Twin Bounded Large Margin Distribution Machine

Haitao Xu[1], Brendan McCane[1], and Lech Szymanski[1]

Department of Computer Science, University of Otago, Otago, 9016, New Zealand
{haitao,mccane,lechszym}@cs.otago.ac.nz

Abstract. In order to speed up the learning time of large margin distribution machine (LDM) and improve the generalization performance of twin bounded support vector machine (TBSVM), a novel method named twin bounded large margin distribution machine (TBLDM) is proposed in this paper. The central idea of TBLDM is to seek a pair of nonparallel hyperplanes by optimizing the positive and negative margin distributions on the base of TBSVM. The experimental results indicate that the proposed TBLDM is a fast, effective and robust classifier.

Keywords: Large margin distribution machine· twin bounded support vector machine· margin distribution· margin mean· margin variance.

1 Introduction

Support vector machines (SVMs) [18] [3] are powerful tools for pattern classification and regression. For the classical binary classification SVM, the optimal hyperplane can be obtained by maximizing a relaxed minimum margin, i.e., the smallest distance from data point to the classification boundary. This optimisation can be expressed as a quadratic programming problem (QPP). Margin theory [17] provides good theoretical support to the generalisation performance of SVMs and it has also been applied to many other machine learning approaches, such as AdaBoost [5]. There was, however, a long debate on whether margin theory plays a significant role in AdaBoost [14, 2]. It had been believed that a single-data-point margin such as minimum margin is not crucial [13, 19]. Gao and Zhou [6] ended the long debate and showed that margin distribution, characterized by margin mean and variance, is critical for generalisation in boosting. Inspired by these results, Zhang and Zhou [23] first focused on the influence of the margin distribution for SVMs and proposed large margin distribution machine (LDM). The margin distribution heuristic can also be applied to clustering [24] and dimensionality reduction [9].

The twin support vector machine (TWSVM) proposed by Jayadeva et al. [7] seeks for two nonparallel boundary hyperplanes and attempts to make each of the two hyperplanes close to one class and far from the other as much as possible. TWSVM solves two smaller size QPPs instead of a single large QPP. This results in TWSVM being faster than SVM. An improved version of TWSVM,

called twin bounded support vector machine (TBSVM) was proposed by Shao et al. [16]. TBSVM implemented the structural risk minimisation principle by introducing a regularization term. Based on statistical learning theory, TBSVM can improve the performance of classification of TWSVM. Recently, many extensions of TWSVM have been proposed, for details, see [15, 8, 12, 21, 20].

In this paper, we propose the twin bounded large margin distribution machine (TBLDM). Similar to LDM, the margin distribution of TBLDM is characterised by first and second order statistics and optimizing the margin distribution is realized by maximizing the margin mean and minimizing the margin variance simultaneously. However, TBLDM tries to optimise the positive and negative margin distributions separately. This is different from LDM, which optimised the whole margin distribution for all training points.

To begin with, we will first provide a brief background on SVM, TWSVM and LDM in Section 2. Our novel approach TBLDM for classification problems will be introduced in Section 3. In Section 4, we will make numerical experiments to verify that our new model is very effective in classification. Discussions and conclusions will be summarized in Section 5.

2 Notation and related work

Given the dataset $T = \{(x_i, y_i)\}_{i=1}^{l}$, where $x_i \in R^n$ is the i-th input sample and $y_i \in \{\pm 1\}$ is the class label of x_i. Let l_1 and l_2 be the numbers of samples belonging to the positive and negative classes, respectively, such that $l = l_1 + l_2$. Denote $X = [x_1, \cdots, x_l] \in R^{n\times l}, A = [x_1^+, \cdots, x_{l_1}^+] \in R^{n\times l_1}$ and $B = [x_1^-, \cdots, x_{l_2}^-] \in R^{n\times l_2}$ as the entire, positive and negative sample matrices. Let $k : R^n \times R^n \to R$ be a kernel function with reproducing kernel Hilbert space (RKHS) $\widetilde{H}$ and nonlinear feature mapping $\phi : R^n \to \widetilde{H}$. Denote $\phi(A) = [\phi(x_1^+), \cdots, \phi(x_{l_1}^+)], \phi(B) = [\phi(x_1^-), \cdots, \phi(x_{l_2}^-)]$ as the positive and negative mapped sample matrices, the kernel matrix $K = \phi(X)^T\phi(X)$ where $\phi(X) = [\phi(x_1), \cdots, \phi(x_l)]$, $K_A = \phi(A)^T\phi(X) \in R^{l_1\times l}, K_B = \phi(B)^T\phi(X) \in R^{l_2\times l}, K(x, X) = [k(x, x_1), \cdots, k(x, x_l)] \in R^{1\times l}$, $\forall x \in R^n$. and $y = (y_1, \cdots, y_l)^T \in R^l$. $y_A = (y_1^+, \cdots, y_{l_1}^+)^T \in R^{l_1}, y_B = (y_1^-, \cdots, y_{l_2}^-)^T \in R^{l_2}$.

