Twin Margin Hyperplanes Distribution Machine with Equality Constraints

Gong Rongfen, Chu Maoxiang*, *Member, IAENG,* Liu Ling, Liu Liming

***Abstract*—In this paper, we introduce a classification framework named twin margin hyperplanes distribution machine with equality constraints (ETMDM). Unlike traditional twin support vector machine (TSVM) constructs boundary hyperplanes through quadratic programming with inequality constraints, ETMDM determines two margin hyperplanes by solving linear equations, thereby completely eliminating inequality constraints. The margin hyperplanes exploit the margin distribution information of all samples by the margin mean and margin variance in large margin distribution machine (LDM). And the margin mean and margin variance are reconstructed by weighted linear loss and optimization scheme. The reconstructed margin distribution information can avoid suffering from the possible negative infinity problem and improve the computational efficiency. The experimental results on different types of datasets demonstrate that our ETMDM has excellent classification accuracy but with less computational time.**

***Index Terms*—twin support vector machine, large margin distribution, equality constraints, weighted linear loss**

I. Introduction

Support vector machines (SVMs), as excellent pattern classification methods, have been widely applied [1-6]. For the standard SVM [7-8], the primal idea is to construct two parallel boundary hyperplanes. For binary classification problem, the two boundary hyperplanes separate the positive and negative samples in the training dataset. And the distance, defined as the margin, between them is maximized. The purpose of the margin maximization is to minimize the structural risk. It also provides a theoretical explanation for the good generalization performance of SVM. Because of this, many scholars have been convinced by SVM and carried out in-depth research and expansion. Many variants of the promoted SVMs have been proposed, such as structural regularized SVM [9], L-0/1 soft-margin loss SVM [10], spheres-based SVM [11], pinball loss SVM [12], local-to-global SVM [13], fuzzy SVM (FSVM) [14] and graph FSVM [15].

Different from the standard SVM, twin support vector machine (TSVM) [16] with non-parallel decision hyperplanes is proposed. TSVM constructs a decision hyperplane for the target samples and a boundary hyperplane for the other samples. The positive and negative samples are targeted in turn. So, TSVM generates two decision hyperplanes. The two decision hyperplanes may not be parallel, which means that they can be used for the classification of cross-planes data. Another advantage is that TSVM is 4 times faster than SVM. The reason is that TSVM addresses two smaller quadratic programming problems (QPPs) with inequality constraints. TSVM did not consider the margin maximization until the twin bounded SVM [17] was proposed. Since then, the different versions of TSVM have been proposed, such as nonparallel SVM [18], angle-based TSVM [19], truncated pinball loss TSVM [20], L-1-norm loss-based projection TSVM [21], Indefinite TSVM [22] and uncertainty-aware TSVM [23].

Although the margin maximization principle plays an important role in the SVMs and TSVMs studies, it does not guarantee the good generalization performance of the models. Some studies have shown that the margin distribution is more important [24-25]. The margin distribution focuses on the potential distribution information of more samples in training dataset, which can improve the generalization performance. For this reason, Zhang et al. [26] proposed a new large margin distribution machine (LDM). In LDM, the margin mean and margin variance are used to describe the distribution information of training dataset. Due to the success of LDM, the models with margin distribution optimization have been developed, such as cost-sensitive LDM [27], adjustable LDM [28], unconstrained LDM [29], Laplacian LDM [30] and twin bounded LDM [31].

However, the LDM suffers from low computational efficiency. In terms of SVMs and TSVMs, there are some models with high computational speed, such as LSSVM [32] and LSTSVM [33]. The reason is that they solve the QPPs with equality constraints. Therefore, one way to solve the problem of low computational efficiency is to apply equality constraints in LDM, such as least squares LDM-based regression [34]. In this paper, we propose a novel twin margin hyperplanes distribution machine with equality constraints (ETMDM) for pattern classification. The ETMDM has the following characteristics:

(1) The basic TSVM framework is reconstructed. Inspired by LDM, the boundary hyperplanes in the original framework are replaced by the margin hyperplanes in ETMDM. Specifically, the margin mean and margin variance

Manuscript received January 7, 2025; revised June 2, 2025.

