\} \),

\( (x_2) = \max\{0, (c_2 - (A w_1^{(i)} + b_1^{(i)})\} \),

where \( (x_1) \) and \( (x_2) \) are the plus functions. Apparently, the objective functions of the unconstrained optimization problems (6) and (7) are convex and non-smooth.

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

**Theorem 1** shows that (6) and (7) are non-smooth, 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).

B. The Polynomial Smooth Function

**Weierstrass Theorem** [17] Set arbitrary continuous function \( f(x), x \in [m, n] \), existing polynomial \( P_m(x) \) makes

\[
\lim_{m,n \to +\infty} \max_{x \in [m,n]} |f(x) - P_m(x)| = 0.
\]

Weierstrass’s 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
\[
\sqrt{1+x} = 1 + \frac{1}{2} x - \frac{1}{2 \cdot 4} x^2 + \frac{1}{2 \cdot 4 \cdot 6} x^3 - \frac{1}{2 \cdot 4 \cdot 6 \cdot 8} x^4 + \cdots = \\
1 + \frac{1}{2} x - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (-x)^n, \quad -1 \leq x \leq 1
\]  
(8)

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

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

**Proof** According to the definition of \( x_+ \), we can get

\[
x_+ = \max(0,x) = \frac{|x|}{2} = \frac{|x|}{2} = \frac{1}{2k} \sqrt{1 + (k^2 x^2) - 1 + \frac{x}{2}}
\]  
(10)

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

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

End.

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

\[
P_n(x,k) = \begin{cases} 
  x, & x \geq \frac{1}{k} \\
  \frac{1}{2k} + \frac{k^2 x^2}{2} - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (1-k^2 x^2)^n \frac{x}{2}, & 0 < \frac{1}{k} < x < \frac{1}{k} \end{cases}
\]  
(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.

![Figure 1. The approximation image of the plus function](image)

**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. \( \lim_{x \to \pm \frac{1}{k}} P_n(x,k) - x_+ = 0 \).

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

\[
\nabla P_n(x,k) = \begin{cases} 
  1, & x \geq \frac{1}{k} \\
  \frac{kx}{2} + \frac{k^2 x^2}{2} - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (1-k^2 x^2)^n - \frac{x}{2}, & 0 < \frac{1}{k} < x < \frac{1}{k} \end{cases}
\]

(13)

\[
\nabla^2 P_n(x,k) = \begin{cases} 
  0, & x \geq \frac{1}{k} \\
  \frac{k^3 x}{2} + \frac{k^2 x^2}{2} - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (1-k^2 x^2)^n - \frac{x}{2}, & 0 < \frac{1}{k} < x < \frac{1}{k} \end{cases}
\]

(14)

\[
\nabla^n P_n(x,k) = \begin{cases} 
  0, & x \geq \frac{1}{k} \\
  \frac{k^n x}{2} + \frac{k^{n-1} x^2}{2} - \sum_{n=2}^{\infty} \frac{(2n-3)!!}{(2n)!!} (1-k^2 x^2)^n - \frac{x}{2}, & 0 < \frac{1}{k} < x < \frac{1}{k} \end{cases}
\]

(15)

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.

**C. 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 \( P_{\epsilon}(x,k) \) approaching to \( x \), is maximum in \( x = 0 \), we can get \( k_{\text{opt}}(n,E) \) when it meets the condition \( P_{\epsilon}(x,k) = x \leq E \) in \( x = 0 \).

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

\[
k_{\text{opt}}(n,E) \geq \frac{1}{2} \sum_{j=1}^{j=n} (2j-3)!! \frac{2^{j}!!}{E^{j}}
\]

(13)

D. PSTWSVM Algorithm

Because \( P_{\epsilon}(x,k) \) has \( n \)-order smoothness when \( n \geq 2 \), Newton-Armijo optimization algorithm can be used to solve the following unconstrained optimization problems.

\[
\begin{align*}
\min_{i=1}^{\frac{1}{2}} & \|w^{i}(0) + e_{j}b^{i}(0)\|^{2} + c_{i}e_{i}P((e_{i} + Bu^{i}) + e_{j}b^{i}(0)),k) \quad (14) \\
\min_{i=1}^{\frac{1}{2}} & \|Bw^{i}(0) + e_{j}b^{i}(0)\|^{2} + c_{i}e_{i}P((e_{i} + Au^{i}) + e_{j}b^{i}(0),k) \quad (15)
\end{align*}
\]

Algorithm 1 PSTWSVM based on the Newton-Armijo method

Input: Give the initial value \((w^{0}, b^{0}) \in R^{r+i}, \eta \), 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 \( P_{\epsilon}(x,k) \) and \( g^{'} = V_{n}(x,k) \).

