Journal of Computational Science 10 (2015) 270–277

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

Twin support vector machine in linear programs

Dewei Li[a], Yingjie Tian[b,c,*]

[a] Information School, Renmin University of China, Beijing 100872, China
[b] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
[c] Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 10 October 2014
Received in revised form 20 May 2015
Accepted 26 May 2015
Available online 30 May 2015

Keywords:
Twin support vector machine
Binary classification
Linear programs
Structural risk minimization

ABSTRACT

This paper propose a new algorithm, termed as LPTWSVM, for binary classification problem by seeking two nonparallel hyperplanes which is an improved method for TWSVM. We improve the recently proposed ITSVM and develop Generalized ITSVM. A linear function is chosen in the object function of Generalized ITSVM which leads to the primal problems of LPTWSVM. Comparing with TWSVM, a 1-norm regularization term is introduced to the objective function to implement structural risk minimization and the quadratic programming problems are changed to linear programming problems which can be solved fast and easily. Then we do not need to compute the large inverse matrices or use any optimization trick in solving our linear programs and the dual problems are unnecessary in the paper. We can introduce kernel function directly into nonlinear case which overcome the serious drawback of TWSVM. Also, we extend LPTWSVM to multi-class classification problem and get a new model MLPTWSVM. MLPTWSVM constructs M hyperplanes to make that the m-th hyperplane is far from the m-th class and close to the rest classes as much as possible which follow the idea of MBSVM. The numerical experiments verify that our new algorithms are very effective.

1. Introduction

Support vector machines (SVMs), as machine learning methods which were constructed on the VC-dimension theory and the principle of structural risk minimization, were proposed by Corinna Cortes and Vapnik in 1995 [1–3]. With the evolution of SVMs, they have shown much advantages in classification with small samples, nonlinear classification and high dimensional pattern recognition and also they can be applied in solving other machine learning problems [4–10]. The standard support vector classification attempts to minimize generalization error by maximizing the margin between two parallel hyperplanes, which results in dealing with an optimization task involving the minimization of a convex quadratic function. But some classifiers based on nonparallel hyperplanes were proposed recently. The generalized eigenvalue proximal support vector machine (GEPSVM) and twin support vector machine (TWSVM) are two typical classification methods and are also very popular. TWSVM seeks two nonparallel hyperplanes and make each hyperplane close to one class and far from the other as much as possible. However, the structural risk was not considered in TWSVM which may affect the computational efficiency and accuracy. Based on TWSVM, TBSVM and ITSVM was proposed in [11,12] which introduces a regularization term into the objective function and their experiments perform better than TWSVM. The main contribution of the twin bounded support vector machine (TBSVM) is that the principle of structural risk minimization is implemented by adding the regularization term in the primal problems. And the advantages of the improved twin support vector machine (ITSVM) are that it can apply kernel trick directly in the nonlinear case and it does not need to compute inverse matrices. Similar as SVM, a least squares version of TWSVM (LSTWSVM) has been presented which only require to solve two systems of linear equations [13]. LSTWSVM can get comparable generalization performance to TWSVM. Tian etc. proposed NPSVM to make TWSVM sparse by replace the quadratic loss function with $\epsilon-$insensitive loss function [14]. But the above mentioned improved algorithms all have their own drawbacks. TBSVM and LSTWSVM cannot introduce kernel function into linear case directly and LSTWSVM does not consider minimizing structural risk. ITSVM and NPSVM need to solve quadratic programs which need many complex tricks. TWSVM have been studied extensively [15–20].

Since multi-class classification problem is a natural extension of the binary classification problem, "One versus the rest" and "One versus one" [21,22] which construct a series of quadratic programs have been proposed and become two typical methods in solving multi-class classification problems. Recently, MBSVM [23] has been presented for multi-class classification inspired by the idea of TWSVM. For $M(M \geq 3)$ class classification, MBSVM seeks for M

* Corresponding author.
E-mail addresses: ruclidewei@126.com (D. Li), tyj@ucas.ac.cn (Y. Tian).

http://dx.doi.org/10.1016/j.jocs.2015.05.005
1877-7503/

hyperplanes such that each hyperplane is proximal to $M-1$ class and as far as possible from the rest one class. Because the decision criterion of MBSVM is the farthest distance of the test point to the hyperplanes, MBSVM have much lower computational complexity and can be solved faster than "One versus the rest" and "One versus one". However, MBSVM has two serious drawbacks which may affect its predict accuracy. It has to compute the inverse matrices though solving optimization problems which can lead to "Curse of Dimensionality" when the sample size is very large. Besides, it has not taken into account the confidence interval of structural risk so that its learning ability and generalization ability cannot meet requirements.

In this paper, we propose a novel approach to classification problem which involves two nonparallel hyperplanes in two linear optimization problems, termed as LPTWSVM, for binary classification. Since ITSVM is a successful method as an improved version of TWSVM, we develop Generalized ITSVM which follows the idea in [24]. Quadratic programs are time-consuming and need to be solved by so many optimization tricks. In comparison, linear programs are very popular because the formulations are nice which can be computed simpler [25]. So we consider using linear programming to obtain the best hyperplane parameters. LPTWSVM replaces the abstract function in the objective function of Generalized ITSVM with 1-norm terms and then we convert them to linear programs which can be solved easily and quickly and inherits the advantage of ITSVM. We can implement the principle of structural minimization and avoid computing inverse matrices. Also kernel function can be introduced to nonlinear case directly as the standard SVMs usually do. We also make an extension for LPTWSVM and solve multi-class classification at last.

The paper is organized as follows: Section 2 briefly introduces two algorithms, the original TWSVM and ITSVM; Section 3 proposes our new method LPTWSVM; we extended LPTWSVM to multi-class classification in Section 4; numerical experiments are implemented in Section 5 and concluding remarks are summarized in Section 6.

2. Background

In this section, we introduce the original TWSVM and its improved algorithm ITSVM.

2.1. TWSVM

Consider the binary classification problem with the training set

$$T = \{(x_1, +1), \cdots, (x_p, +1), (x_{p+1}, -1), \cdots, (x_{p+q}, -1)\}, \tag{2.1}$$

where $x_i \in R^n$, $i = 1, \cdots, p+q$. Let $A = (x_1, \cdots, x_p)^{\mathrm{T}} \in R^{p\times n}$, $B = (x_{p+1}, \cdots, x_{p+q})^{\mathrm{T}} \in R^{q\times n}$, and $l = p+q$.