This work was supported in part by the Natural Science Foundation of Liaoning Province of China (2022-MS-353), Basic Scientific Research Project of Education Department of Liaoning Province of China (2020LNZD06 and LJKMZ20220640).

R. F. Gong is an associate professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China. (e-mail: fx_gong@hotmail.com).

M. X. Chu is a professor of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China. (corresponding author to provide phone: 0412-5929164; e-mail: chu52_2004@163.com).

L. Liu is a postgraduate student of School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, Liaoning 114051 China. (e-mail: ll15566271785@163.com).

L. M. Liu is a PhD candidate of School of Electronic and Information E ngineering, University of Science and Technology Liaoning, Anshan, Liaon ing 114051 China. (e-mail: llm06101021@hotmail.com).

are reconstructed to generate the margin hyperplanes. The margin hyperplanes not only contain the margin distribution information, but also constrain the other samples to be on one side.

(2) The margin distribution is optimized. By analyzing the margin variance, we find that it has duplicate terms with the objective function of TSVM. So, the terms of the margin variance in ETMDM are reduced. In addition, some redundant terms are eliminated.

(3) The equality constraints are adapted. In ETMDM, the inequality constraints are removed. The decision hyperplanes are generated by unconstrained optimization with quadratic loss. The margin hyperplanes are generated by minimizing the unconstrained margin variance and maximizing the equality-constrained margin mean with linear loss. The optimized margin mean can suffer from the negative infinity problem. So, the linear loss of margin mean is limited by weighted parameters.

II. Related works

A. LDM

Consider a sample matrix $\boldsymbol{x}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_m]^T$ and the corresponding label vector $\boldsymbol{y}=[y_1,y_2,\cdots,y_m]^T$, where $\boldsymbol{x}\in R^{m\times n}$. Define the margin mean and margin variance as follows:

$$\bar{\gamma}=\frac{1}{m}\sum_{k=1}^{m}y_k\boldsymbol{w}^T\boldsymbol{x}_k, \tag{1}$$

$$\hat{\gamma}=\sum_{k1=1}^{m}\sum_{k2=1}^{m}(y_{k1}\boldsymbol{w}^T\boldsymbol{x}_{k1}-y_{k2}\boldsymbol{w}^T\boldsymbol{x}_{k2})^2, \tag{2}$$

where $\boldsymbol{w}\in R^{n\times 1}$. LDM optimizes the margin distribution by maximizing the margin mean and minimizing the margin variance. This leads to the following QPP:

$$\begin{aligned}&\min_{w^+b^+}\frac{1}{2}\boldsymbol{w}^T\boldsymbol{w}+c_1\sum_{k=1}^{m}\xi_k-c_2\bar{\gamma}+c_3\hat{\gamma}\\&s.t.\ y_k\boldsymbol{w}^T\boldsymbol{x}_k\ge 1-\xi_k,\\&\qquad \xi_k\ge 0,\ k=1,2,...m,\end{aligned} \tag{3}$$

where ξ_k represents the error variable for $\boldsymbol{x}_k$. c_1, c_2 and c_3 are used to choose a trade-off among four terms in the objective function. The first term embodies the margin maximization principle. The second term with inequality constraints is used to generate two parallel boundary hyperplanes $\boldsymbol{w}^T\boldsymbol{x}_t=1$ and $\boldsymbol{w}^T\boldsymbol{x}_t=-1$, where $\boldsymbol{x}_t$ is a sample vector. The last two terms mean that the margin distribution is trimmed. The decision hyperplane $\boldsymbol{w}^T\boldsymbol{x}_t=0$ is driven by both the two boundary hyperplanes and the last two terms. It can be seen that the computational efficiency of LDM is low, because it solves the QPP with inequality constraint.