2.1 Support vector machine (SVM)

SVM tries to find a hyperplane $f(x) = w^T\phi(x) = 0$, where f is linear and $w \in \widetilde{H}$ is a linear predictor. According to [3] and [17], the margin of the individual sample (x_i, y_i) is defined as

$$\gamma_i = y_i w^T \phi(x_i), i = 1, \cdots, l. \tag{1}$$

In separable cases, all the γ_i will be non-negative. So we can get the geometric distance from each x_i to $w^T\phi(x) = 0$ by scaling each γ_i with $1/\|w\|$:

$$\hat{\gamma}_i = y_i \frac{w^T}{\|w\|}\phi(x_i), i = 1, \cdots, l.$$

For the separable case, SVM maximizes the minimum distance:

$$\begin{aligned} &\max_{w} \hat{\gamma} \\ &s.t.\ \hat{\gamma}_i \geq \hat{\gamma}, i = 1, \cdots, l. \end{aligned}$$

It can be written as

$$\begin{aligned} &\max_{w} \frac{\gamma}{\|w\|} \\ &s.t.\ \gamma_i \geq \gamma, i = 1, \cdots, l. \end{aligned}$$

We can simply set γ as 1 since it doesn't have influence on the optimization. Note that maximizing $1/\|w\|$ is equivalent to minimizing $\|w\|^2$, we can get the classic formulation of hard-margin SVM as follows:

$$\begin{aligned} &\min_{w} \frac{1}{2}\|w\|^2 \\ &s.t.\ y_i w^T \phi(x_i) \geq 1, i = 1, \cdots, l. \end{aligned}$$

For non-separable case, SVM can be written as

$$\begin{aligned} &\max_{w,\xi_i} \gamma_0 - \bar{C} \sum_{i=1}^{l} \xi_i \\ &s.t.\ \gamma_i \geq \gamma_0 - \xi_i, \\ &\quad \xi_i \geq 0, i = 1, \cdots, l, \end{aligned}$$

where γ_0 is a relaxed minimum margin, ξ_i is slack variable and $\bar{C}$ is the trading-off parameter. The above formula can be rewritten as

$$\begin{aligned} &\min_{w,\xi_i} \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} \xi_i \\ &s.t.\ y_i w^T \phi(x_i) \geq 1 - \xi_i, \\ &\quad \xi_i \geq 0, i = 1, \cdots, l, \end{aligned}$$

where C is a trading-off parameter. We can see that SVMs for both separable and non-separable cases consider only single-data-point margins but not the whole margin distribution.

2.2 Twin bounded support vector machine (TBSVM)

Different from conventional SVM, TWSVM seeks for a pair of nonparallel hyperplanes $f_+(x) = w_+^T \phi(x) = 0$ and $f_-(x) = w_-^T \phi(x) = 0$. As an improved version of TWSVM, TBSVM consider the structural risk minimization principle by adding a regularization term. The training time of TBSVM is approximately four times faster than SVM. We introduce non-linear TBSVM in this subsection,

for linear case and other details, see [7, 16]. The unknown vectors $w_+, w_- \in R^n$ of TBSVM can be obtained by solving the following two QPPs:

$$\begin{aligned} &\min_{w_+,\xi_2} \frac{c_1}{2}||w_+||^2 + \frac{1}{2}||\phi(A)^T w_+||^2 + c_3 e_2^T \xi_2 \\ &s.t.\ \ -\phi(B)^T w_+ + \xi_2 \geq e_2,\ \xi_2 \geq 0, \end{aligned} \tag{2}$$

$$\begin{aligned} &\min_{w_-,\xi_1} \frac{c_2}{2}||w_-||^2 + \frac{1}{2}||\phi(B)^T w_-||^2 + c_4 e_1^T \xi_1 \\ &s.t.\ \ \phi(A)^T w_- + \xi_1 \geq e_1,\ \xi_1 \geq 0, \end{aligned} \tag{3}$$

where $c_1, \cdots, c_4 > 0$ are trade-off parameters, $\xi_1 \in R^{l_1}, \xi_2 \in R^{l_2}$ are slack variable vectors and $e_1 \in R^{l_1}, e_2 \in R^{l_2}$ are vectors of ones. A new input $\widetilde{x} \in R^n$ is assigned the class k depending on which of the two hyperplanes it is closer to. That is, the class label $y_{\widetilde{x}}$ can be obtained by $y_{\widetilde{x}} = \arg\min_{k=\pm} \frac{|f_k(\widetilde{x})|}{||w_k||}$.

Similar to the definition of the margin of individual sample in (1), the positive and negative margin of individual sample can be formulated as

$$\gamma_j^+ = y_j^+ f_-(x_j^+) = y_j^+ w_-^T \phi(x_j^+), j = 1, \cdots, l_1, \tag{4}$$

$$\gamma_j^- = y_j^- f_+(x_j^-) = y_j^- w_+^T \phi(x_j^-), j = 1, \cdots, l_2, \tag{5}$$

respectively. We can see that TBSVM tries to maximize the minimal negative margin between the negative samples and positive decision hyperplane by (2) and maximize the minimal positive margin by (3).

