PAC-Bayes Analysis for Twin Support Vector Machines

Xijiong Xie and Shiliang Sun

Abstract— **Twin support vector machines are a powerful learning method for binary classification. Compared to standard support vector machines, they learn two hyperplanes rather than one as in standard support vector machines, and work faster and sometimes perform better than support vector machines. However, relatively little is known about their theoretical performance. As recent tightest bounds for practical applications, PAC-Bayes bounds are based on a prior and posterior over the distribution of classifiers. In this paper, we study twin support vector machines from a theoretical perspective and use the PAC-Bayes bound to measure the generalization error bound of twin support vector machines. Experimental results on real-world datasets show better predictive capabilities of the PAC-Bayes bound for twin support vector machines compared to the PAC-Bayes bound for support vector machines.**

I. Introduction

SUPPORT vector machines (SVMs) [1], [2] have been developed rapidly during recent years. They are a powerful tool for pattern classification and regression which has been applied to a variety of practical problems such as object detection, text categorization, bioinformatics and image classification. They seek the best tradeoff between the model complexity and the learning ability according to the limited example information in order to obtain the best generalization ability. They originate from the idea of structural risk minimization in statistical learning theory and output an optimal hyperplane which is obtained by maximizing the margin between two parallel hyperplanes. The optimization involves the minimization of a quadratic programming (QP) problem. SVMs can also deal with the nonlinear problem by the use of the kernel trick [3].

Recently, the research of nonparallel hyperplane classifiers has been active. A nonparallel hyperplane classifier called generalized eigenvalue proximal SVMs (GEPSVMs) was proposed by Mangasarian and Wild [4] for binary classification. GEPSVMs seek two nonparallel hyperplanes such that each hyperplane is as close as possible to examples from one class and as far as possible to examples from the other class. The two hyperplanes are obtained by eigenvectors corresponding to the smallest eigenvalues of two related generalized eigenvalue problems. Later another nonparallel hyperplane classifier called twin support vector machines (TSVMs) was proposed by Jayadeva et al. [5], which aims to generate two nonparallel hyperplanes such that one of the hyperplanes is closer to one class and has a certain distance to the other class. In SVMs, the QP has all examples in constraints while TSVMs solve a pair of QP problems for which examples of one class give the constraints of the other QP and vice versa, so that its time complexity is about $\frac{1}{4}$ of standard SVMs [6]. Experimental results [5] validated that nonparallel hyperplane classifier TSVMs can indeed improve the performance of traditional SVMs. Researchers also proposed some improved versions of TSVMs such as TBSVMs [7], [8] and CDMTSVMs [9]. The significant advantage of TBSVMs over TSVMs is that the structural risk minimization principle is implemented by introducing a regularization term. The CDMTSVMs using coordinate descent in TSVMs lead to very fast training. Moreover, least squares twin support vector machines [10], weighted least squares twin support vector machines [11], [12] and least squares twin parametric-margin support vector machines [13] have been proposed, which can lead to simple and fast algorithms through replacing inequality constraints with equality constraints. Several works [14], [15], [16] commonly attempted to use the centroid of the class to improve TSVMs, such that the examples of one class are closest to its class centroid while the examples of different classes are separated as far as possible. Robust twin support vector machines [17] and centroid twin support vector machines [18] have been proposed to deal with data with measurement noise. Structural twin support vector machines [19] have been proposed considering structural information of data. Probabilistic outputs for twin support vector machines were also proposed to improve the final classifier [20]. There are also some extensions of TSVMs to other learning frameworks. For example, TSVMs are extended to multitask learning [18], multi-view learning [21] and semi-supervised learning [22]. TSVMs are also extended to solve regression problems, which are called TSVR [23] and the multiclass classification problem by the one-versus-all method [24].

For the classification problem, a good classifier is expected to minimize the generalization error. The VC bounds [2] are generally very loose despite their large influence on our understanding of learning. They only consider the data-dependencies coming through the training error of the classifiers. In fact, there exist VC lower bounds that are asymptotically identical to the corresponding upper bounds. This suggests that significantly tighter bounds can only come through extra data-dependent properties such as the distribution of margins obtained by a classifier on the training dataset [25].