B. TSVM

Let $\boldsymbol{x}$ be divided into two parts $\boldsymbol{x}^+=[\boldsymbol{x}_1^+,\boldsymbol{x}_2^+,\cdots,\boldsymbol{x}_{m^+}^+]^T$ and $\boldsymbol{x}^-=[\boldsymbol{x}_1^-,\boldsymbol{x}_2^-,\cdots,\boldsymbol{x}_{m^-}^-]^T$, where $m=m^++m^-$ and $\boldsymbol{x}=[(\boldsymbol{x}^+)^T\ \ (\boldsymbol{x}^-)^T]^T$. And $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ represent the positive and negative sample matrices, respectively. When the target samples are positive, TSVM solves QPP (4). Otherwise, TSVM solves QPP (5).

$$\begin{aligned}&\min_{w^+b^+}\frac{1}{2}\sum_{i=1}^{m^+}\left((\boldsymbol{w}^+)^T\boldsymbol{x}_i^++b^+\right)^2+c^+\sum_{j=1}^{m^-}\xi_j^-\\&s.t.\ (\boldsymbol{w}^+)^T\boldsymbol{x}_j^-+b^+\le -1+\xi_j^-,\\&\qquad \xi_j^-\ge 0,\ j=1,2,...m^-,\end{aligned} \tag{4}$$

$$\begin{aligned}&\min_{w^-b^-}\frac{1}{2}\sum_{j=1}^{m^-}\left((\boldsymbol{w}^-)^T\boldsymbol{x}_j^-+b^-\right)^2+c^-\sum_{i=1}^{m^+}\xi_i^+\\&s.t.\ (\boldsymbol{w}^-)^T\boldsymbol{x}_i^++b^-\ge 1-\xi_i^+,\\&\qquad \xi_i^+\ge 0,\ i=1,2,...m^+,\end{aligned} \tag{5}$$

where $\boldsymbol{w}^+\in R^{n\times 1}$, $\boldsymbol{w}^-\in R^{n\times 1}$, $b^+\in R^{1\times 1}$ and $b^-\in R^{1\times 1}$. ξ_j^-, ξ_i^+ represent the error variables. c^+ and c^- are the trade-off parameters. TSVM constructs two non-parallel decision hyperplanes $(\boldsymbol{w}^+)^T\boldsymbol{x}_t+b^+=0$ and $(\boldsymbol{w}^-)^T\boldsymbol{x}_t+b^-=0$. And the corresponding boundary hyperplanes $(\boldsymbol{w}^+)^T\boldsymbol{x}_t+b^+=-1$ and $(\boldsymbol{w}^-)^T\boldsymbol{x}_t+b^-=1$ are constructed. In QPP (4), the target samples are positive and the other samples are negative. The first term requires the target samples are proximal to the decision hyperplane $(\boldsymbol{w}^+)^T\boldsymbol{x}_t+b^+=0$. The second term with inequality constraints requires the other samples to be on one side of the boundary hyperplane $(\boldsymbol{w}^+)^T\boldsymbol{x}_t+b^+=-1$. For QPP (5), the similar conclusions can be obtained. Although TSVM is four times faster than the standard SVM, the computational efficiency is still not high enough.

III. ETMDM

A. Formulation

First, we present an immature linear model that is a combination of TSVM and LDM, as follows

$$\begin{aligned}&\min_{w^+b^+}\frac{1}{2}\sum_{i=1}^{m^+}\left((\boldsymbol{w}^+)^T\boldsymbol{x}_i^++b^+\right)^2+c_1^+\sum_{j=1}^{m^-}\xi_j^--c_2^+\bar{\gamma}^++c_3^+\hat{\gamma}^+\\&s.t.\ (\boldsymbol{w}^+)^T\boldsymbol{x}_j^-+b^+\le -1+\xi_j^-,\\&\qquad \xi_j^-\ge 0,\ j=1,2,...m^-,\end{aligned} \tag{6}$$