2.3 Large margin distribution machine (LDM)

LDM tries to achieve a strong generalization performance by optimizing the margin distribution of samples on the basis of soft-margin SVM. The margin distribution is characterized by first- and second-order statistics. Optimizing margin distribution is realized by maximizing the margin mean and minimizing the margin variance simultaneously. Based on (1), the margin mean $\bar{\gamma}$ and the margin variance $\hat{\gamma}$ can be calculated by $\bar{\gamma} = \frac{1}{l}\sum_{i=1}^{l} \gamma_i$ and $\hat{\gamma} = \frac{1}{l}\sum_{i=1}^{l} (\gamma_i - \bar{\gamma})^2$. The unknown $w \in \widetilde{H}$ can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min_{w,\xi_i} \frac{1}{2} w^T w + \lambda_1 \hat{\gamma} - \lambda_2 \bar{\gamma} + C \sum_{i=1}^{l} \xi_i \\ &s.t.\ y_i w^T \phi(x_i) \geq 1 - \xi_i,\ \xi_i \geq 0, i = 1, \cdots, l, \end{aligned}$$

where $\lambda_1, \lambda_2 > 0$ are the parameters for trading-off the margin variance, the margin mean and the model complexity. It is obvious that LDM can be reduced to soft-margin SVM when $\lambda_1 = \lambda_2 = 0$.

3 Twin bounded large margin distribution machine (TBLDM)

In this section, we will introduce our novel classification method named as twin bounded large margin distribution machine (TBLDM). Based on the concepts of positive margin and negative margin in (4) and (5), the positive margin mean $\bar{\gamma}^+$ and the positive margin variance $\hat{\gamma}^+$ can be calculated by $\bar{\gamma}^+ = \frac{1}{l_1}\sum_{j=1}^{l_1}\gamma_j^+ = \frac{1}{l_1}y_A^T\phi(A)^T w_-$, and $\hat{\gamma}^+ = \frac{1}{l_1}\sum_{i=1}^{l_1}(\gamma_i^+ - \bar{\gamma}^+)^2 = w_-^T\phi(A)Q_1\phi(A)^T w_-$ respectively. Here $Q_1 = \frac{l_1 I_{l_1} - y_A y_A^T}{l_1^2}$ is a symmetric matrix. Since $Q_1^2 = \frac{1}{l_1}Q_1$, it can be concluded that Q_1 is a symmetric nonnegative definite matrix. Similarly, we can get the negative margin mean $\bar{\gamma}^-$ and the negative margin variance $\hat{\gamma}^-$ by $\bar{\gamma}^- = \frac{1}{l_2}y_B^T\phi(B)^T w_+, \hat{\gamma}^- = w_+^T\phi(B)Q_2\phi(B)^T w_+$, where $Q_2 = \frac{l_2 I_{l_2} - y_B y_B^T}{l_2^2}$ is also a symmetric nonnegative definite matrix.

3.1 TBLDM

Specifically, TBLDM seeks a pair of unknown vectors $w_+, w_- \in \widetilde{H}$ by maximizing the positive and negative margin mean and minimizing the positive and negative margin variance simultaneously, that is, by considering the following two optimization problems:

$$\begin{aligned} \min_{w_+,\xi_2} \ & \frac{c_1}{2}\|w_+\|^2 + \frac{1}{2}\|\phi(A)^T w_+\|^2 - \lambda_1\bar{\gamma}^- + \lambda_2\hat{\gamma}^- + c_3 e_2^T\xi_2 \\ s.t. \ & -\phi(B)^T w_+ + \xi_2 \geq e_2, \ \xi_2 \geq 0, \end{aligned} \tag{6}$$

$$\begin{aligned} \min_{w_-,\xi_1} \ & \frac{c_2}{2}\|w_-\|^2 + \frac{1}{2}\|\phi(B)^T w_-\|^2 - \lambda_3\bar{\gamma}^+ + \lambda_4\hat{\gamma}^+ + c_4 e_1^T\xi_1 \\ s.t. \ & \phi(A)^T w_- + \xi_1 \geq e_1, \ \xi_1 \geq 0, \end{aligned} \tag{7}$$

where $\lambda_1, \cdots, \lambda_4 > 0$ are the parameters for trading-off the margin variances, the margin means and the complexity of models. It is obvious that TBLDM can be reduced to the nonlinear TBSVM when $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are equal to 0. Substituting $\bar{\gamma}^-$ and $\hat{\gamma}^-$ into the models (6), we can get the following:

$$\begin{aligned} \min_{w_+,\xi_2} \ & \frac{c_1}{2}\|w_+\|^2 + \frac{1}{2}\|\phi(A)^T w_+\|^2 - \frac{\lambda_1}{l_2}y_B^T\phi(B)^T w_+ + \lambda_2 w_+^T\phi(B)Q_2\phi(B)^T w_+ + c_3 e_2^T\xi_2 \\ s.t. \ & -\phi(B)^T w_+ + \xi_2 \geq e_2, \ \xi_2 \geq 0, \end{aligned} \tag{8}$$