Some early bounds are based on covering number computations, while later bounds have considered Rademacher complexity [26]. Among the data-dependent bounds, the tightest bounds appear to be the PAC-Bayes bound [27].

Xijiong Xie and Shiliang Sun are with Shanghai Key Laboratory of Multidimensional Information Processing, Department of Computer Science and Technology, East China Normal University, 500 Dongchuan Road, Shanghai 200241, P. R. China (email: xjxie11@gmail.com, slsun@cs.ecnu.edu.cn).

This work is supported by the National Natural Science Foundation of China under Projects 61370175, and Shanghai Knowledge Service Platform Project (No. ZF1213).

The PAC-Bayes bound is a basic and very general method for data-dependent theoretical analysis in machine learning [28], [29], [30], [25]. By now, it has been applied in supervised learning, unsupervised learning, reinforcement learning and so on, leading to many algorithms and accompanying generalization bounds. The original PAC-Bayes bound may use a Gaussian prior centered at the origin in the weight space. Then the prior PAC-Bayes bound was proposed, which uses part of the training dataset to compute a more informative prior and compute the bound on the remainder of the examples relative to this prior. Later expectation-prior PAC-Bayes bound [31] was proposed which does not require the existence of the separate dataset. The PAC-Bayes bounds are presented for many famous classification methods like SVMs [27], maximum entropy classifiers [32], Gaussian process classification [33] and so on.

By now, theoretical analysis on twin support vector machines has not been studied. In this paper, we use the PAC-Bayes theory to analyze the generalization error bound of twin support vector machines. After reviewing background knowledge in Section II, we introduce the PAC-Bayes bound for twin support vector machines in Section III. After reporting experimental results in Section IV, we give conclusions and future work in Section V.

II. Background

In this section, we give a brief review of SVMs, TSVMs and the PAC-Bayes bound.

A. SVMs

SVMs have been introduced in the framework of structural risk minimization and are based on the theory of VC bounds [1], [2], [18]. Suppose there are m examples represented by $T = \{(x_1, y_1), ..., (x_m, y_m)\}$. Let $y_i \in \{1, -1\}$ denote the class to which the ith example belongs. First we review the linearly separable case. Classifier parameters $w \in R^d$ and $b \in R$ need to satisfy $y_i(w^\top x_i + b) \geq 1$. The hyperplane described by $w^\top x + b = 0$ lies midway between the bounding hyperplanes given by $w^\top x + b = 1$ and $w^\top x + b = -1$. The margin of separation between the two classes is given by $\frac{2}{\|w\|_2}$, where $\|w\|_2$ represents the L_2 norm of w. Support vectors are those training examples lying on the above two hyperplanes. The standard SVMs are obtained by solving the following optimization problem

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2} w^\top w \\ \text{s.t.} \ & \forall i : y_i(w^\top x_i + b) \geq 1. \end{aligned} \tag{1}$$

The decision function is $f(x) = \text{sign}(w^\top x + b)$. When the two classes are not strictly linearly separable, classifier parameters w and b need to satisfy $y_i(w^\top x_i + b) \geq 1 - \xi_i$. The optimization problem of (1) can be adapted to

$$\begin{aligned} \min_{w,b} \ & \frac{1}{2} w^\top w + c \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \ & \forall i : y_i(w^\top x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \end{aligned} \tag{2}$$

where c is a penalty parameter and ξ_i are the slack variables. The dual optimization problem of (2) is written as

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

where α_i are Lagrangian multipliers. The optimal solution is

$$w = \sum_{i=1}^{m} \alpha_i^* y_i x_i, \ b = \frac{1}{N_s} (y_j - \sum_{i=1}^{N_s} \alpha_i^* y_i (x_i \cdot x_j)), \tag{4}$$

where α^* is the solution of the dual optimization problem (3), and N_s represents the number of support vectors satisfying $0 < \alpha < c$. The decision function is $f(x) = \text{sign}(w^\top x + b)$.