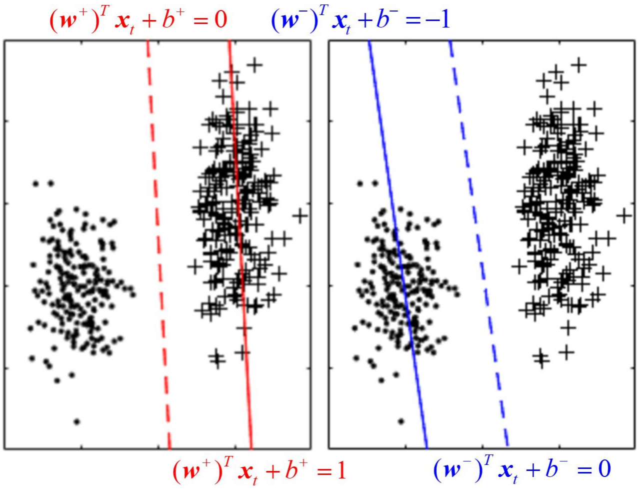
$$\begin{aligned}&\min_{w^-b^-}\frac{1}{2}\sum_{j=1}^{m^-}\left((\boldsymbol{w}^-)^T\boldsymbol{x}_j^-+b^-\right)^2+c_1^-\sum_{i=1}^{m^+}\xi_i^+-c_2^-\bar{\gamma}^-+c_3^-\hat{\gamma}^-\\&s.t.\ (\boldsymbol{w}^-)^T\boldsymbol{x}_i^++b^-\ge 1-\xi_i^+,\\&\qquad \xi_i^+\ge 0,\ i=1,2,...m^+,\end{aligned} \tag{7}$$

where c_1^+, c_2^+, c_3^+, c_1^-, c_2^- and c_3^- are trade-off parameters. $\bar{\gamma}^+$, $\hat{\gamma}^+$, $\bar{\gamma}^-$ and $\hat{\gamma}^-$ can be derived from formulas (1) and (2). The immature model inherits the excellent genes of TSVM and LDM. But the immature model has the following unresolved issues:

(1) It solves the QPPs (6) and (7) with inequity constraints, which leads to low computational efficiency.

(2) It oversimplified the introduction of margin mean and margin variance. For QPP (6), $\bar{\gamma}^+$ and $\hat{\gamma}^+$ are optimized for the decision hyperplane $(\boldsymbol{w}^+)^T\boldsymbol{x}_t+b^+=0$. Their optimization results in the decision hyperplane separating the positive and negative samples. This does not meet the original functional requirements of the decision hyperplane. The same problem exists for QPP (7).

(3) For QPP (6), $\hat{\gamma}^+$ has a similar function as the first item of objective function. And $\hat{\gamma}^+$ has redundant functionality.



(a) (b)

Fig. 1. The margin hyperplanes and the decision hyperplanes.

To address the above issues, we optimize the immature model. First, the margin hyperplane is proposed. Fig. 1(a) shows the margin hyperplane $(\boldsymbol{w}^+)^T \boldsymbol{x}_t + b^+ = 0$ and decision hyperplane $(\boldsymbol{w}^+)^T \boldsymbol{x}_t + b^+ = 1$ for positive samples. the margin hyperplane is obtained by maximizing the margin mean $\bar{\gamma}^+$ and minimizing the margin variance $\hat{\gamma}^+$. Meanwhile, the margin hyperplane also fulfilled the requirement that the negative samples be on one side. So, the boundary hyperplane is not adopted here. This also produces an excellent result that the newly optimized model no longer contains inequality constraints.