Step2: If \( \|g^{'}\| \leq \eta \), select \((w^{i'}, b^{i'}) = (w^{i}, b^{i})\), then terminate programs. Otherwise according to \( V^{2}P_{\epsilon}(x,k)d^{i} = -g^{'} \), calculate the down direction \( d^{i} \).

Step3: take \( \delta \in (0, \frac{1}{2}) \), \( \lambda_{i} = \max\{1, \frac{1}{2}, \frac{1}{4}, \cdots\} \), let \( P_{\epsilon}(x,k) = P_{\epsilon}((w^{i'}, b^{i'}) + \lambda d^{i}, k) \geq -\delta \lambda d^{i} \), then let \((w^{i+1}, b^{i+1}) = (w^{i'}, b^{i'}) + \lambda d^{i} \).

Step4: Let \( i = i+1 \), turn to Step2.

E. 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

\[
K(x^{T}, C^{T})u_{i} + b_{i} = 0 , K(x^{T}, C^{T})u_{i} + b_{i} = 0 , \quad (16)
\]

where \( C^{T} = \begin{bmatrix} A & B \end{bmatrix} \), \((u_{i}, b_{i}) \in (R^{n} \times R) \quad (i = 1, 2)\)

and \( K \) is an chosen kernel. The nonlinear TWSVM are obtained by solving the following optimization problems.

\[
\begin{align*}
\min_{w^{(2)}, b^{(2)}, \xi^{(2)}} & \frac{1}{2} |K(A,C^{T})w^{(0)} + e_{j}b^{i}(0)|^{2} + c_{i}e_{i}^{T} \xi^{(2)} \\
\text{s.t.} \quad & -K(B,C^{T})w^{(0)} + e_{j}b^{i}(0) \geq c_{i} - \xi^{(2)}, \xi^{(2)} \geq 0. \quad (17)
\end{align*}
\]

Introducing the plus function, (17) and (18) can be transformed into the following optimization problems without constraint.

\[
\begin{align*}
\min_{w^{(2)}, b^{(2)}, \xi^{(2)}} & \frac{1}{2} |K(A,C^{T})w^{(0)} + e_{j}b^{i}(0)|^{2} + c_{i}e_{i}^{T} \xi^{(2)} \\
\text{s.t.} \quad & (K(,C^{T})w^{(2)} - e_{j}b^{i}(2)) \geq e_{i} - \xi^{(2)}, \xi^{(2)} \geq 0. \quad (18)
\end{align*}
\]

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

III. PSTWSVM BASED ON INVASIVE WEED OPTIMIZATION ALGORITHM

A. Analysis the Penalty Parameters of PSTWSVM

The role of penalty parameters \( c_{i} \) and \( c_{i} \) is to adjust the ratio between the confidence range with the experience risk in the defining feature, so that the generalization ability of PSTWSVM can achieve the best state. The values of \( c_{i} \) and \( c_{i} \) smaller expresses the punishment on empirical error smaller. Do it this way, the complexity of PSTWSVM is smaller, but its fault tolerant ability is worse. The values of \( c_{i} \) and \( c_{i} \) are greater, the data fitting degree is higher, but its generalization capacity will be reduced. From the above analysis, we can know that the parameters selection is very important for PSTWSVM.

After the above analysis, in this paper, Invasive Weed Optimization (IWO) algorithm which has fast global searching ability is used to select the PSTWSVM parameters and the mixed kernel parameters.

B. Invasive Weed Optimization

In 2006, a novel stochastic optimization model, invasive weed optimization (IWO) algorithm [16], was proposed by Mehrabian and Lucas, which is inspired from a common phenomenon in agriculture: colonization of invasive weeds. Not only it has the robustness, but also it is easy to understand and program. So far, it has been applied in many engineering fields [19-20].

In the classical IWO, weeds represent the feasible solutions of problems and population is the set of all weeds. A finite number of weeds are being dispersed over the search space by normally distributed random numbers with a mean equal to zero. This process continues until maximum number of weeds is reached. Only the weeds with better fitness can survive and produce seed, others are being eliminated. The process continues until maximum iterations are reached or hopefully the weed with best fitness is closest to optimal solution.
The process is addressed in details as follows:

**Step 1: Initialize a population**
A population of initial solutions is being dispersed over the \( D \) dimensional search space with random positions.

**Step 2: Reproduction**
The higher the weed’s fitness is, the more seeds it produces. The formula of seeds producing seeds is

\[
weed_n = f - f_{\text{min}} (s_{\text{max}} - s_{\text{min}}) + s_{\text{min}}
\]

where, \( f \) is the current weed’s fitness. \( f_{\text{max}} \) and \( f_{\text{min}} \) respectively represent the maximum and the least fitness of the current population. \( s_{\text{max}} \) and \( s_{\text{min}} \) respectively represent the maximum and the least value of a weed.