B. TSVMs

Here we introduce TSVMs [5], [18]. Suppose examples belonging to classes 1 and -1 are represented by matrices $\bar{A}$ and $\bar{B}$, and the size of $\bar{A}$ and $\bar{B}$ are $(m_1 \times d)$ and $(m_2 \times d)$, respectively. We define two matrices A, B and four vectors v_1, v_2, e_1, e_2, where e_1 and e_2 are vectors of ones of appropriate dimensions and

$$A = (\bar{A}, e_1), \ B = (\bar{B}, e_2), \ v_1 = \begin{pmatrix} w_1 \\ b_1 \end{pmatrix}, \ v_2 = \begin{pmatrix} w_2 \\ b_2 \end{pmatrix}.$$

TSVMs obtain two nonparallel hyperplanes

$$w_1^\top x + b_1 = 0 \quad \text{and} \quad w_2^\top x + b_2 = 0 \tag{5}$$

around which the examples of the corresponding class are clustered. The classifier is given by solving the following QPs separately
(TSVM1)

$$\begin{aligned} \min_{v_1, q_1} \ & \frac{1}{2} (Av_1)^\top (Av_1) + c_1 e_2^\top q_1 \\ \text{s.t.} \ & (Bv_1) + q_1 \geq e_2, \ q_1 \geq 0, \end{aligned} \tag{6}$$

(TSVM2)

$$\begin{aligned} \min_{v_2, q_2} \ & \frac{1}{2} (Bv_2)^\top (Bv_2) + c_2 e_1^\top q_2 \\ \text{s.t.} \ & (Av_2) + q_2 \geq e_1, \ q_2 \geq 0, \end{aligned} \tag{7}$$

where c_1, c_2 are nonnegative parameters and q_1, q_2 are slack vectors of appropriate dimensions.

The Lagrangian of the problem TSVM1 is given by

$$\begin{aligned} L(v_1, q_1, \alpha, \beta) = & \frac{1}{2} (Av_1)^\top (Av_1) + c_1 e_2^\top q_1 \\ & - \alpha^T (Bv_1 + q_1 - e_2) - \beta^T q_1, \end{aligned} \tag{8}$$

where $\alpha = (\alpha_1, \alpha_2 \cdots, \alpha_{m_2})^\top$, $\beta = (\beta_1, \beta_2 \cdots, \beta_{m_2})^\top$ are the vectors of Lagrange multipliers. The Karush-Kuhn-Tucker (KKT) optimality conditions for (TSVM1) are given

by

$$A^\top A v_1 - B^\top \alpha = 0, \tag{9}$$
$$c_1 e_2 - \alpha - \beta = 0, \tag{10}$$
$$B v_1 + q_1 \geq e_2, q_1 \geq 0, \tag{11}$$
$$\alpha^\top (B v_1 + q_1 - e_2) = 0,\ \beta^\top q_1 = 0, \tag{12}$$
$$\alpha \geq 0,\ \beta \geq 0. \tag{13}$$

Since $\beta \geq 0$, from (10), we have $0 \leq \alpha \leq c_1$. From (9), v_1 can be given by

$$v_1 = (A^\top A)^{-1} B^\top \alpha. \tag{14}$$

To avoid ill-conditioning of $A^\top A$, we use a regularization term ϵI, where $\epsilon > 0$, I is an identity matrix of appropriate dimensions. Therefore, (14) is adapted to

$$v_1 = (A^\top A + \epsilon I)^{-1} B^\top \alpha. \tag{15}$$

Using (8), (14) and the KKT conditions, the dual problem is

$$\begin{aligned} \max_{\alpha}\ & e_2^\top \alpha - \frac{1}{2} \alpha^\top B (A^\top A)^{-1} B^\top \alpha \\ \text{s.t.}\ & 0 \leq \alpha \leq c_1. \end{aligned} \tag{16}$$

Similarly, we consider TSVM2 and obtain its dual as

$$\begin{aligned} \max_{\gamma}\ & e_1^\top \gamma - \frac{1}{2} \gamma^\top A (B^T B)^{-1} A^\top \gamma \\ \text{s.t.}\ & 0 \leq \gamma \leq c_2. \end{aligned} \tag{17}$$

The augmented vector v_2 is given by

$$v_2 = (B^T B)^{-1} A^T \gamma. \tag{18}$$

The label of a new example x is determined by the minimum of $|x^\top w_r + b_r|$ $(r = 1, 2)$ which are the perpendicular distances of x to the two hyperplanes given in (5).