Due to $\widetilde{H} = \text{span}\{\phi(x_1), \cdots, \phi(x_l)\}$, we can let $w_+ = \phi(X)\beta_1$ and $w_- = \phi(X)\beta_2$, where $\beta_1, \beta_2 \in R^l$ are coefficient vectors, and then we can deduce that

$$\begin{aligned} \|w_+\|^2 = \beta_1^T K\beta_1, \qquad & |w_-\|^2 = \beta_2^T K\beta_2, \\ \phi(A)^T w_+ = K_A\beta_1, \qquad & \phi(B)^T w_+ = K_B\beta_1, \\ \phi(A)^T w_- = K_A\beta_2, \qquad & \phi(B)^T w_- = K_B\beta_2, \\ f_+(x) = w_+^T\phi(x) = K(x,X)\beta_1, \ & f_-(x) = w_-^T\phi(x) = K(x,X)\beta_2. \end{aligned} \tag{9}$$

Substituting (9) into the models (8), we have

$$\begin{aligned}\min_{\beta_1,\xi_2}\ & \frac{c_1}{2}\beta_1^T K\beta_1+\frac{1}{2}\beta_1^T K_A^T K_A\beta_1-\frac{\lambda_1}{l_2}y_B^T K_B\beta_1+\lambda_2\beta_1^T K_B^T Q_2 K_B\beta_1+c_3e_2^T\xi_2\\ s.t.\ & -K_B\beta_1+\xi_2\geq e_2,\xi_2\geq 0,\end{aligned}\tag{10}$$

Let

$$\begin{aligned}G_1&=c_1K+K_A^TK_A+2\lambda_2K_B^TQ_2K_B\in R^{l\times l},\\ G_2&=c_2K+K_B^TK_B+2\lambda_4K_A^TQ_1K_A\in R^{l\times l}.\end{aligned}$$

Obviously, G_1 and G_2 are symmetric nonnegative definite matrices. The models (10) can be rewritten as

$$\begin{aligned}\min_{\beta_1,\xi_2}\ & \frac{1}{2}\beta_1^T G_1\beta_1-\frac{\lambda_1}{l_2}y_B^T K_B\beta_1+c_3e_2^T\xi_2\\ s.t.\ & -K_B\beta_1+\xi_2\geq e_2,\xi_2\geq 0,\end{aligned}\tag{11}$$

Considering the Lagrangian function of the model (11)

$$L_1(\beta_1,\xi_2,\alpha_1,\delta_1)=\frac{1}{2}\beta_1^TG_1\beta_1-\frac{\lambda_1}{l_2}y_B^TK_B\beta_1+c_3e_2^T\xi_2-\alpha_1^T(-K_B\beta_1+\xi_2-e_2)-\delta_1^T\xi_2,$$

where $\alpha_1,\delta_1\in R^{l_2}$ are nonnegative Lagrangian multipliers vectors, and letting $\partial L_1/\partial\beta_1=\partial L_1/\partial\xi_2=0$, we get

$$\begin{aligned}G_1\beta_1&=\frac{\lambda_1}{l_2}K_B^Ty_B-K_B^T\alpha_1,\\ c_3e_2-\alpha_1-\delta_1=0&\Rightarrow 0\leq\alpha_1\leq c_3e_2.\end{aligned}\tag{12}$$

Without loss of generality, we can assume that G_1 is an invertible matrix; otherwise, it can be regularized, that is, it can be replaced by the matrix $G_1+t_1I_l$, where $t_1>0$ is a small positive number called regularized coefficient. Consequently, it can be deduced from (12) that

$$\beta_1=G_1^{-1}(\frac{\lambda_1}{l_2}K_B^Ty_B-K_B^T\alpha_1).\tag{13}$$

Submitting (13) and (12) into the Lagrangian function, we can obtain the Wolfe dual form of the model (11):

$$\begin{aligned}\min_{\alpha_1}\ & \frac{1}{2}\alpha_1^TH_1\alpha_1-(\frac{\lambda_1}{l_2}H_1y_B+e_2)^T\alpha_1\\ s.t.\ & 0\leq\alpha_1\leq c_3e_2,\end{aligned}\tag{14}$$

where $H_1=K_BG_1^{-1}K_B^T$. Similarly, we can get

$$\beta_2=G_2^{-1}(\frac{\lambda_1}{l_1}K_A^Ty_A+K_A^T\alpha_2),\tag{15}$$

and then the Wolfe dual form of the model (7) is:

$$\begin{aligned} &\min_{\alpha_2} \frac{1}{2}\alpha_2^T H_2 \alpha_2 + (\frac{\lambda_3}{l_1} H_2 y_A - e_1)^T \alpha_2 \\ &s.t.\ 0 \leq \alpha_2 \leq c_4 e_1, \end{aligned} \tag{16}$$

where $\alpha_2 \in R^{l_1}$ is a nonnegative Lagrangian multipliers vector and $H_2 = K_A G_2^{-1} K_A^T$. A new input $\widetilde{x} \in R^n$ is assigned the class i ($i = 1, 2$ denotes the positive and negative classes, respectively) depending on which of the two hyperplanes is closer to, that is, label $(\widetilde{x}) = \arg\min_{i=1,2} \frac{|K(\widetilde{x}, X)\beta_i|}{\sqrt{\beta_i K \beta_i}}$. The specific procedure is listed in Algorithm 1.

Algorithm 1 TBLDM

Input: Training set T, testing sample $\widetilde{x}$, kernel function $k : R^n \times R^n \to R$, model parameters $\lambda_i, \cdots, \lambda_4$ and $c_i, \cdots, c_4$, regularized parameters t_1, t_2 and kernel parameters;

1: Solve the QPP (14) and obtain the optimal solution α_1^*;
2: Compute β_1^* by (13) with $\alpha_1 = \alpha_1^*$;
3: Solve the QPP (16) and obtain the optimal solution α_2^*;
4: Compute β_2^* by (15) with $\alpha_2 = \alpha_2^*$;
5: For $\widetilde{x}$, predict its label by label $(\widetilde{x}) = \arg\min_{i=1,2} \frac{|K(\widetilde{x}, X)\beta_i^*|}{\sqrt{\beta_i^* K \beta_i^*}}$.

3.2 TBLDM for large scale datasets

It can be seen that we need to compute G_1^{-1} and G_2^{-1} and kernel matrix K, K_A, K_B before solving the dual problems (14) and (16). This is infeasible when the number of samples is significantly large both in terms of memory and computation. To effectively handle large scale problems, in this subsection, we first choose a kernel approximation method, Nyström method [22] to explicitly map features onto subspaces in the RKHS. In this case, the embedding features are obtained without constructing the complete kernel matrix for the data set. Given the kernel-specific embedding, we perform linear TBLDM. Because the inverse matrices of AA^T and BB^T still need to be computed to get the dual problem of linear TBLDM, we solve the primal problem of linear TBLDM here with stochastic gradient descent (SGD) algorithm.

Linear TBLDM is a special case of TBLDM with linear kernel function $k(u, v) = \langle u, v \rangle$ for any $u, v \in R^n$. In this case, the models (6) and (7) are reduced into the following two QPPs:

$$\begin{aligned} &\min_{w_+, \xi_2} \frac{c_1}{2}\|w_+\|^2 + \frac{1}{2}\|A^T w_+\|^2 - \frac{\lambda_1}{l_2} y_B^T B^T w_+ + \lambda_2 w_+^T B Q_2 B^T w_+ + c_3 e_2^T \xi_2 \\ &s.t.\ -B^T w_+ + \xi_2 \geq e_2,\ \xi_2 \geq 0, \end{aligned} \tag{17}$$

$$\min_{w_-,\xi_1} \frac{c_2}{2}\|w_-\|^2 + \frac{1}{2}\|B^T w_-\|^2 - \frac{\lambda_3}{l_1} y_A^T A^T w_- + \lambda_4 w_-^T A Q_1 A^T w_- + c_4 e_1^T \xi_1$$
$$s.t.\ \ A^T w_- + \xi_1 \geq e_1,\ \xi_1 \geq 0. \tag{18}$$

To solve formulas (17) and (18) in primal case, we express them equivalently as two unconstraint optimization problems:

$$\min_{w_+,\xi_2} g_1(w_+) = \frac{c_1}{2}\|w_+\|^2 + \frac{1}{2}\|A^T w_+\|^2 - \frac{\lambda_1}{l_2} y_B^T B^T w_+ + \lambda_2 w_+^T B Q_2 B^T w_+ + c_3 \sum_{i=1}^{l_2} \max\{0, 1 + w_+^T x_i^-\}, \tag{19}$$

$$\min_{w_-,\xi_1} g_2(w_-) = \frac{c_2}{2}\|w_-\|^2 + \frac{1}{2}\|B^T w_-\|^2 - \frac{\lambda_3}{l_1} y_A^T A^T w_- + \lambda_4 w_-^T A Q_1 A^T w_- + c_4 \sum_{i=1}^{l_1} \max\{0, 1 - w_-^T x_i^+\}. \tag{20}$$