TWSVM constructs the following primal problems

$$\min_{w_+,b_+,\xi_-} \frac{1}{2}(Aw_+ + e_+b_+)^{\mathrm{T}}(Aw_+ + e_+b_+) + c_1e_-^{\mathrm{T}}\xi_-, \tag{2.2}$$

$$s.t. -(Bw_+ + e_-b_+) + \xi_- \geq e_-, \xi_- \geq 0, \tag{2.3}$$

and

$$\min_{w_-,b_-,\xi_+} \frac{1}{2}(Bw_- + e_-b_-)^{\mathrm{T}}(Bw_- + e_-b_-) + c_2e_+^{\mathrm{T}}\xi_+, \tag{2.4}$$

$$s.t.(Aw_- + e_+b_-) + \xi_+ \geq e_+, \xi_+ \geq 0, \tag{2.5}$$

where c_i, i=1,2,3,4 are the penalty parameters, e_+ and e_- are vectors of ones, ξ_+ and ξ_- are slack vectors, e_+, $\xi_+ \in R^p$, e_-, $\xi_- \in R^q$. The decision function is denoted by

$$Class = arg\min_{k=-,+} |(w_k \cdot x) + b_k| \tag{2.6}$$

where $|\cdot|$ is the absolute value.

2.2. ITSVM

ITSVM is the abbreviation of improved twin support vector machine which changes the form of TWSVM and construct the following primal problems in linear case

$$\min_{w_+,b_+,\eta_+,\xi_-} \frac{1}{2}c_3(||w_+||^2 + b_+^2) + \frac{1}{2}\eta_+^{\mathrm{T}}\eta_+ + c_1e_-^{\mathrm{T}}\xi_-, \tag{2.7}$$

$$s.t.Aw_+ + e_+b_+ = \eta_+, \tag{2.8}$$

$$-(Bw_+ + e_-b_+) + \xi_- \geq e_-, \xi_- \geq 0, \tag{2.9}$$

and

$$\min_{w_-,b_-,\eta_-,\xi_+} \frac{1}{2}c_4(||w_-||^2 + b_-^2) + \frac{1}{2}\eta_-^{\mathrm{T}}\eta_- + c_2e_+^{\mathrm{T}}\xi_+, \tag{2.10}$$

$$s.t.Bw_- + e_-b_- = \eta_-, \tag{2.11}$$

$$(Aw_- + e_+b_-) + \xi_+ \geq e_+, \xi_+ \geq 0, \tag{2.12}$$

where c_i, i=1,2,3,4 are the penalty parameters, e_+ and e_- are vectors of ones, ξ_+ and ξ_- are slack vectors, e_+, ξ_+, $\eta_+ \in R^p$, e_-, ξ_-, $\eta_- \in R^q$.

The following dual problems are considered to be solved

$$\max_{\lambda,\alpha} -\frac{1}{2}(\lambda^{\mathrm{T}} \quad \alpha^{\mathrm{T}})\hat{Q}(\lambda^{\mathrm{T}} \quad \alpha^{\mathrm{T}})^{\mathrm{T}} + c_3e_-^{\mathrm{T}}\alpha, \tag{2.13}$$

$$s.t.0 \leq \alpha \leq c_1e_-, \tag{2.14}$$

and

$$\max_{\theta,\gamma} -\frac{1}{2}(\theta^{\mathrm{T}} \quad \gamma^{\mathrm{T}})\tilde{Q}(\theta^{\mathrm{T}} \quad \gamma^{\mathrm{T}})^{\mathrm{T}} + c_4e_+^{\mathrm{T}}\gamma, \tag{2.15}$$

$$s.t.0 \leq \gamma \leq c_2e_+, \tag{2.16}$$

where

$$\hat{Q} = \begin{pmatrix} AA^{\mathrm{T}} + c_3I_+ & AB^{\mathrm{T}} \\ BA^{\mathrm{T}} & BB^{\mathrm{T}} \end{pmatrix} + E, \tag{2.17}$$

$$\tilde{Q} = \begin{pmatrix} BB^{\mathrm{T}} + c_4I_- & BA^{\mathrm{T}} \\ AA^{\mathrm{T}} & AA^{\mathrm{T}} \end{pmatrix} + E, \tag{2.18}$$

and I_+ is the $p \times p$ identity matrix, I_- is the $q \times q$ identity matrix, E is the $l \times l$ matrix with all entries equal to one. Thus a new point $x \in R^n$ is predicted to the class by (2.6) where

$$w_+ = -\frac{1}{c_3}(A^{\mathrm{T}}\lambda + B^{\mathrm{T}}\alpha), b_+ = -\frac{1}{c_3}(e_+^{\mathrm{T}}\lambda + e_-^{\mathrm{T}}\alpha), \tag{2.19}$$

$$w_- = -\frac{1}{c_4}(B^{\mathrm{T}}\theta - A^{\mathrm{T}}\gamma), b_- = -\frac{1}{c_4}(e_-^{\mathrm{T}}\theta - e_+^{\mathrm{T}}\gamma), \tag{2.20}$$

For the nonlinear case, after introducing the kernel function, the two corresponding problems in the Hilbert space H are

$$\min_{w_+,b_+,\eta_+,\xi_-} \frac{1}{2}c_3(||w_+||^2 + b_+^2) + \frac{1}{2}\eta_+^{\mathrm{T}}\eta_+ + c_1e_-^{\mathrm{T}}\xi_-, \tag{2.21}$$

$$s.t.\Phi(A)w_+ + e_+b_+ = \eta_+, \tag{2.22}$$

$$-(\Phi(B)w_+ + e_-b_+) + \xi_- \geq e_-, \xi_- \geq 0, \tag{2.23}$$

and

$$\min_{w_-,b_-,\eta_-,\xi_+} \frac{1}{2}c_4(||w_-||^2 + b_-^2) + \frac{1}{2}\eta_-^{\mathrm{T}}\eta_- + c_2e_+^{\mathrm{T}}\xi_+, \tag{2.24}$$

$$s.t.\Phi(B)w_- + e_-b_- = \eta_-, \tag{2.25}$$

$$(\Phi(A)w_- + e_+b_-) + \xi_+ \geq e_+, \xi_+ \geq 0, \tag{2.26}$$

Their dual problems are constructed and can be solved directly.

2.3. Generalized SVM

Generalized SVM is proposed by Mangasarian which used to verify that maximizing the margin between the two support hyperplanes can be accomplished by minimizing any expected norm of support vector multipliers [24]. For a classification problem with training set (2.1), a mathematical programs can be stated as following to obtain a separate hyperplane

$$\min_{w,b,\xi} f(w) + C\sum_{i=1}^{l}\xi_i, \tag{2.27}$$

$$s.t. y_i((w\cdot x_i)+b) \geq 1-\xi_i, i=1,\cdots,l, \tag{2.28}$$

$$\xi_i \geq 0, i=1,\cdots,l. \tag{2.29}$$

where f is a convex function on R^n. It had been proofed that the problem (2.27)–(2.29) has a solution whenever f is a linear or quadratic function.

3. LPTWSVM

In this section, based on ITSVM, we first develop Generalized ITSVM and then introduce our LPTWSVM which changes the 2-norm terms in the objective function of ITSVM to 1-norm terms and then get a pair of linear programs.