C. PAC-Bayes Bound

Here we briefly review the PAC-Bayes bound theorem [28], [30], [31]. We first state the general PAC-Bayes bound after giving two relevant definitions. Suppose that a distribution D of pattern x lies in a certain input space χ, with their corresponding output labels y $(y \in \{-1, 1\})$. Moreover, let us consider a distribution Q over the classifiers c. For every classifier c, the following two error measures are defined:

Definition 2.1: (True error) The true error c_D of a classifier c is defined as the probability of misclassifying a pair pattern-label (x, y) selected at random from D

$$c_D \equiv Pr_{(x,y)\sim D}(c(x) \neq y) \tag{19}$$

Definition 2.2: (Empirical error) The empirical error $\hat{c}_S$ of a classifier c on a sample S of size m is defined as the error rate on S

$$\hat{c}_S \equiv Pr_{(x,y)\sim S}(c(x) \neq y) = \frac{1}{m} \sum_{i=1}^{m} I(c(x_i) \neq y_i) \tag{20}$$

where $I(\cdot)$ represents an indicator function equal to 1 if the argument is true and equal to 0 if the argument is false.

Two error measures on the distribution of classifiers are defined as $Q_D \equiv E_{c\sim Q} c_D$ (the average true error) which means the probability of misclassifying an instance x chosen uniformly from D with a classifier c chosen according to Q and $\hat{Q}_S \equiv E_{c\sim Q} \hat{c}_S$ (the average empirical error) which means the probability of classifier c chosen according to Q misclassifying an instance x chosen from a sample S.

For these two quantities we can derive the PAC-Bayes bound on the true error of the distribution of classifiers:

Theorem 2.1: (PAC-Bayes bound) For all prior distributions $P(c)$ over the classifiers c, and for any $\delta \in (0, 1]$

$$\begin{aligned} & Pr_{S\sim D^m}\big(\forall Q(c) : KL(\hat{Q}_S \parallel Q_D) \leq \\ & \frac{KL(Q(c)\|P(c)) + \ln(\frac{m+1}{\delta})}{m}\big) \geq 1 - \delta, \end{aligned} \tag{21}$$

where $KL(Q(c)\|P(c)) = E_{c\sim Q} \ln \frac{Q(c)}{P(c)}$ is the Kullback-Leibler divergence, and $KL(p\|q) = q \ln \frac{q}{p} + (1-q) \ln \frac{1-q}{1-p}$.

The proof of the theorem can be found in [28]. This bound can be generalized to the case of linear classifiers. The m training examples define a linear classifier that can be represented by

$$c(x) = \text{sign}(v^\top \phi(x)) \tag{22}$$

where $\phi(x)$ is a nonlinear projection to a certain feature space where the original nonlinear problem can be solved by transforming it to a linear problem, and v is a vector from that feature space that determines the classification hyperplane.

For any vector w $(\|w\| = 1)$, a stochastic classifier is defined. Assume the prior $P(c)$ is a spherical Gaussian with an identity covariance matrix, centered at origin, that is $v \sim N(0, I)$. Simultaneously, assume the posterior $Q(w, u)$ is a spherical Gaussian with an identity covariance matrix, centered along the direction pointed by w at a distance u from the origin, that is $v \sim N(uw, I)$. The generalization performance of the classifier in the form of equation (22) can be bounded as

Corollary 2.1: (PAC-Bayes bound for SVMs) For all distributions D, for all $\delta \in (0, 1]$, we have

$$\begin{aligned} & Pr_{S\sim D^m}\big(\forall w, u : KL(\hat{Q}_S(w, u) \parallel Q_D(w, u)) \leq \\ & \frac{\frac{u^2}{2} + \ln(\frac{m+1}{\delta})}{m}\big) \geq 1 - \delta. \end{aligned} \tag{23}$$

It can be easily proved using a standard expression for the KL divergence between two Gaussians in an N dimensional space,

$$\begin{aligned} & KL(N(u_0, \Sigma_0) \parallel N(u_1, \Sigma_1)) = \frac{1}{2}\big(\ln(\frac{det\Sigma_1}{det\Sigma_0}) + \\ & tr(\Sigma_1^{-1}\Sigma_0) + (u_1 - u_0)^\top \Sigma_1^{-1}(u_1 - u_0) - N\big). \end{aligned} \tag{24}$$