The formulas for $\bar{\gamma}^+$ and $\hat{\gamma}^+$ are as follows:

$$\bar{\gamma}^+ = \frac{1}{m}\sum_{k=1}^{m} y_k \left((\boldsymbol{w}^+)^T \boldsymbol{x}_k + b^+ \right), \tag{8}$$

$$\begin{aligned}\hat{\gamma}^+ &= \hat{\gamma}_+^+ + \hat{\gamma}_-^+ + \hat{\gamma}_\pm^+ \\ &= \sum_{k1=1}^{m}\sum_{k2=1}^{m} \left(y_{k1}\left((\boldsymbol{w}^+)^T \boldsymbol{x}_{k1} + b^+ \right) - y_{k2}\left((\boldsymbol{w}^+)^T \boldsymbol{x}_{k2} + b^+ \right) \right)^2,\end{aligned} \tag{9}$$

where

$$\begin{aligned}\hat{\gamma}_+^+ &= \sum_{i1=1}^{m^+}\sum_{i2=1}^{m^+} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_{i1}^+ - (\boldsymbol{w}^+)^T \boldsymbol{x}_{i2}^+ \right)^2, \hat{\gamma}_-^+ \\ &= \sum_{j1=1}^{m^-}\sum_{j2=1}^{m^-} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_{j1}^- - (\boldsymbol{w}^+)^T \boldsymbol{x}_{j2}^- \right)^2,\end{aligned} \tag{10}$$

$$\hat{\gamma}_\pm^+ = 2\sum_{i=1}^{m^+}\sum_{j=1}^{m^-} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_i^+ + (\boldsymbol{w}^+)^T \boldsymbol{x}_j^- + 2b^+ \right)^2. \tag{11}$$

$\hat{\gamma}_\pm^+$ represents the margin variance between the positive and negative samples. Minimizing $\hat{\gamma}_\pm^+$ means that the margin distribution based on $(\boldsymbol{w}^+)^T \boldsymbol{x}_t + b^+ = 0$ is optimized. Minimizing $\hat{\gamma}_+^+$ drives the positive samples to cluster in the margin direction, which tends to be the same as the effect of minimizing the first term of the objective function in QPP (6). So, $\hat{\gamma}_+^+$ is can be removed. On one hand, minimizing $\hat{\gamma}_-^+$ drives the negative samples to cluster in the margin direction. On the other hand, the negative samples are mainly required to be on one side of $(\boldsymbol{w}^+)^T \boldsymbol{x}_t + b^+ = 0$ so as to be at least 1 away from $(\boldsymbol{w}^+)^T \boldsymbol{x}_t + b^+ = 1$. Therefore, we consider $\hat{\gamma}_-^+$ is redundant and can be abandoned.

In addition, maximizing the margin mean $\bar{\gamma}^+$ is equivalent to

$$\begin{aligned}&\min \sum_{k=1}^{m} \xi_k^+ \\ &s.t.\ y_k \left((\boldsymbol{w}^+)^T \boldsymbol{x}_k + b^+ \right) = -\xi_k^+, \\ &\quad k = 1, 2, \cdots, m.\end{aligned} \tag{12}$$

As you can see, it is an optimization problem with linear loss. the optimization problem may suffer the negative infinity problem. The solution is to use the weighted linear loss [35]. In other words, the target function in (12) is modified to $v_k^+ \xi_k^+$. v_k^+ is given by

$$v_k^+ = \begin{cases} 1 & \xi_k^+ \geq \xi_{meam}^+, \\ 10^{-4} & otherwise, \end{cases} \tag{13}$$

where ξ_{meam}^+ is the mean of all ξ_k^+ with $v_k^+ = 1$. It can be seen from (13) that the excessively small ξ_k^+ is ignored to avoid the possible infinity problem.