If examples (x_i^+, y_i^+), (x_j^+, y_j^+), (x_k^+, y_k^+) are randomly sampled from the positive training set and (x_i^-, y_i^-), (x_j^-, y_j^-), (x_k^-, y_k^-) are randomly sampled from the negative training set independently, it is straightforward to prove that

$$\nabla g_1(w_+, x_i^+, x_j^-, x_k^-) = c_1 w_+ + l_1 x_i^+ {x_i^+}^T w_+ + 2\lambda_2 x_j^- {x_j^-}^T w_+ - 2\lambda_2 x_j^- {x_k^-}^T w_+ + \lambda_1 x_j^- + c_3 l_2 x_j^- \mathbb{I}(j \in I_1), \tag{21}$$

$$\nabla g_2(w_-, x_i^-, x_j^+, x_k^+) = c_2 w_- + l_2 x_i^- {x_i^-}^T w_- + 2\lambda_4 x_j^+ {x_j^+}^T w_- - 2\lambda_4 x_j^+ {x_k^+}^T w_- - \lambda_3 x_j^+ - c_4 l_1 x_j^+ \mathbb{I}(j \in I_2). \tag{22}$$

are the unbiased estimation of $\nabla g_1(w_+)$ and $\nabla g_2(w_-)$ respectively. $\mathbb{I}(\cdot)$ is the indicator function that returns 1 when the argument holds, and 0 otherwise. I_1, I_2 are the index sets defined as $I_1 = \{j | w_+^T x_j^- > -1\}, I_2 = \{j | w_-^T x_j^+ < 1\}$. So we can update w_+, w_- by $w_+ \leftarrow w_+ - r_1 \nabla g_1(w_+, x_i^+, x_j^-, x_k^-)$ and $w_- \leftarrow w_- - r_2 \nabla g_2(w_-, x_i^-, x_j^+, x_k^+)$, r_1, r_2 are learning rates for each iteration of SGD algorithm. The detailed procedure is listed in Algorithm 2.

4 Experiments and results analysis

In order to demonstrate the effectiveness of TBLDM, a series of comparative experiments with SVM, TBSVM and LDM are performed. The experiments focus on the aspects of classification accuracy and computational time on sixteen regular scale datasets and four large-scale datasets. These datasets are taken

Algorithm 2 Nyström + linear TBLDM for large scale problems

Input: Positive training set A, negative training set B, testing sample $\widetilde{x}$, model parameters $\lambda_1, \cdots, \lambda_4$ and $c_1, \cdots, c_4$ and learning rates r_1, r_2;

Get data embedding A_e, B_e and $\widetilde{x}_e$ by Nyström method;
while w_+, w_- not converged **do**
  Randomly select a mini-batch $x_b^+ = \{x_i^+, x_j^+, x_k^+\}$ and a mini-batch $x_b^- = \{x_i^-, x_j^-, x_k^-\}$;
  for $x_b^+ \subset A_e$ and $x_b^- \subset B_e$ **do**
    Compute the gradient $\nabla g_1(w_+, x_i^+, x_j^-, x_k^-)$ by (21);
    Compute the gradient $\nabla g_2(w_-, x_i^-, x_j^+, x_k^+)$ by (22);
    $w_+ \leftarrow w_+ - r_1 \nabla g_1(w_+, x_i^+, x_j^-, x_k^-)$;
    $w_- \leftarrow w_- - r_2 \nabla g_2(w_-, x_i^-, x_j^+, x_k^+)$;
  end for
end while
For $\widetilde{x}$, predict its label by label $(\widetilde{x}) = \arg\min_{i=\pm} \frac{|w_i^T \widetilde{x_e}|}{\|w_i\|}$.

from UCI database [4] and real-world databases[1], respectively. All the computational time involved is the sum of the training time and the testing time and all the classification accuracy involved is the testing accuracy, that is, the classification accuracy on testing sets.

4.1 Experiments on regular-scale datasets

The statistics of the regular-scale datasets are listed in the first four rows in Table 1, where l and n denote the number and the dimensionality of samples, respectively. Gaussian radial basis function (RBF) kernel $k(u, v) = \exp(-\|u - v\|^2/\gamma)$

Table 1. Statistics of datasets

Data set	l	n	Data set	l	n	Data set	l	n	Data set	l	n
australian	690	14	parkinsons	195	22	bupa	345	6	ringnorm	400	20
ecoli	336	7	sonar	208	60	german	1000	24	spect	80	22
haberman	306	3	transfusion	748	4	heart	270	13	twonorm	400	20
ionosphere	351	34	wdbc	569	30	monks2	432	6	wpbc	198	32
cod-rna	216948	8	ijcnn1	141691	22	skin	245057	3	w8a	64700	300

for $u, v \in R^n$ is selected and SMO [11] algorithm is used for SVM, where $\gamma > 0$ is a kernel parameter. We use SOR solver [10] for fast training TBSVM; the source code of Zhang and Zhou [23] for LDM; for TBLDM, the 'quadprog' toolbox in MATLAB [1] is used to solve QPPs (14) and (16). All the experiments are operated in MATLAB. For the convenience of computation, we take all the model parameters $C, c_1, c_2, c_3, c_4 = 1$, the kernel parameter γ and λ_1, λ_2 are chosen

[1] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

from $[2^{-6}, 2^6]$ by using 5-fold cross validation method. Experiments are repeated for 5 times with random data partitions to calculate the average accuracies and variances. The experimental results are listed in Table 2, from which we can see that for computational time, TBLDM is obviously faster than LDM except on spect and wpbc datasets, and faster than TBSVM on 12 datasets and similar on the remaining 4 datasets. For classification accuracy, TBLDM is higher than LDM on 11 datasets and same on wdbc data set, and is higher than TBSVM on 13 datasets. In addition, SVM only gets the highest classification accuracy on wdbc data set although its computational time is the fastest.