So $KL(N(0, I) \parallel N(uw, I)) = \frac{u^2}{2}$. It can be shown (see [28]) that

$$\hat{Q}_S(w, u) = E_m[\tilde{F}(u\gamma(x, y))] \tag{25}$$

where E_m is the average over the m training examples, $\gamma(x, y)$ is the normalised margin of the training examples

$$\gamma(x, y) = \frac{y w^\top \phi(x)}{\|\phi(x)\| \|w\|} \tag{26}$$

and $\tilde{F} = 1 - F$, where F is the cumulative normal distribution

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx. \tag{27}$$

It is observed above that SVMs are computed by the means of the kernel trick. The generalization error of such a classifier can be bounded by at most twice the average true error $Q_D(w,u)$ of the corresponding stochastic classifier in Corollary 2.1. For all u, we have

$$Pr_{(x,y)\sim D}\big(\text{sign}(w^\top \phi(x)) \neq y\big) \leq 2Q_D(w,u). \tag{28}$$

III. PAC-Bayes Bound for twin support vector machines

TSVMs can improve the performance and time complexity compared to SVMs. However, there does not exist formal theoretical analysis about TSVMs. In this section, we attempt to analyze the PAC-Bayes generalization error bound of TSVMs. At first, we analyze the classifier of TSVMs. In order to analyze the PAC-Bayes bound of TSVMs, we can rewrite the final decision function of TSVMs in this form

$$\begin{aligned} f(x) = \text{sign}\big(&(\frac{w_2^\top}{\|w_2\|}\bar{S}(w_2^\top x + b_2) - \frac{w_1^\top}{\|w_1\|} \\ &\bar{S}(w_1^\top x + b_1))x + (\frac{b_2}{\|w_2\|}\bar{S}(w_2^\top x + b_2) \\ &- \frac{b_1}{\|w_1\|}\bar{S}(w_1^\top x + b_1))\big), \end{aligned} \tag{29}$$

where $\bar{S}(\cdot)$ represents an indicator function equal to 1 if the argument is nonnegative and equal to -1 if the argument is negative. We define $\bar{w} = (\frac{w_2^\top}{\|w_2\|}\bar{S}(w_2^\top x + b_2) - \frac{w_1^\top}{\|w_1\|}\bar{S}(w_1^\top x + b_1))^\top$ and $\bar{b} = \frac{b_2}{\|w_2\|}\bar{S}(w_2^\top x + b_2) - \frac{b_1}{\|w_1\|}\bar{S}(w_1^\top x + b_1)$. Then we can get the final linear classifier

$$f(x) = \text{sign}(\bar{w}^\top x + \bar{b}). \tag{30}$$

The classifier can also be written in the kernelized form

$$c(x) = \text{sign}(v^\top \phi(x)). \tag{31}$$

For any vector w ($\|w\| = 1$), a stochastic classifier is defined. We define prior of classifier $P(c)$ to be a spherical Gaussian with an identity covariance matrix, centered at the origin, that is $v \sim N(0, I)$. Simultaneously, we define posterior $Q(w,u)$ to be a spherical Gaussian with an identity covariance matrix, centered along the direction pointed by w at a distance u from the origin, that is $v \sim N(uw, I)$. Then we present the PAC-Bayes bound for TSVMs.

Corollary 3.1: (PAC-Bayes bound for TSVMs) For all distributions D, for all $\delta \in (0,1]$, we have

$$\begin{aligned} &Pr_{S\sim D^m}\big(\forall w,u : KL(\hat{Q}_S(w,u) \parallel Q_D(w,u)) \leq \\ &\frac{\frac{u^2}{2} + \ln(\frac{m+1}{\delta})}{m}\big) \geq 1 - \delta. \end{aligned} \tag{32}$$

It can be shown that

$$\hat{Q}_S(w,u) = E_m[\tilde{F}(u\gamma(x,y))], \tag{33}$$

where E_m is the average over the m training examples, $\gamma(x,y)$ is the normalised margin of the training examples

$$\gamma(x,y) = \frac{yw^\top \phi(x)}{\|\phi(x)\|\|w\|} \tag{34}$$

and $\tilde{F} = 1 - F$, where F is the cumulative normal distribution

$$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx. \tag{35}$$

The above TSVMs are also computed by the means of the kernel trick. The generalization error of such a classifier can be bounded by at most twice the average true error $Q_D(w,u)$ of the corresponding stochastic classifier in Corollary 3.1. For all u, we have

$$Pr_{(x,y)\sim D}\big(\text{sign}(w^\top \phi(x)) \neq y\big) \leq 2Q_D(w,u). \tag{36}$$

IV. Experimental Results

A. Datasets

In this section, we perform experiments of binary classification problems using real-world datasets. Details about the five datasets are given as follows.