**Step 3: Spatial dispersal**
The generated seeds are randomly distributed over the \( D \) dimensional search space with normally distributed random numbers with a mean equal to zero, but with a varying variance. This ensures that seeds will be randomly distributed so that they abide near to the parent plant. However, standard deviation (\( \sigma \)) of the random function will be reduced from a previously defined initial value (\( \sigma_{\text{init}} \)) to a final value (\( \sigma_{\text{final}} \)) in every generation.

In simulations, a nonlinear alteration has shown satisfactory performance, given as follows

\[
\sigma_{\text{cur}} = \frac{(\text{iter}_{\text{max}} - \text{iter})}{(\text{iter}_{\text{max}})} (\sigma_{\text{init}} - \sigma_{\text{final}}) + \sigma_{\text{final}}
\]

Where, \( \text{iter}_{\text{max}} \) is the maximum number of iterations, \( \sigma_{\text{cur}} \) is the standard deviation at the present time step and \( n \) is the nonlinear modulation index. Generally, \( n \) is set to 3.

**Step 4: Competitive exclusion**
After passing some iteration, the number of weeds in a colony will reach its maximum (\( P_{\text{MAX}} \)) by fast reproduction. At this time, each weed is allowed to produce seeds. The produced seeds are then allowed to spread over the search area. When all seeds have found their position in the search area, they are ranked together with their parents (as a colony of weeds). Next, weeds with lower fitness are eliminated to reach the maximum allowable population in a colony. In this way, weeds and seeds are ranked together and the ones with better fitness survive and are allowed to replicate. The population control mechanism also is applied to their offspring to the end of a given run, realizing competitive exclusion.

C. The Algorithm steps of PSTWSVM-IWO

The accuracy in the sense of CV is used for the fitness of IWO. So the algorithm steps of PSTWSVM-IWO are as follows:

Step1: Select the training dataset and the testing dataset.
Step2: Preprocessing the dataset.
Step3: Construct the PSTWSVM model.
Step4: Select the optimal parameters using IWO algorithm.
Step5: Train the PSTWSVM model using the optimal parameters.
Step6: Predict the testing dataset.
Step7: Output the classification accuracy.

IV. EXPERIMENT RESULTS AND ANALYSIS

In order to verify the efficiency of PSTWSVM and PSTWSVM-IWO, we conduct two experiments. In the first experiment, in order to show the advantage of PSTWSVM, we make experiments on several benchmark datasets using four algorithms, that is, GEPSVM, TWSVM, STWSVM and PSTWSVM. In the second experiment, we make experiment on NDC dataset to compare PSTWSVM with PSTWSVM-IWO. The dual QPPs arising in TWSVM are solved using mosek optimization toolbox for MATLAB [21] which implements fast interior point based algorithms. Classification accuracy of each algorithm is measured by standard tenfold cross-validation methodology.

A. The First Experiment

In this experiment, we make experiments on several benchmark datasets using four algorithms, that is, GEPSVM, TWSVM, STWSVM and PSTWSVM. The optimal parameters of these algorithms are searched from \( \{2^i | i = -6, \ldots, 4\} \) and \( 2^j | i = 6, 4, 2, 0, 1, 2, 4, 6 \) using the grid search algorithm. In PSTWSVM, the parameter of Newton-Armijo method is set \( \epsilon_1 = 1.0E^{-3} \), the approximation accuracy of smooth function is set \( \epsilon_2 = 1.0E^{-3} \). For the nonlinear case, we only consider the Gaussian kernel function. The optimal value of Gaussian kernel parameter is selected over the range \( 2^n | i = -6, -4, -2, 0, 1, 2, 4, 6 \). The order of polynomial is set \( n = 5 \).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>PSTWSVM</th>
<th>STWSVM</th>
<th>TWSVM</th>
<th>GEPSVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hepatitis</td>
<td>78.05±4.31</td>
<td>77.39±2.15</td>
<td>78.08±2.16</td>
<td>77.28±2.78</td>
</tr>
<tr>
<td>Housing</td>
<td>86.21±2.39</td>
<td>84.42±3.87</td>
<td>85.42±4.53</td>
<td>74.81±2.85</td>
</tr>
<tr>
<td>Wdbc</td>
<td>96.10±6.32</td>
<td>94.89±4.31</td>
<td>96.22±6.67</td>
<td>92.81±2.54</td>
</tr>
<tr>
<td>Glass6</td>
<td>96.52±4.56</td>
<td>95.70±6.05</td>
<td>96.55±2.40</td>
<td>96.21±2.72</td>
</tr>
<tr>
<td>Votes</td>
<td>95.50±1.23</td>
<td>94.96±4.24</td>
<td>95.85±2.24</td>
<td>95.63±2.74</td>
</tr>
</tbody>
</table>