3.1. Generalized ITSVM

For the solution (2.19) and (2.20), let $\lambda_+ = -1/c_3\lambda$, $\alpha_+ = 1/c_3\alpha$, $\lambda_- = 1/c_4\theta$, $\alpha_- = 1/c_4\gamma$, and then

$$w_+ = A^{\mathrm{T}}\lambda_+ - B^{\mathrm{T}}\alpha_+, b_+ = e_+^{\mathrm{T}}\lambda_+ - e_-^{\mathrm{T}}\alpha_+, \tag{3.1}$$

$$w_- = -B^{\mathrm{T}}\lambda_- + A^{\mathrm{T}}\alpha_-, b_- = -e_-^{\mathrm{T}}\lambda_- + e_+^{\mathrm{T}}\alpha_-, \tag{3.2}$$

We introduce (3.1) and (3.2) into the primal problems (2.7)–(2.9) and (2.10)–(2.12) thus get

$$\min_{\lambda_+,\alpha_+,\eta_+,\xi_-} \frac{1}{2}c_3(\lambda_+^{\mathrm{T}} \ \alpha_+^{\mathrm{T}})Q_+(\lambda_+^{\mathrm{T}} \ \alpha_+^{\mathrm{T}})^{\mathrm{T}} + \frac{1}{2}\eta_+^{\mathrm{T}}\eta_+ + c_1e_-^{\mathrm{T}}\xi_-, \tag{3.3}$$

$$s.t.(AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\lambda_+ - (AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\alpha_+ = \eta_+, \tag{3.4}$$

$$-(BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\lambda_+ + (BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\alpha_+ + \xi_- \geq e_-, \tag{3.5}$$

$$\alpha_+ \geq 0, \xi_- \geq 0, \tag{3.6}$$

and

$$\min_{\lambda_-,\alpha_-,\eta_-,\xi_+} \frac{1}{2}c_4(\lambda_-^{\mathrm{T}} \ \alpha_-^{\mathrm{T}})Q_-(\lambda_-^{\mathrm{T}} \ \alpha_-^{\mathrm{T}})^{\mathrm{T}} + \frac{1}{2}\eta_-^{\mathrm{T}}\eta_- + c_2e_+^{\mathrm{T}}\xi_+, \tag{3.7}$$

$$s.t. -(BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\lambda_- + (BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\alpha_- = \eta_-, \tag{3.8}$$

$$-(AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\lambda_- + (AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\alpha_- + \xi_+ \geq e_+, \tag{3.9}$$

$$\alpha_- \geq 0, \xi_+ \geq 0, \tag{3.10}$$

where

$$Q_+ = \begin{pmatrix} AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}} & -AB^{\mathrm{T}} - e_+e_-^{\mathrm{T}} \\ -BA^{\mathrm{T}} - e_-e_+^{\mathrm{T}} & BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}} \end{pmatrix}, \tag{3.11}$$

$$Q_- = \begin{pmatrix} BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}} & -BA^{\mathrm{T}} - e_-e_+^{\mathrm{T}} \\ -AB^{\mathrm{T}} - e_+e_-^{\mathrm{T}} & AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}} \end{pmatrix}, \tag{3.12}$$

We state mathematical programs by replacing the first term in the object function with abstract function f, g as follows:

$$\min_{\lambda_+,\alpha_+,\eta_+,\xi_-} f(\lambda_+,\alpha_+) + \frac{1}{2}\eta_+^{\mathrm{T}}\eta_+ + c_1e_-^{\mathrm{T}}\xi_-, \tag{3.13}$$

$$s.t.(AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\lambda_+ - (AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\alpha_+ = \eta_+, \tag{3.14}$$

$$-(BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\lambda_+ + (BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\alpha_+ + \xi_- \geq e_-, \tag{3.15}$$

$$\alpha_+ \geq 0, \xi_- \geq 0, \tag{3.16}$$

and

$$\min_{\lambda_-,\alpha_-,\eta_-,\xi_+} g(\lambda_-,\alpha_-) + \frac{1}{2}\eta_-^{\mathrm{T}}\eta_- + c_2e_+^{\mathrm{T}}\xi_+, \tag{3.17}$$

$$s.t. -(BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\lambda_- + (BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\alpha_- = \eta_-, \tag{3.18}$$

$$-(AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\lambda_- + (AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\alpha_- + \xi_+ \geq e_+, \tag{3.19}$$

$$\alpha_- \geq 0, \xi_+ \geq 0, \tag{3.20}$$

where f, g are some convex functions on $R^p \times R^q$.

3.2. Linear programming ITSVM (LPTWSVM)

In this section, we consider using linear function f, g in the objective function generated from the mathematical program (3.13)–(3.20) thus leading to linear programs. We chose 1-norm $\lambda_+, \alpha_+, \lambda_-, \alpha_-$ for f, g and change the η term to 1-norm form which leads to the following primal problems:

$$\min_{\lambda_+,\alpha_+,\eta_+,\xi_-} c_3(||\lambda_+|| + ||\alpha_+||) + ||\eta_+|| + c_1e_-^{\mathrm{T}}\xi_-, \tag{3.21}$$

$$s.t.(AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\lambda_+ - (AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\alpha_+ = \eta_+, \tag{3.22}$$

$$-(BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\lambda_+ + (BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\alpha_+ + \xi_- \geq e_-, \tag{3.23}$$

$$\alpha_+, \xi_- \geq 0, \tag{3.24}$$

and

$$\min_{\lambda_-,\alpha_-,\eta_-,\xi_+} c_4(||\lambda_-|| + ||\alpha_-||) + ||\eta_-|| + c_2e_+^{\mathrm{T}}\xi_+, \tag{3.25}$$

$$s.t. -(BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\lambda_- + (BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\alpha_- = \eta_-, \tag{3.26}$$

$$-(AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\lambda_- + (AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\alpha_- + \xi_+ \geq e_+, \tag{3.27}$$

$$\alpha_-, \xi_+ \geq 0, \tag{3.28}$$

where $||\cdot||$ denote 1-norm, c_i, i=1,2,3,4 are the penalty parameters.