Table 2. Experimental results with regular size datasets

	SVM		TBSVM		LDM		TBLDM	
DATASETS	acc(mean±std)	time(s)	acc(mean±std)	time(s)	acc(mean±std)	time(s)	acc(mean±std)	time(s)
australian	0.8565±0.0262	0.0353	0.8574±0.0360	0.2015	0.8557±0.0258	1.4757	**0.8672±0.0194**	0.1500
bupa	0.6736±0.0591	0.0292	0.6986±0.0495	0.0818	0.6980±0.0542	0.1893	**0.7014±0.0366**	0.0820
ecoli	0.9637±0.0264	0.0189	0.9648±0.0331	0.0743	**0.9672±0.0232**	0.1785	0.9637±0.0175	0.0939
german	0.7224±0.0367	0.0611	0.7510±0.0211	0.5345	0.7590±0.0186	4.9356	**0.7724±0.0223**	0.4213
haberman	0.7333±0.0235	0.0241	0.7210±0.0203	0.1320	0.7353±0.0344	0.1337	**0.7380±0.0396**	0.0805
heart	0.8333±0.0367	0.0220	0.8356±0.0567	0.0577	0.8326±0.0461	0.0948	**0.8363±0.0464**	0.0491
ionosphere	0.9345±0.0327	0.0233	0.9248±0.0224	0.0688	**0.9441±0.0254**	0.2189	0.8872±0.0308	0.0511
monks2	0.7940±0.0450	0.0330	0.8065±0.0212	0.0595	0.8074±0.0391	0.3659	**0.8320±0.0448**	0.0509
parkinsons	0.9159±0.0387	0.0270	0.8995±0.0386	0.0399	0.9344±0.0421	0.0451	**0.9415±0.0378**	0.0358
ringnorm	0.9530±0.0273	0.0270	0.9560±0.0226	0.0594	**0.9675±0.0189**	0.3284	0.8485±0.0337	0.0448
sonar	0.8066±0.0460	0.0245	0.8489±0.0480	0.0467	0.8568±0.0519	0.0592	**0.8738±0.0449**	0.0296
spect	0.6900±0.1119	0.0217	0.6875±0.0633	0.0213	**0.7025±0.1094**	0.0038	**0.7025±0.1033**	0.0214
transfusion	0.7348±0.0262	0.0487	0.7628±0.0175	0.8928	**0.7939±0.0264**	1.8919	0.7839±0.0249	0.6641
twonorm	0.9725±0.0186	0.0195	0.9720±0.0158	0.0647	0.9695±0.0205	0.2988	**0.9730±0.0165**	0.0636
wdbc	**0.9761±0.0116**	0.0215	0.9708±0.0119	0.1345	0.9743±0.0146	0.8451	0.9743±0.0148	0.1185
wpbc	0.7627±0.0118	0.0306	0.7697±0.0176	0.0420	0.7988±0.0472	0.0428	**0.8002±0.0405**	0.0439

4.2 Experiments on large-scale datasets

The statistics of the large-scale datasets are listed in the last row of Table 1. All of these four large-scale datasets are split into training and test parts. To compare with our method, we employ linear SVM, linear LDM and linear TBLDM after Nyström method. We choose Liblinear for linear SVM; the source code of Zhang and Zhou [23] for linear LDM. A nonlinear SVM also runs directly on these large-scale datasets. For the convenience of computation, $C, c_1, c_2, c_3, c_4, \lambda_1, \lambda_2, \lambda_3, \lambda_4$ are all set to 1, γ that used for nonlinear SVM and Nyström method is set to the average squared distance between data points and the sample mean. The number of landmark points of Nyström method is chosen as $m = 50, 100$. Table 3 tells us that all linear classifiers running after the Nyström method can get a close classification accuracy result compared to nonlinear SVM, even with such small number of landmark points m. However, we can see from Table 4 that the running time of all linear classifier frameworks plus Nyström method are much faster than that of nonlinear SVM. Moreover, we can see that TBLDM is the fastest if we only compared the time running by three linear classifiers. In addition to nonlinear SVM, all classifiers labelled as SVM, LDM and TBLDM in Table 3 and Table 4 are linear.