Contraceptive Method Choice (CMC). The dataset comes from UCI Machine Learning Repository. It contains 1473 examples and has nine attributes.

Face Detection. The dataset comes from the MIT CBCL repository. It is a binary classification problem which intends to identify whether a picture is a human face or not. In this experiment, 2000 face and non-face images are used, where half of them are faces and each image is a 19×19 gray picture.

Handwritten Digit Classification. The dataset comes from UCI Machine Learning Repository. The dataset we used here contains 2400 examples of digits 3 and 8 chosen from the MNIST digital images, where half of the data are digit 3 and the image sizes are 28×28.

Pima. The dataset comes from UCI Machine Learning Repository. The dataset contains 768 examples and has eight attributes.

German. The dataset comes from UCI Machine Learning Repository. The dataset contains 1000 examples and has twenty attributes.

B. Experimental Setting

For every dataset, we obtain 10 different training/test set partitions with 80% of the examples forming the training dataset and the remaining 20% forming the test dataset. We then change the training sizes from 20% to 100% of the formed training datasets. We perform experiments with Gaussian RBF kernel. The Gaussian kernel can be written as

$$K(x_i, x_j) = exp(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}), \tag{37}$$

where σ is the width of the Gaussian kernel. The optimal pair (c, σ) of SVMs is sought by grid search to select the best parameters in the region $\{10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000\}$ through a five-fold

TABLE I

PAC-BAYES ERROR BOUNDS (%) AND ACCURACIES (%) ON CMC.

Rate of training dataset	Bound for SVMs	Accuracy for SVMs	Bound for TSVMs	Accuracy for TSVMs
20%	66.01±0.01	64.86±2.35	65.98±0.03	64.00±1.73
40%	61.83±0.00	68.45±1.50	61.83±0.01	65.84±2.62
60%	59.89±0.00	68.99±1.37	59.89±0.00	70.41±1.43
80%	58.69±0.00	71.51±3.79	58.69±0.00	71.02±1.23
100%	57.87±0.00	71.48±2.21	57.87±0.00	73.40±1.30

TABLE II

PAC-BAYES ERROR BOUNDS (%) AND ACCURACIES (%) ON FACE DETECTION.

Rate of training dataset	Bound for SVMs	Accuracy for SVMs	Bound for TSVMs	Accuracy for TSVMs
20%	62.52±0.01	96.49±1.12	62.35±0.38	95.78±0.42
40%	59.21±0.00	98.57±0.42	58.91±0.27	98.15±0.41
60%	57.68±0.00	99.36±0.25	57.52±0.18	99.02±0.27
80%	56.74±0.00	99.65±0.13	56.51±0.08	99.62±0.19
100%	56.09±0.00	99.88±0.12	55.97±0.11	99.85±0.12

TABLE III

PAC-BAYES ERROR BOUNDS (%) AND ACCURACIES (%) ON HANDWRITTEN DIGIT CLASSIFICATION.

Rate of training dataset	Bound for SVMs	Accuracy for SVMs	Bound for TSVMs	Accuracy for TSVMs
20%	61.55±0.01	96.30±0.38	61.46±0.06	95.94±0.81
40%	58.49±0.00	97.79±0.26	58.47±0.01	97.41±0.40
60%	57.07±0.00	98.53±0.27	57.07±0.01	98.46±0.25
80%	56.21±0.00	99.34±0.25	56.21±0.00	99.16±0.26
100%	55.61±0.00	99.38±0.15	55.61±0.00	99.56±0.09

TABLE IV

PAC-BAYES ERROR BOUNDS (%) AND ACCURACIES (%) ON PIMA.