We introduce the variables $s_+ = (s_{+,1}, s_{+,2}, \cdots, s_{+,p})^{\mathrm{T}}$, $t_+ = (t_{+,1}, t_{+,2}, \cdots, t_{+,p})^{\mathrm{T}}$, $s_- = (s_{-,1}, s_{-,2}, \cdots, s_{-,q})^{\mathrm{T}}$, $t_- = (t_{-,1}, t_{-,2}, \cdots, t_{-,q})^{\mathrm{T}}$, then we can convert the primal problems (3.21)–(3.24) and (3.25)–(3.28) to linear programming formulation:

$$\min_{\lambda_+,\alpha_+,s_+,t_+,\xi_-} c_3(e_+^{\mathrm{T}}s_+ + e_-^{\mathrm{T}}\alpha_+) + e_+^{\mathrm{T}}t_+ + c_1e_-^{\mathrm{T}}\xi_-, \tag{3.29}$$

$$s.t. -t_+ \leq (AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\lambda_+ - (AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\alpha_+ \leq t_+, \tag{3.30}$$

$$-(BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\lambda_+ + (BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\alpha_+ + \xi_- \geq e_-, \tag{3.31}$$

$$-s_+ \leq \lambda_+ \leq s_+, \tag{3.32}$$

$$s_+, t_+, \alpha_+, \xi_- \geq 0, \tag{3.33}$$

and

$$\min_{\lambda_-,\alpha_-,s_-,t_-,\xi_+} c_4(e_-^{\mathrm{T}}s_- + e_+^{\mathrm{T}}\alpha_-) + e_-^{\mathrm{T}}t_- + c_2e_+^{\mathrm{T}}\xi_+, \tag{3.34}$$

$$s.t. -t_- \leq -(BB^{\mathrm{T}} + e_-e_-^{\mathrm{T}})\lambda_- + (BA^{\mathrm{T}} + e_-e_+^{\mathrm{T}})\alpha_- \leq t_-, \tag{3.35}$$

$$-(AB^{\mathrm{T}} + e_+e_-^{\mathrm{T}})\lambda_- + (AA^{\mathrm{T}} + e_+e_+^{\mathrm{T}})\alpha_- + \xi_+ \geq e_+, \tag{3.36}$$

$$-s_- \leq \lambda_- \leq s_-, \tag{3.37}$$

$$s_-, t_-, \alpha_-, \xi_+ \geq 0, \tag{3.38}$$

Then an unknown point is predicted to the class by (2.6) where w_+, b_+, w_-, b_- are same as (3.1) and (3.2).

In nonlinear case, we introduce the transformation $x = \Phi(x)$ and the corresponding kernel function $K(x, x') = (\Phi(x) \cdot \Phi(x'))$ where $x \in H$, H is the Hilbert space. So the training set (2.1) becomes

$$\tilde{T} = \{(x_1, +1), \cdots, (x_p, +1), (x_{p+1}, -1), \cdots, (x_{p+q}, -1)\}, \tag{3.39}$$

Then we construct the following primal problems

$$\min_{\lambda_+, \alpha_+, s_+, t_+, \xi_-} c_3(e_+^T s_+ + e_-^T \alpha_+) + e_+^T t_+ + c_1 e_-^T \xi_-, \tag{3.40}$$

$$s.t. - t_+ \le (K(A, A^T) + e_+ e_+^T)\lambda_+ - (K(A, B^T) + e_+ e_-^T)\alpha_+ \le t_+, \tag{3.41}$$

$$-(K(B, A^T) + e_- e_+^T)\lambda_+ + (K(B, B^T) + e_- e_-^T)\alpha_+ + \xi_- \ge e_-, \tag{3.42}$$

$$-s_+ \le \lambda_+ \le s_+, \tag{3.43}$$

$$s_+, t_+, \alpha_+, \xi_- \ge 0, \tag{3.44}$$

and

$$\min_{\lambda_-, \alpha_-, s_-, t_-, \xi_+} c_4(e_-^T s_- + e_+^T \alpha_-) + e_-^T t_- + c_2 e_+^T \xi_+, \tag{3.45}$$

$$s.t. - t_- \le -(K(B, B^T) + e_- e_-^T)\lambda_- + (K(B, A^T) + e_- e_+^T)\alpha_- \le t_-, \tag{3.46}$$

$$-(K(A, B^T) + e_+ e_-^T)\lambda_- + (K(A, A^T) + e_+ e_+^T)\alpha_- + \xi_+ \ge e_+, \tag{3.47}$$

$$-s_- \le \lambda_- \le s_-, \tag{3.48}$$

$$s_-, t_-, \alpha_-, \xi_+ \ge 0, \tag{3.49}$$

Then an unknown point is predicted to the class by

$$Class = arg \min_{k=-,+} |f_k(x)|, \tag{3.50}$$

where

$$f_+(x) = K(x, A^T)\lambda_+ - K(x, B^T)\alpha_+ + e_+^T \lambda_+ - e_-^T \alpha_+ \tag{3.51}$$

$$f_-(x) = -K(x, B^T)\lambda_- + K(x, A^T)\alpha_- - e_-^T \lambda_- + e_+^T \alpha_- \tag{3.52}$$

4. LPTWSVM for multi-class classification

4.1. Classic models for multi-class classification

In this section, we introduce several algorithms for multi-class classification. We first give the training set for the multi-class classification as following:

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\}, \tag{4.1}$$

where $x_i \in R^n$, $y_i \in \mathcal{Y} = \{1, 2, \cdots, M\}$, $i = 1, \cdots, l$. The (x_i, y_i) is the i-th data point, $x_i \in R^n$ is the input and the class label y_i is the output. Our task is to find a decision function $f(x)$ in R^n, such that the class number y for any x can be predicted by $y = f(x)$.

"One versus One" regard the original problem as $M(M-1)/2$ binary problems and construct binary classifier for each problem. It use voting scheme methods based on combining many binary classification decision function. "One versus the rest" constructs M binary classifiers and predicts an new input in the class which has the largest corresponding decision function. We then introduce a recently proposed algorithm for multi-class classification.

The number of data points of the k-th class in the training set (4.1) was denoted as l_k and the following matrixes were defined: the inputs belonging to the k-th class are represented by the matrix $A_k \in R^{l_k \times n}$, $k = 1, \cdots, M$. In addition, we define the matrix

$$B_k = [A_1^T, \cdots, A_{k-1}^T, A_{k+1}^T, \cdots, A_M^T], \tag{4.2}$$

that is, $B_k \in R^{(l-l_k)\times n}$ is comprised of the patterns belonging to all classes except the k-th class, $k = 1, \cdots, M$.

For the linear case, MBSVM constructs M primal problems as following:

$$(\text{MBSVMk}) \min_{w_k, b_k, \xi_k} \frac{1}{2}||B_k w_k + e_{k1} b_k||^2 + C_k e_{k2}^T \xi_k, \tag{4.3}$$

$$s.t.(A_k w_k + e_{k2} b_k) + \xi_k \ge e_{k2}, \xi_k \ge 0, \tag{4.4}$$

where the matrix A_k, B_k were defined as above, $e_{k1} \in R^{(l-l_k)}$ and $e_{k2} \in R^{l_k}$ are the vector of ones, ξ_k is the slack variable, $C_k > 0$ is the penalty parameter.