© 2014 ACADEMY PUBLISHER
Table 1 shows the comparison of classification accuracy for PSTWSVM with GEPSVM, TWSVM and STWSVM for linear kernel on five UCI datasets. Table 2 shows the comparison of classification performance for nonlinear extensions of PSTWSVM with GEPSVM, TWSVM and STWSVM. Table 1 and table 2 show that the accuracy performance of PSTWSVM is better than STWSVM. Therefore, we can know the approximation ability of polynomial function is better than the sigmoid function.
B. The Second Experiment

Similar to SVM and TWSVM, the learning performance and generalization ability of PSTWSVM is very dependent on its parameters selection. In the above experiment, we used the grid search algorithm to find the parameters values, which is a commonly used method. However, it will lead to low efficiency when dealing with big data. In this paper, we try to use invasive weed optimization (IWO) algorithm to optimize PSTWSVM and propose an algorithm called PSTWSVM-IWO. In this section, we will conduct experiment on NDC datasets which are generated by David Musicant’ NDC Data Generator [22] to test the ability of our algorithm for dealing with big data. The parameters of IWO are as follows: \( D = 5 \), \( P_{MAX} = 30 \), \( s_{max} = 5 \), \( s_{min} = 1 \), \( n = 3 \), \( \sigma_{init} = [1,0.1,1,1,1] \), \( \sigma_{final} = [0.1,0.1,0.1,0.1,0.1] \). In IWO algorithm, the accuracy in the sense of CV is used for the fitness of IWO. Therefore, the fitness value is closer to 100, the obtained parameters is closer to the optimal value. Table 3 gives a description of NDC datasets. Table 4 shows the comparison of computing time and accuracy for three algorithms with linear kernel. On the other hand, table 5 shows the comparison of classification performance for these algorithms with Gaussian kernel. Figure 2–3 are the fitness curves of IWO searching the optimal parameters for dealing with NDC-500, NDC-700, respectively.

From table 4, we can see that in view of the high computing time, TWSVM can’t work when the training samples reach 100000. However, PSTWSM-IWO and PSTWSVM can get reasonable accuracy in the relatively short time when the training samples reach 500000, which indicates that PSTWSM-IWO and PSTWSVM have the advantage on dealing with big data comparing with TWSVM. Furthermore, from table 4 and 5, we also see that the classification accuracy of PSTWSVM-IWO and the computing time are better than PSTWSVM. Therefore, PSTWSVM-IWO is suitable for dealing with big data. Figure 2 and figure 3 shows that the optimization ability of IWO is very strong.

V. CONCLUSION AND FUTURE WORK

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 show that the proposed algorithm can obtain better classification than STWSVM. In view of the good optimization ability of Invasive Weed Optimization (IWO) algorithm, it is used to optimize PSTWSVM in this paper. And then we propose an algorithm called polynomial smooth twin support vector machines based on invasive weed optimization algorithm (PSTWSVM-IWO). We enhance our algorithm to deal with big data, the results indicate that PSTWSVM-IWO is a good method to deal with large datasets.

REFERENCES


Huajuan Huang, born in 1984, received her B.Sc. degree and M.Sc. degree in applied computer science from Guangxi University of Nationalities, Guangxi, China, in 2006 and 2009 respectively. Since 2011, she has been a Ph.D. degree candidate in applied computer science from the University of Mining and Technology, Xuzhou, China. Her current research interests include data mining, pattern recognition and computational intelligence.

Junzhao Yu is currently a graduate student now studying at School of Computer Science and Technology, China University of Mining and Technology. He received his B.Sc. degree in computer science from China University of Mining and Technology in 2011. His research interests include pattern recognition, machine learning, and twin support vector machines et al.

Fulin Wu is currently a graduate student now studying in School of Computer Science and Technology, China University of Mining and Technology, and his supervisor is Prof. Shifei Ding. He received his B.Sc. degree in computer science from China University of Mining and Technology in 2012. His research interests include cloud selection, pattern recognition, machine learning, feature learning et al.

Shifei Ding received his bachelor's degree and master's degree from Qufu Normal University in 1987 and 1998 respectively. He received his Ph.D degree from Shandong University of Science and Technology in 2004. He received postdoctoral degree from Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences in 2006. And now, he works in China University of Mining and Technology as a professor and Ph.D supervisor. 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.