Table 3. Classification accuracy results on 4 large-scale datasets

		m=50			m=100		
DATASETS	Nonlinear-SVM	SVM	LDM	TBLDM	SVM	LDM	TBLDM
cod-rna	0.8778	0.8650	0.8542	0.8536	0.8651	0.8618	0.8541
ijcnn1	0.9840	0.9138	0.9050	0.9050	0.9357	0.9203	0.9159
skin	0.9756	0.9982	0.9972	0.9759	0.9985	0.9978	0.9807
w8a	0.9939	0.9696	0.9697	0.9698	0.9721	0.9709	0.9707

Table 4. Time (seconds) comparison on 4 large-scale data sets

		m=50				m=100			
DATASETS	Nonlinear-SVM	Nyström	SVM	LDM	TBLDM	Nyström	SVM	LDM	TBLDM
cod-rna	358.88	0.41	0.50	0.49	0.33	0.71	0.55	0.53	0.34
ijcnn1	46.28	0.38	0.63	0.67	0.12	0.65	1.09	1.23	0.15
skin	1357.9	0.86	0.99	1.64	0.92	1.45	1.49	2.42	0.84
w8a	533.02	1.39	0.30	0.40	0.05	1.77	0.54	0.73	0.07

5 Conclusions

Inspired by the idea of LDM and TBSVM, in this paper, we introduce the notions of positive margin and negative margin of samples and then present a novel classification method, TBLDM, by optimizing the positive and negative margin distributions. The experimental results on sixteen regular scale datasets and four large scale datasets indicate that, compared with SVM, TBSVM and LDM, the proposed TBLDM is a fast, effective and robust classifier. From the derivation process in Section 3, we can see that the technique used in this paper has a certain commonality. Therefore, it will be interesting to generalize the idea of TBLDM to regression models and other learning settings, which will be our next work.

Bibliography

[1] MATLAB version 9.2.0.538062 (R2017a). The Mathworks, Inc., Natick, Massachusetts (2017)
[2] Breiman, L.: Prediction games and arcing algorithms. Neural computation 11(7), 1493–1517 (1999)
[3] Cortes, C., Vapnik, V.: Support-vector networks. Machine learning 20(3), 273–297 (1995)
[4] Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), `http://archive.ics.uci.edu/ml`
[5] Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 55(1), 119–139 (1997)
[6] Gao, W., Zhou, Z.H.: On the doubt about margin explanation of boosting. Artificial Intelligence 203, 1–18 (2013)
[7] Jayadeva, Khemchandani, R., et al.: Twin support vector machines for pattern classification. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29(5), 905–910 (2007)

[8] Khemchandani, R., Sharma, S.: Robust least squares twin support vector machine for human activity recognition. Applied Soft Computing 47, 33–46 (2016)
[9] Luo, X., Durrant, R.J.: Maximum margin principal components. arXiv preprint arXiv:1705.06371 (2017)
[10] Mangasarian, O.L., Musicant, D.R.: Successive overrelaxation for support vector machines. IEEE Transactions on Neural Networks 10(5), 1032–1037 (1999)
[11] Platt, J., et al.: Sequential minimal optimization: A fast algorithm for training support vector machines (1998)
[12] Rastogi, R., Sharma, S., Chandra, S.: Robust parametric twin support vector machine for pattern classification. Neural Processing Letters pp. 1–31 (2017)
[13] Reyzin, L., Schapire, R.E.: How boosting the margin can also boost classifier complexity. In: Proceedings of the 23rd international conference on Machine learning. pp. 753–760. ACM (2006)
[14] Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of statistics pp. 1651–1686 (1998)
[15] Shao, Y.H., Chen, W.J., Wang, Z., Li, C.N., Deng, N.Y.: Weighted linear loss twin support vector machine for large-scale classification. Knowledge-Based Systems 73, 276–288 (2015)
[16] Shao, Y.H., Zhang, C.H., Wang, X.B., Deng, N.Y.: Improvements on twin support vector machines. Neural Networks, IEEE Transactions on 22(6), 962–968 (2011)
[17] Vapnik, V.: The nature of statistical learning theory. Springer Science & Business Media (2013)
[18] Vapnik, V.N., Vapnik, V.: Statistical learning theory, vol. 1. Wiley New York (1998)
[19] Wang, L., Sugiyama, M., Yang, C., Zhou, Z.H., Feng, J.: On the margin explanation of boosting algorithms. In: COLT. pp. 479–490. Citeseer (2008)
[20] Xu, H., Fan, L., Gao, X.: Projection twin smms for 2d image data classification. Neural Computing and Applications 26(1), 91–100 (2015)
[21] Xu, Y., Pan, X., Zhou, Z., Yang, Z., Zhang, Y.: Structural least square twin support vector machine for classification. Applied Intelligence 42(3), 527–536 (2015)
[22] Zhang, K., Kwok, J.T.: Clustered nyström method for large scale manifold learning and dimension reduction. IEEE Transactions on Neural Networks 21(10), 1576–1587 (2010)
[23] Zhang, T., Zhou, Z.H.: Large margin distribution machine. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 313–322. ACM (2014)
[24] Zhang, T., Zhou, Z.H.: Optimal margin distribution clustering (2018)