The margin hyperplane $(\boldsymbol{w}^-)^T \boldsymbol{x}_t + b^- = 0$ and the decision hyperplane $(\boldsymbol{w}^-)^T \boldsymbol{x}_t + b^- = -1$ for negative samples are shown in Fig. 1(b). $\hat{\gamma}_\pm^-$ and v_k^- are given by

$$\hat{\gamma}_\pm^- = 2\sum_{j=1}^{m^-}\sum_{i=1}^{m^+} \left((\boldsymbol{w}^-)^T \boldsymbol{x}_j^- + (\boldsymbol{w}^-)^T \boldsymbol{x}_i^+ + 2b^- \right)^2, \tag{14}$$

$$v_k^- = \begin{cases} 1 & \xi_k^- \geq \xi_{meam}^-, \\ 10^{-4} & otherwise. \end{cases} \tag{15}$$

After the above optimization, the linear ETMDM in this paper can be derived as

$$\begin{aligned}\min_{w^+ b^+} \ &\frac{1}{2}\sum_{i=1}^{m^+} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_i^+ + b^+ - 1 \right)^2 + c_1^+ \sum_{k=1}^{m} v_k^+ \xi_k^+ \\ &+ 2c_2^+ \sum_{i=1}^{m^+}\sum_{j=1}^{m^-} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_i^+ + (\boldsymbol{w}^+)^T \boldsymbol{x}_j^- + 2b^+ \right)^2. \\ s.t.\ &y_k \left((\boldsymbol{w}^+)^T \boldsymbol{x}_k + b^+ \right) = -\xi_k^+,\ k = 1, 2, \ldots m,\end{aligned} \tag{16}$$

$$\begin{aligned}\min_{w^- b^-} \ &\frac{1}{2}\sum_{j=1}^{m^-} \left((\boldsymbol{w}^-)^T \boldsymbol{x}_j^- + b^- + 1 \right)^2 + c_1^- \sum_{k=1}^{m} v_k^- \xi_k^- \\ &+ 2c_2^- \sum_{j=1}^{m^-}\sum_{i=1}^{m^+} \left((\boldsymbol{w}^-)^T \boldsymbol{x}_j^- + (\boldsymbol{w}^-)^T \boldsymbol{x}_i^+ + 2b^- \right)^2, \\ s.t.\ &y_k \left((\boldsymbol{w}^-)^T \boldsymbol{x}_k + b^- \right) = -\xi_k^-,\ k = 1, 2, \ldots, m.\end{aligned} \tag{17}$$

B. Solution to linear ETMDM

First, consider the solution of QPP (16). Substituting the equality constraints and the formula (11) into the objective function, we can get

$$\begin{aligned}L = &\frac{1}{2}\sum_{i=1}^{m^+} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_i^+ + b^+ - 1 \right)^2 - c_1^+ \sum_{k=1}^{m} v_k^+ y_k \left((\boldsymbol{w}^+)^T \boldsymbol{x}_k + b^+ \right) \\ &+ 2c_2^+ \sum_{i=1}^{m^+}\sum_{j=1}^{m^-} \left((\boldsymbol{w}^+)^T \boldsymbol{x}_i^+ + (\boldsymbol{w}^+)^T \boldsymbol{x}_j^- + 2b^+ \right)^2.\end{aligned} \tag{18}$$

It can also be rewritten as

$$\begin{aligned}L=&\frac{1}{2}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+}-\boldsymbol{e}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+}-\boldsymbol{e}^{+})\\&-c_1^{+}(\boldsymbol{v}^{+})^{T}(\boldsymbol{Yxw}^{+}+\boldsymbol{y}b^{+})\\&+2c_2^{+}m^{-}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})\\&+2c_2^{+}m^{+}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})^{T}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})\\&+4c_2^{+}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+}),\end{aligned}\tag{19}$$

where $\boldsymbol{Y}$ is a diagonal matrix with elements $y_1, y_2, \cdots, y_m$. $\boldsymbol{e}^{+}$ and $\boldsymbol{e}^{-}$ are vectors of ones of appropriate dimension. Then, setting the gradient of L with respect to $\boldsymbol{w}^{+}$ and b^{+} to be 0, we can get