Rate of training dataset	Bound for SVMs	Accuracy for SVMs	Bound for TSVMs	Accuracy for TSVMs
20%	68.89±0.01	73.13±2.87	68.81±0.08	66.50±2.24
40%	64.06±0.00	75.41±2.15	64.05±0.00	73.29±1.53
60%	61.79±0.00	76.19±1.02	61.79±0.00	79.41±1.04
80%	60.38±0.00	77.31±0.93	60.38±0.00	84.95±1.23
100%	59.40±0.00	77.23±0.70	59.39±0.00	88.32±2.38

TABLE V

PAC-BAYES ERROR BOUNDS (%) AND ACCURACIES (%) ON GERMAN.

Rate of training dataset	Bound for SVMs	Accuracy for SVMs	Bound for TSVMs	Accuracy for TSVMs
20%	66.91±0.00	73.54±1.68	66.87±0.07	70.41±2.57
40%	62.54±0.00	75.79±2.29	62.52±0.04	74.26±1.75
60%	60.49±0.00	77.94±1.71	60.47±0.03	78.25±2.53
80%	59.22±0.00	79.22±0.75	59.22±0.00	81.05±1.65
100%	58.35±0.00	80.17±0.62	58.34±0.00	83.34±1.89

cross-validation. The optimal pair (c_1, c_2, σ) of TSVMs is also sought by grid search to select the best parameters in the region $\{10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000\}$ through a five-fold cross-validation. In the experiments, we set $\delta = 0.01$.

C. Experimental Results and Analysis

We show experimental results which compare PAC-Bayes bounds (Q_D) for TSVMs with PAC-Bayes bounds for SVMs. The test accuracies and PAC-Bayes bounds for SVMs and TSVMs are averaged for 10 times. We complete the average with the standard deviation. The results are shown in Tables I, II, III, IV, V.

From the experimental results, we can find that as the rate of training dataset increases, the bounds for SVMs and TSVMs become tighter. In Tables II, IV, V, the bounds for TSVMs are almost tighter than the bounds for SVMs. In Tables I, III, the bounds for SVMs and TSVMs are nearly the same. We can also conclude that when the rate of training dataset is low, the performance of TSVMs is not better than SVMs. When the rate of training dataset is high, the performance of TSVMs is better than or close to SVMs. In summary, the experimental results verify the good predictive capabilities of the PAC-Bayes bound for twin support vector machines.

V. Conclusion and Future work

Many practical applications and extended algorithms for twin support vector machines have been proposed. However, there are no exist theoretical justifications on twin support vector machines. In this paper, we use the PAC-Bayes bound to analyze the generalization error bound of twin support vector machines. Comparative experiments on real-world datasets verify the better predictive capabilities of the PAC-Bayes bound for twin support vector machines. In the future, we can use the informative prior inspired by [31] to tighten the bound.

References

[1] J. Shawe-Taylor, S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, pp. 3609-3618, 2011.

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

[3] B. Scholkopf, A. Smola, *Learning with Kernels*, Cambridge: MIT Press, 2003.

[4] O.L. Mangasarian, E.W. Wild, "MultisurFace proximal support vector machine classification via generalized eigenvalues," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 69-74, 2006.

[5] R. Jayadeva, S. Khemchandani, Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 74, pp. 905-910, 2007.

[6] S. Ghorai, Mukherjee, P.K. Dutta, "Nonparallel plane proximal classifier," *Signal Processing*, vol. 89, pp. 510-522, 2009.

[7] Y. Shao, C. Zhang, X. Wang, N. Deng, "Improvements on twin support vector machines," *IEEE Transactions on Neural Networks*, vol. 22, pp. 962-968, 2011.

[8] S. Ding, Y. Zhao, B. Qi, H. Huang, "An overview on twin support vector machines," *Artificial Intelligence Review*, 2012.

[9] Y.H. Shao, N.Y. Deng, "A coordinate descent margin based-twin support vector machine for classification," *Neural Networks*, vol. 25, pp. 114-121, 2012.

[10] M.A. Kumar, M. Gopal, "Least squares twin support vector machines for pattern classification," *Expert Systems with Applications*, vol. 36, pp. 7535-7543, 2009.

[11] J. Chen, "Weighted least squares twin support vector machines for pattern classification," *Proceedings of the 2nd International Conference on Computer and Automation Engineering*, pp. 242-246, 2010.