Introducing the non-negative vectors of Lagrangian multiplier $\alpha_k \in R^{l_k}$ and $\beta_k \in R^{l_k}$ and using the KKT conditions, the dual problem of MBSVMk is

$$(\text{DMBSVMk}) \max_{\alpha_k} e_{k2}^T \alpha_k - \frac{1}{2}\alpha_k G_k (H_k^T H)^{-1} G_k^T \alpha_k, \tag{4.5}$$

$$s.t. 0 \le \alpha_k \le C_k, \tag{4.6}$$

where the penalty parameter $C_k > 0$ and $H_k = [B_k \ e_{k1}]$, $G_k = [A_k \ e_{k2}]$.

Then the solutions of problems (4.3)–(4.4) can be obtained by

$$[w_k^T, b_k]^T = (H_k^T H_k)^{-1} G_k^T \alpha_k. \tag{4.7}$$

A new input $x \in R^n$ is predicted to the class by

$$f(x) = arg \max_{k=1,\cdots,M} |(w_k \cdot x) + b_k| \tag{4.8}$$

where $|\cdot|$ is the absolute value.

MBSVM has claimed that it has low complexity and can be computed faster than "One versus One" and "One versus the rest", but it has two serious drawbacks which may affect its accuracy. First, it cannot avoid computing inverse matrices, which is not tractable in practice; second, it has not considered structural risk. As we all know, one notable advantage of SVC is that it have implemented the structural risk minimization principle. However, only the empirical risk is considered in the primal problems of MBSVM. Then we can improve MBSVM in two points: (1) change the 2-norm term in the objective function to 1-norm term; (2) introduce a regularization term to the objective function.

4.2. MLPTWSVM

Follow the idea of MBSVM, we can also solve multi-class classification problems based on LPTWSVM, termed as MLPTWSVM, which can avoid computing the inverse matrices and implement the principle of structural risk minimization. The principle of MLPTWSVM is depicted in Fig. 1.

A. Linear MLPTWSVM

For the problem with the training set (4.1), our linear MLPTWSVM seeks M hyperplanes

$$(w_k \cdot x) + b_k = 0, k = 1, \cdots, M, \tag{4.9}$$

each hyperplane is assigned to a class. We make the inputs in the k-th class as far as possible to the k-th hyperplane while the inputs in the rest $M-1$ are proximal to the k-th hyperplane. A new input is predicted to class y when it is farthest to the hyperplane which was assigned to class y. Then the primal problems are constructed as following

$$\min_{\lambda_k, \alpha_k, \eta_k, \xi_k} d_k(||\lambda_k|| + ||\alpha_k||) + ||\eta_k|| + c_k e_{k1}^T \xi_k, \tag{4.10}$$

$$s.t. -(B_k B_k^T + e_{k2} e_{k2}^T)\lambda_k + (B_k A_k^T + e_{k2} e_{k1}^T)\alpha_k = \eta_k, \tag{4.11}$$

$$-(A_k B_k^T + e_{k1} e_{k2}^T)\lambda_k + (A_k A_k^T + e_{k1} e_{k1}^T)\alpha_k + \xi_k \ge e_{k1}, \tag{4.12}$$

$$\alpha_k, \xi_k \ge 0, \tag{4.13}$$

where $||\cdot||$ denote 1-norm, $d_k > 0$, $c_k > 0$, $k = 1, \cdots, M$ are the penalty parameters. The matrix A_k, B_k are the same as that in

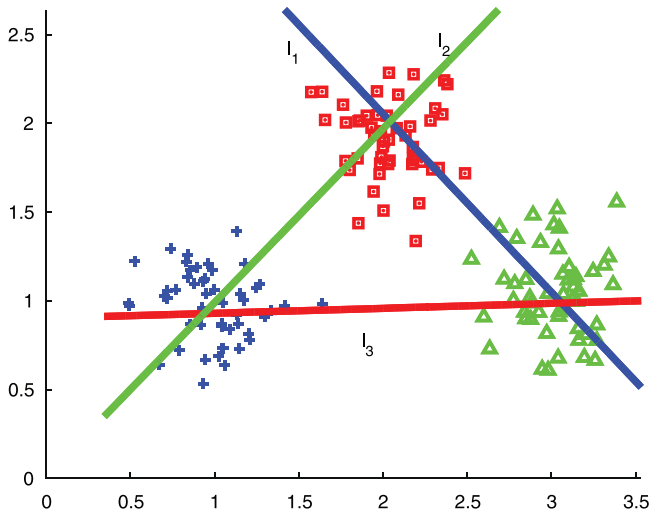

Fig. 1. The concept figure of MLPTWSVM [23].

MBSVM, $e_{k2} \in R^{(l-l_k)}$ and $e_{k1} \in R^{l_k}$ are the vector of ones, ξ_k is the slack variable.

Since the objective function in primal problem (4.10)–(4.13) is not differentiable, we introduce the variables $s_k = (s_{k,1}, s_{k,2}, \cdots, s_{k,n})^{\mathrm{T}}$, $t_k = (t_{k,1}, t_{k,2}, \cdots, t_{k,n})^{\mathrm{T}}$, then we can get the following linear programs

$$\min_{\lambda_k,\alpha_k,s_k,t_k,\xi_k} d_k(e_{k2}^{\mathrm{T}} s_k + e_{k1}^{\mathrm{T}} \alpha_k) + e_{k2}^{\mathrm{T}} t_k + c_k e_{k1}^{\mathrm{T}} \xi_k, \tag{4.14}$$

$$s.t. - t_k \le -(B_k B_k^{\mathrm{T}} + e_{k2} e_{k2}^{\mathrm{T}})\lambda_k + (B_k A_k^{\mathrm{T}} + e_{k2} e_{k1}^{\mathrm{T}})\alpha_k \le t_k, \tag{4.15}$$

$$-(A_k B_k^{\mathrm{T}} + e_{k1} e_{k2}^{\mathrm{T}})\lambda_k + (A_k A_k^{\mathrm{T}} + e_{k1} e_{k1}^{\mathrm{T}})\alpha_k + \xi_k \ge e_{k1}, \tag{4.16}$$

$$-s_k \le \lambda_k \le s_k, \tag{4.17}$$

$$s_k, t_k, \alpha_k, \xi_k \ge 0, \tag{4.18}$$

Then we can get the solutions $w_k, b_k (k = 1, \cdots, M)$ of the primal problems (4.10)–(4.13). A new input $x \in R^n$ is predicted to class k depending on the decision function (4.8).