$$\begin{aligned}&4c_2^{+}(\boldsymbol{x}^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})\\&+4c_2^{+}(\boldsymbol{x}^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})\\&+4c_2^{+}m^{-}(\boldsymbol{x}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})\\&+4c_2^{+}m^{+}(\boldsymbol{x}^{-})^{T}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})\\&+(\boldsymbol{x}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+}-\boldsymbol{e}^{+})-c_1^{+}(\boldsymbol{x})^{T}\boldsymbol{Yv}^{+}=0,\end{aligned}\tag{20}$$

$$\begin{aligned}&4c_2^{+}(\boldsymbol{e}^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})\\&+4c_2^{+}(\boldsymbol{e}^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})\\&+4c_2^{+}m^{-}(\boldsymbol{e}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})\\&+4c_2^{+}m^{+}(\boldsymbol{e}^{-})^{T}(\boldsymbol{x}^{-}\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})\\&+(\boldsymbol{e}^{+})^{T}(\boldsymbol{x}^{+}\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+}-\boldsymbol{e}^{+})-c_1^{+}\boldsymbol{y}^{T}\boldsymbol{v}^{+}=0.\end{aligned}\tag{21}$$

Define $\boldsymbol{G}^{+}=[\boldsymbol{x}^{+}\ \boldsymbol{e}^{+}]$, $\boldsymbol{G}^{-}=[\boldsymbol{x}^{-}\ \boldsymbol{e}^{-}]$ and $\boldsymbol{G}=[\boldsymbol{x}\ \boldsymbol{e}]$. Rearranging (20) and (21) in matrix form leads to

$$\begin{aligned}&\Big((\boldsymbol{G}^{+})^{T}\boldsymbol{G}^{+}+4c_2^{+}m^{-}(\boldsymbol{G}^{+})^{T}\boldsymbol{G}^{+}\\&+4c_2^{+}m^{+}(\boldsymbol{G}^{-})^{T}\boldsymbol{G}^{-}+4c_2^{+}(\boldsymbol{G}^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}\boldsymbol{G}^{-}\\&+4c_2^{+}(\boldsymbol{G}^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}\boldsymbol{G}^{+}\Big)\begin{bmatrix}\boldsymbol{w}^{+}\\b^{+}\end{bmatrix}=c_1^{+}\boldsymbol{G}^{T}\boldsymbol{Yv}^{+}+(\boldsymbol{G}^{+})^{T}\boldsymbol{e}^{+}.\end{aligned}\tag{22}$$

Finally, the solution to QPP (16) is obtained

$$\begin{bmatrix}\boldsymbol{w}^{+}\\b^{+}\end{bmatrix}=\left(\boldsymbol{Q}^{+}+\varepsilon\boldsymbol{I}^{+}\right)^{-1}\left(c_1^{+}\boldsymbol{G}^{T}\boldsymbol{Yv}^{+}+(\boldsymbol{G}^{+})^{T}\boldsymbol{e}^{+}\right),\tag{23}$$

$$\begin{aligned}\boldsymbol{Q}^{+}=&(\boldsymbol{G}^{+})^{T}\boldsymbol{G}^{+}+4c_2^{+}m^{-}(\boldsymbol{G}^{+})^{T}\boldsymbol{G}^{+}+4c_2^{+}m^{+}(\boldsymbol{G}^{-})^{T}\boldsymbol{G}^{-}\\&+4c_2^{+}(\boldsymbol{G}^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}\boldsymbol{G}^{-}+4c_2^{+}(\boldsymbol{G}^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}\boldsymbol{G}^{+}.\end{aligned}\tag{24}$$