[12] Y. Xu, X. Lv, Z. Wang, L. Wang "A weighted least squares twin support vector machine," *Journal of Information Science and Engineering*, vol. 30, pp. 1773-1787, 2014.

[13] Y.H. Shao, Z. Wang, W.J. Chen, N.Y. Deng, "Least squares twin parametric-margin support vector machines for classification," *Applied Intelligence*, vol. 39, pp. 451-464, 2013.

[14] Y.H. Shao, N.Y. Deng, Z.M. Yang, "Least squares recursive projection twin support vector machine for classification," *Pattern Recognition*, vol. 45, pp. 2299-2307, 2012.

[15] Y.H. Shao, Z. Wang, W.J. Chen, N.Y. Deng, "A regularization for the projection twin support vector machine," *Knowledge-Based Systems*, vol. 37, pp. 203-210, 2013.

[16] X. Chen, J. Yang, Q. Ye, J. Liang, "Recursive projection twin support vector machine via within-class variance minimization," *Pattern Recognition*, vol. 44, pp. 2643-2655, 2011.

[17] Z. Qi, Y. Tian, Y. Shi, "Robust twin support vector machine for pattern classification," *Pattern Recognition*, vol. 46, pp. 305-316, 2014.

[18] X. Xie, S. Sun, "Multitask centroid twin support vector machines," *Neurocomputing*, vol. 149, pp. 1085-1091, 2015.

[19] Z. Qi, Y. Tian, Y. Shi, "Structural twin support vector machine for classification," *Knowledge-Based Systems*, vol. 43, pp. 74-81, 2013.

[20] Y.H. Shao, Y. Tian, Y. Shi, "Probabilistic outputs for twin support vector machines," *Knowledge-Based Systems*, vol. 33, pp. 145-151, 2012.

[21] X. Xie, S. Sun, "Multi-view Laplacian twin support vector machines," *Applied Intelligence*, vol. 41, pp. 1059-1068, 2014.

[22] Z. Qi, Y. Tian, Y. Shi, "Laplacian twin support vector machine for semi-supervised classification," *Neural Networks*, vol. 35, pp. 46-53, 2012.

[23] X. Peng, "TSVR: An efficient twin support vector machine for regression," *Neural Networks*, vol. 23, pp. 365-372, 2010.

[24] J. Xie, K. Hone, W. Xie, X. Gao, Y. Shi, X. Liu, "Extending twin support vector machine classifier for multi-category classification problems,"

[25] P. Germain, A. Lacasse, F. Laviolette, M. Marchand, "PAC-Bayesian learning of linear classifiers," *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 353-360, 2009. *Intelligent Data Analysis*, vol. 17, pp. 649-664, 2013.

[26] P. Bartlett, S. Mendelson, "Rademacher and Gaussian complexities: risk bounds and structural results," *Journal of Machine Learning Research*, vol. 3, pp. 463-482, 2002.

[27] A. Ambroladze, E. Parrado-Hernández, J. Shawe-Taylor, "Tighter PAC-Bayes bounds," *Proceeding of the 2006 Conference on Neural Information Processing Systes*, vol. 41, pp. 9-16, 2007.

[28] J. Langford, "Tutorial on practical prediction theory for classification," *Jounal of Machine Learning Research*, vol. 6, pp. 273-306, 2005.

[29] J. Langford, J. Shawe-Taylor, "PAC-Bayes & Margins," *Advances in Neural Information Processing Systems*, vol. 14, pp. 423-430, 2002.

[30] G. Lever, F. Laviolette, J. Shawe-Taylor, "Tighter PAC-Bayes bounds through distribution-dependent priors," *Theoretical Computer Science*, vol. 473, pp. 4-28, 2013.

[31] E. Parrado-Hernández, A. Ambroladze, J. Shawe-Taylor, S. Sun, "PAC-Bayes bounds with data dependent priors," *Journal of Machine Learning Research*, vol. 13, pp. 3507-3531, 2012.

[32] J. Shawe-Taylor, D. R. Hardoon, "PAC-Bayes analysis of maximum entropy learning," *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, pp. 480-487, 2009.

[33] M. Seeger, "PAC-Bayesian generalisation error bounds for Gaussian process classification," *Journal of Machine Learning Research*, vol. 3, pp. 233-269, 2002.