B. Nonlinear MLPTWSVM

For the nonlinear case, we can extend our linear MLPTWSVM to nonlinear case and construct the following primal problems

$$\min_{\lambda_k,\alpha_k,\eta_k,\xi_k} d_k(||\lambda_k|| + ||\alpha_k||) + ||\eta_k|| + c_k e_{k1}^{\mathrm{T}} \xi_k, \tag{4.19}$$

$$s.t. (K(B_k, A_k^{\mathrm{T}}) + e_{k2} e_{k1}^{\mathrm{T}})\alpha_k - (K(B_k, B_k^{\mathrm{T}}) + e_{k2} e_{k2}^{\mathrm{T}})\lambda_k = \eta_k, \tag{4.20}$$

$$(K(A_k, A_k^{\mathrm{T}}) + e_{k1} e_{k1}^{\mathrm{T}})\alpha_k - (K(A_k, B_k^{\mathrm{T}}) + e_{k1} e_{k2}^{\mathrm{T}})\lambda_k + \xi_k \ge e_{k1}, \tag{4.21}$$

$$\alpha_k, \xi_k \ge 0, \tag{4.22}$$

also it can be converted to the following program

$$\min_{\lambda_k,\alpha_k,s_k,t_k,\xi_k} d_k(e_{k2}^{\mathrm{T}} s_k + e_{k1}^{\mathrm{T}} \alpha_k) + e_{k2}^{\mathrm{T}} t_k + c_k e_{k1}^{\mathrm{T}} \xi_k, \tag{4.23}$$

$$s.t. - t_k \le (K(B_k, A_k^{\mathrm{T}}) + e_{k2} e_{k1}^{\mathrm{T}})\alpha_k - (K(B_k, B_k^{\mathrm{T}}) + e_{k2} e_{k2}^{\mathrm{T}})\lambda_k \le t_k, \tag{4.24}$$

$$-(K(A_k, A_k^{\mathrm{T}}) + e_{k1} e_{k1}^{\mathrm{T}})\alpha_k + (K(A_k, B_k^{\mathrm{T}}) + e_{k1} e_{k2}^{\mathrm{T}})\lambda_k + \xi_k \ge e_{k1} \tag{4.25}$$

$$-s_k \le \lambda_k \le s_k, \tag{4.26}$$

$$s_k, t_k, \alpha_k, \xi_k \ge 0, \tag{4.27}$$

The decision function is

$$f(x) = arg \max_{k=1,\cdots,M} |K(x, C^{\mathrm{T}})u_k + b_k| \tag{4.28}$$

where $|\cdot|$ is the absolute value.

It can be seen that we have introduced a regularization term in the objective function which is used to minimize the structural risk. And we converted the primal problems to linear programs which can make us avoid computing inverse matrices. Since all our programs are linear programs which can be solved directly and easily, we do not need any extra tricks in optimization.

5. Numerical experiments

In this section, we have made numerical experiments to show the advantages of our algorithms. All experiments were implemented in MATLAB 2012a on a PC with a Intel Core 2 processor with 2GB RAM and all datasets were come from the UCI machine learning Repository. In order to get more objective results, some datasets were partially selected from the primary datasets to get

Table 1
Average accuracy in linear case of binary classification (%).

Dataset inst. × attr.	*SVM* Accuracy C	TWSVM Accuracy c_1/c_2	ITSVM Accuracy/time (s) $c_1/c_2/c_3/c_4$	LPTWSVM Accuracy/time (s) $c_1/c_2/c_3/c_4$
WDBC	94.98	93.48	**95.97**/1.21	**95.97**/0.80
(200 × 30)	100	1/1	0.1/0.1/0.1/0.1	1/1/0.01/0.1
Votes	93.00	94.33	94.00/2.20	**95.00**/3.57
(300 × 16)	1	1/0.1	0.1/0.1/0.01/0.01	1/1/1/1
Hepatitis	80.96	82.60	**85.16**/0.98	83.93/0.46
(155 × 19)	0.1	10/1	10/100/100/100	1/1/0.1/0.1
Heart-statlog	83.33	82.59	84.07/2.05	**85.93**/1.49
(270 × 13)	1	0.1/0.1	0.01/0.01/1/1	1/0.1/10/10
Heart-c	82.99	83.01	83.51/0.83	**86.30**/0.73
(200 × 13)	0.1	0.1/0.1	0.1/0.1/0.1/0.1	10/10/10/10
Ionosphere	84.05	89.02	86.01/5.33	**89.04**/3.58
(200 × 34)	10	100/1	1/1/0.01/0.01	100/1/0.01/0.01
Sonar	79.32	77.87	**80.27**/2.04	**80.27**/0.87
(208 × 60)	1	10/1	10/10/10/10	10/100/10/1
Bupa	68.41	69.57	**69.86**/0.65	66.96/2.60
(345 × 6)	100	0.01/0.01	0.01/0.01/0.01/0.01	1/1/0.01/0.01
Hungarian	83.05	84.00	84.54/2.08	**85.49**/1.41
(200 × 13)	10	100/100	100/100/1/1	100/100/0.01/0.01
Blood	67.33	68.00	68.00/0.56	**68.67**/1.26
(300 × 4)	1	1/1	0.1/0.1/1/1	1/1/0.01/0.01

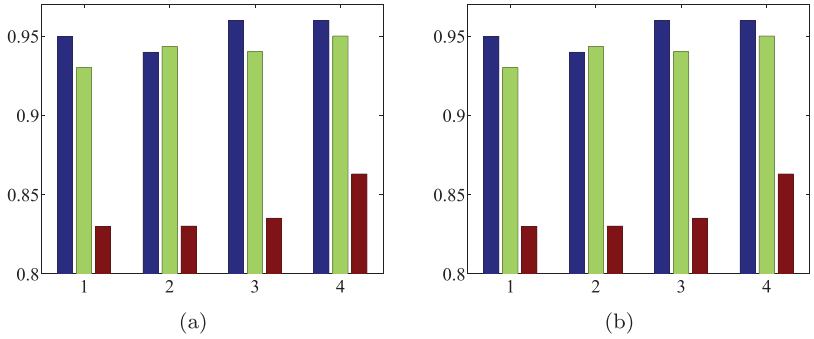

Fig. 2. Comparison of prediction accuracy. The left subfigure is in linear case and the right one is in nonlinear case. For the *X* axis in both subfigures, 1,2,3,4 represents SVM, TWSVM, ITSVM and LPTWSVM respectively. For each algorithm, blue, green and brown represents WDBC, Votes and Heart-c dataset severally.

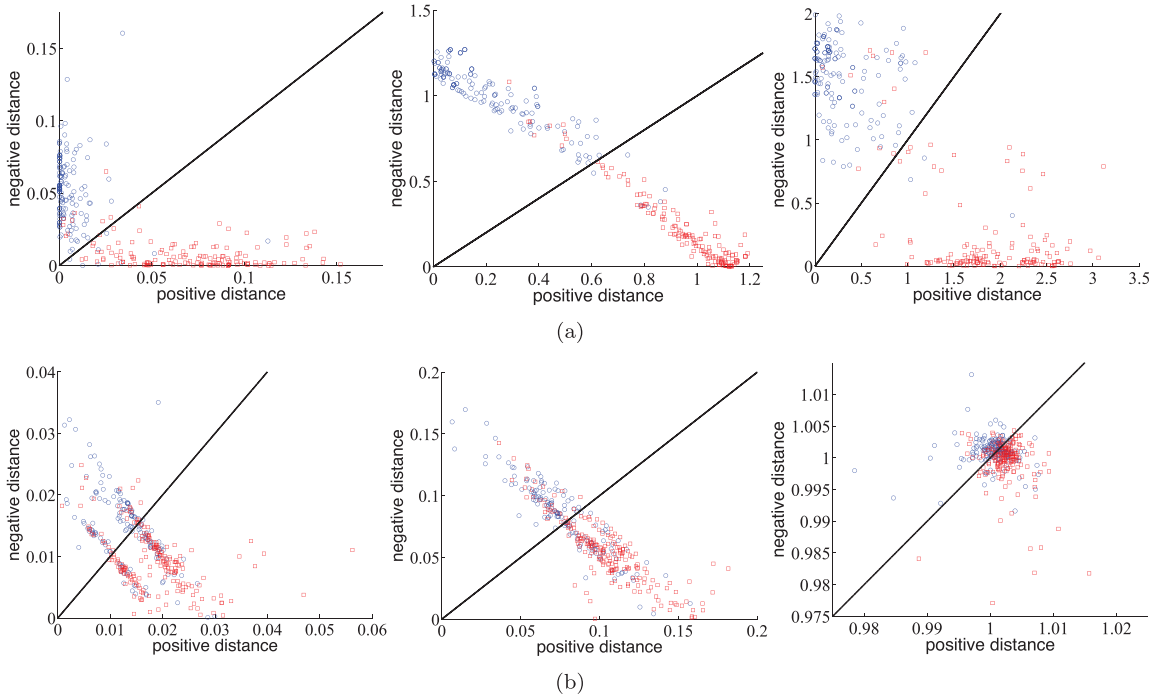

Fig. 3. The distances between points and hyperplanes. (a) Figures of **Votes**. From left to right: TWSVM, ITSVM, LPTWSVM. (b) Figures of **Bupa**. From left to right: **TWSVM, ITSVM, LPTWSVM**. In all figures, the red points belong to positive class and the blue points belong to negative points. The black slants denote function $y = x$. (For interpretation of the references to color in this text, the reader is referred to the web version of the article.)

Table 2
Average accuracy in nonlinear case of binary classification(%).

	SVM	TWSVM	ITSVM	LPTWSVM
Dataset	Accuracy	Accuracy	Accuracy/time (s)	Accuracy/time (s)
inst. × attr.	C/μ	$c_1/c_2/\mu$	$c_1/c_2/c_3/c_4/\mu$	$c_1/c_2/c_3/c_4/\mu$
WDBC	94.77	95.99	95.48/0.50	**95.99**/0.54
(200 × 30)	1/1	0.1/0.1/0.01	1/1/0.1/0.1/0.1	100/100/0.1/0.1/0.1
Votes	94.00	94.67	95.00/0.52	**95.33**/2.48
(300 × 16)	10/1	0.1/0.1/0.1	1/1/1/1/1	100/100/0.1/0.1/0.1
Hepatitis	82.62	82.60	**83.23**/0.27	**83.23**/0.34
(155 × 19)	1/0.1	0.1/0.01/0.01	1/1/0.1/0.1/0.01	10/10/10/10/0.01
Heart-statlog	83.70	83.33	83.33/0.63	**83.70**/1.09
(270 × 13)	100/0.01	0.01/0.1/0.01	10/10/1/1/0.01	1/1/1/1/0.1
Heart-c	83.99	83.54	84.00/0.52	**85.49**/0.70
(200 × 13)	100/0.1	0.1/10/0.01	10/10/10/10/0.01	1/1/0.1/0.1/0.1
Ionosphere	93.09	91.04	**93.60**/0.92	92.04/3.50
(200 × 34)	100/1	0.01/0.1/0.01	1/1/0.01/0.01/1	1/1/0.1/0.1/0.1
Sonar	87.49	85.09	**88.46**/0.33	84.14/0.81
(208 × 60)	10/1	1/1/1	10/10/0.01/0.01/1	1/1/0.01/0.01/0.1
Bupa	72.75	73.33	74.20/9.79	**74.78**/5.61
(345 × 6)	10/1	1/1/0.1	1/1/0.1/0.1/1	10/10/0.01/0.01/0.1
Hungarian	83.04	84.57	**85.04**/2.11	83.51/2.44
(200 × 13)	100/0.1	0.01/0.1/0.01	10/10/100/100/0.1	100/100/0.1/0.1/0.1
Blood	69.00	68.33	68.33/7.61	**69.67**/1.88
(300 × 4)	10/1	1/1/0.1	1/1/0.01/0.01/1	10/0.1/10/0.1/0.01

Table 3
Average accuracy in nonlinear case of multi-class classification(%).

	1-v-1	1-v-r	MBSVM	MLPTWSVM
Dataset	Accuracy	Accuracy	Accuracy	Accuracy
inst. × attr. ♯class	C/μ	C/μ	C/μ	d/c/μ
Iris	96.67	97.33	96.00	**97.33**
(150 × 4♯ 3)	1/10	1/10	1/0.1	0.1/1/1
Wine	98.31	98.87	97.18	**98.88**
(178 × 13 ♯ 3)	1/1	0.1/0.1	1/1	0.01/1/1
Seeds	93.81	94.76	95.71	**96.19**
(210 × 7♯ 3)	10/1	0.1/100	10/0.1	0.01/100/0.1
Glass	65.71	66.67	65.71	**66.67**
(214 × 10♯ 6)	1/100	1/1	10/1	0.1/100/10
Thyroid	97.21	**97.22**	95.84	96.27
(215 × 5♯ 3)	1/100	0.1/0.01	0.1/1	10/100/1
Segment	95.29	**96.14**	95.29	93.43
(700 × 19♯ 7)	1/100	1/100	1/1	0.01/10/1

balanced datasets. All samples were scaled to the interval [0,1] before training to improve the computational efficiency. All the optimal parameters are selected through searching the set $\{10^{-2}, \cdots, 10^{2}\}$. The best accuracy of each dataset is obtained by three-fold cross validation method.

We performed the experiments in linear case and nonlinear case respectively. For all the methods, we applied the RBF kernel $K(x, x') = exp(-\mu\|x-x'\|^2)$ to nonlinear case. The "Accuracy" used to evaluate methods is defined same as [12]. Accuracy = (TP + TN)/(TP + FP + TN + FN), where TP, FP, TN and FN is the number of true positive, false positive, true negative and false negative, respectively. The experiments results are listed in Tables 1 and 2. The best accuracy is signed by bold-face.

In Table 1, we compare our LPTWSVM with SVC, TWSVM and ITSVM in linear case, the classification accuracy and the optimal parameters are listed. In fact, for ITSVM, the choice of c_3 and c_4 affects the results significantly which shows that there are only two parameters to be tuned in practice [12]. For LPTWSVM, there are four parameters need to be tuned and every parameter can affect the results markedly. But it is observed that, in most cases, the best accuracy is got when $c_1 = c_2$ and $c_3 = c_4$ which can save much training time in practice. We have more space to adjust the parameters which leads to higher classification accuracy. We compare LPTWSVM with SVC, TWSVM and ITSVM in nonlinear case in Table 2. All methods can get better accuracy since the kernel function is introduced. Our LPTWSVM have achieved the best accuracy in seven datasets. We have recorded the CPU time to compare computational speed. In both linear and nonlinear case, LPTWSVM have comparable speed with ITSVM. Besides, we have made bar chart in Fig. 2 to compare accuracy more intuitively. And the Fig. 3 show us the distances between points and hyperplanes which were obtained in the experiments on **Votes** and **Bupa**. In the tables and figures, we can see that LPTWSVM has got higher accuracy than the other methods on most datasets which demonstrate the effectiveness of our LPTWSVM on binary classification.

In multi-class classification, we made numerical experiments to compare with other algorithms, including 1-v-1,1-v-r and MBSVM. The experiments results are listed in Table 3. Since the results in linear case are not as good as that in nonlinear case, we have only show the nonlinear classification results. The results verify that the extended algorithm MLPTWSVM can make multi-class classification problem effectively.

6. Conclusion

In this paper, based on TWSVM and ITSVM, we have proposed another improved method for binary classification, terms as LPTWSVM. Instead of solving a large sized programming problems in standard SVMS, we solve two linear optimization problems of a smaller size in LPTWSVM similar as ITSVM. ITSVM has been developed to Generalized ITSVM and we get our primal problems by using a linear function in the objective function of Generalized ITSVM. In contrast to the original TWSVM, our LPTWSVM introduced 1-norm regularization term to the objective function which contribute to the structural risk minimization. The primal problems of LPTWSVM can be converted to linear programs easily and the weights between the regularization term and empirical risk can be adjusted freely. Furthermore, we do not need to calculate the inversion of matrices which can affect the computational accuracy. Numerical experiments have been made on ten datasets and the results show that our LPTWSVM performs better on most datasets, namely, we have higher generalization ability. Besides, we have extended LPTWSVM to multi-class classification which follow the idea of MBSVM and the experiments also verify that MLPTWSVM can solve multi-class classification problem effectively. The idea can be introduced to regression machine, knowledge-based learning in future work.

Acknowledgements

This work has been partially supported by grants from National Natural Science Foundation of China (Nos. 61472390, 11271361, 71331005), Major International (Ragional) Joint Research Project (No. 71110107026), the Ministry of water resources' special funds for scientific research on public causes (No. 201301094).

References

[1] C. Cortes, V.N. Vapnik, Support-vector network, Mach. Learn. 20 (3) (1995) 273–297.
[2] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1996.
[3] V.N. Vapnik, Statistical Learning Theory, John Wiley and Sons, New York, 1998.
[4] N. Deng, Y. Tian, C. Zhang, Support Vector Machine-theories, Algorithms and Extensions, Science Press, Beijing, 2009.
[5] J. Zhu, S. Rosset, T. Hastie, R. Tibshirani, 1-Norm support vector machines, Neural Inf. Process. Syst. (2003) 16.
[6] O.L. Mangasarian, Exact 1-norm support vector machines via unconstrained convex differentiable minimization, Mach. Learn. Res. 7 (2006) 1517–1530.
[7] Y. Tian, Y. Shi, X. Liu, Recent advances on support vector machines research, Technol. Econ. Dev. Econ. 18 (1) (2012) 5–33.
[8] P.S. Bradley, O.L. Mangasarian, Feature selection via concave minimization and support vector machines Machine Learning Proceedings of the Fifteenth International Conference, vol. 98, 1998, pp. 82–90.
[9] C. Zhang, D. Li, J. Tan, The support vector regression with adaptive norms, Procedia Comput. Sci. 18 (2013) 1730–1736.
[10] C. Zhang, X. Shao, D. Li, Knowledge-based support vector classification based on c-svc, Procedia Comput. Sci. 17 (2013) 1083–1090.
[11] Y. Tian, X. Ju, Z. Qi, Y. Shi, Improved twin support vector machine, Sci. China Math. 57 (2) (2014) 417–432.
[12] Y. Shao, C. Zhang, X. Wang, N. Deng, Improvements on twin support vector machines, IEEE Trans. Neural Netw. 22 (6) (2011) 962–968.
[13] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (2009) 7535–7543.

[14] Y. Tian, Z. Qi, X. Ju, Y. Shi, X. Liu, Nonparallel support vector machines for pattern classification, IEEE Trans. Cybern. 44 (7) (2014) 1067–1079.
[15] R.K. Jayadeva, J.R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[16] Z. Qi, Y. Tian, Yongshi, Robust twin support vector machine for pattern classification, Pattern Recognit. 46 (1) (2013) 305–316.
[17] Z. Qi, Y. Tian, Yongshi, Laplacian twin support vector machine for semi-supervised classification, Neural Netw. 35 (2012) 46–53.
[18] Z. Qi, Y. Tian, Yongshi, Twin support vector machine with universum data, Neural Netw. 36 (2012) 112–119.
[19] M.A. Kumar, M. Gopal, Application of smoothing technique on twin support vector machines, Pattern Recognit. Lett. 29 (2008) 1842–1848.
[20] J.R. Khemchandani, R.K. Jayadeva, S. Chandra, Optimal kernel selection in twin support vector machines, Optim. Lett. 3 (2009) 77–88.
[21] B.L, C.C., D.J.S., Comparison of classifier methods: a case study in handwritten digit recognition, in: Proceedings of the 12th IAPR International Conference on IEEE, vol. 2, 1994, pp. 77–82.
[22] B. Scholkopf, C.J. Burges, A.J. Smola, Advances in Kernel Methods: Support Vector Learning, MIT Press, 1999.
[23] Z. Yang, Y. Shao, X. Zhang, Multiple birth support vector machine for multi-class classification, Neural Comput. Appl. (2012) 1–9.
[24] O.L. Mangasarian, Generalized support vector machines, in: Advances in Large Margin Classifiers, 1998, pp. 135–146.
[25] D.G. Luenberger, Y. Ye, Linear and Nonlinear Programming, vol. 116, Springer Science & Business Media, 2008.

Dewei Li received his bachelor degree in mathematics in 2012 and now is a graduate student of information school of Renmin University of China. He is currently interested in SVMs, optimization theory and has published several conference papers.

Yingjie Tian received the First degree in mathematics in 1994, the masters degree in applied mathematics in 1997, and the Ph.D. degree in management science and engineering. He is currently a Professor with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing, China. He has published four books about support vector machines (SVMs), one of which has been cited over 1000 times. His current research interests include SVMs, optimization theory and applications, data mining, intelligentknowledge management, and risk management.