In the same way, the solution to QPP (17) is gotten

$$\begin{bmatrix}\boldsymbol{w}^{-}\\b^{-}\end{bmatrix}=\left(\boldsymbol{Q}^{-}+\varepsilon\boldsymbol{I}^{-}\right)^{-1}\left(c_1^{-}\boldsymbol{G}^{T}\boldsymbol{Yv}^{-}-(\boldsymbol{G}^{-})^{T}\boldsymbol{e}^{-}\right),\tag{25}$$

$$\begin{aligned}\boldsymbol{Q}^{-}=&(\boldsymbol{G}^{-})^{T}\boldsymbol{G}^{-}+4c_2^{-}m^{-}(\boldsymbol{G}^{+})^{T}\boldsymbol{G}^{+}+4c_2^{-}m^{+}(\boldsymbol{G}^{-})^{T}\boldsymbol{G}^{-}\\&+4c_2^{-}(\boldsymbol{G}^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}\boldsymbol{G}^{-}+4c_2^{-}(\boldsymbol{G}^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}\boldsymbol{G}^{+},\end{aligned}\tag{26}$$

where $\boldsymbol{I}^{+}$ and $\boldsymbol{I}^{-}$ are identity matrixes of appropriate dimensions. ε is a minimal positive value that is used to avoid the possible ill conditioning of $\boldsymbol{Q}^{+}$ and $\boldsymbol{Q}^{-}$.

Once $\boldsymbol{w}^{+}$, $\boldsymbol{w}^{-}$, b^{+} and b^{-} are determined, a new sample $\boldsymbol{x}_t$ can be decided through the decision hyperplanes. The decision function of linear ETMDM is as follows:

$$class=\arg\min_{l=+,-}|h^{l}(\boldsymbol{x}_t)|,\tag{27}$$

where $h^{+}(\boldsymbol{x}_t)=\boldsymbol{x}_t^{T}\boldsymbol{w}^{+}+b^{+}-1$ and $h^{-}(\boldsymbol{x}_t)=\boldsymbol{x}_t^{T}\boldsymbol{w}^{-}+b^{-}+1$. $|\cdot|$ is the distance of point $\boldsymbol{x}_t$ from the decision lanes.

The algorithm of linear ETMDM is summarized as follows:

Algorithm 1:

1) Input the training dataset $\boldsymbol{x}=[(\boldsymbol{x}^{+})^{T}\ \ (\boldsymbol{x}^{-})^{T}]^{T}$.

2) Set $\boldsymbol{v}^{+}=\boldsymbol{e}^{+}$ and $\boldsymbol{v}^{-}=\boldsymbol{e}^{-}$. Choose the appropriate parameters c_1^{+}, c_2^{+}, c_1^{-} and c_2^{-} by using 5-fold cross validation method.

3) Calculate $\boldsymbol{w}^{+}$, b^{+}, $\boldsymbol{w}^{-}$ and b^{-} by (23) and (25). Next, calculate ξ_k^{+} and ξ_k^{-} by (16) and (17).

4) Calculate the mean ξ_{meam}^{+} and ξ_{meam}^{-} from ξ_k^{+} and ξ_k^{-} $(k=1,2,\cdots,m)$, respectively. Then, obtain $\boldsymbol{v}^{+}$ and $\boldsymbol{v}^{-}$ by (13) and (15).

5) Calculate the solutions $\boldsymbol{w}^{+}$, b^{+}, $\boldsymbol{w}^{-}$ and b^{-} of (23) and (25) with the new $\boldsymbol{v}^{+}$ and $\boldsymbol{v}^{-}$.

6) Determines the class label of a new sample $\boldsymbol{x}_t$ by (27).

C. Nonlinear ETMDM

In order to extend the linear ETMDM to the nonlinear case, we consider the following kernel-generated decision hyperplanes

$$\psi(\boldsymbol{x}_t^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+b^{+}=1,\ \psi(\boldsymbol{x}_t^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+b^{-}=-1,\tag{28}$$

where $\psi(,)$ is a chosen kernel function. the kernel-generated margin hyperplanes are

$$\psi(\boldsymbol{x}_t^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+b^{+}=0,\ \psi(\boldsymbol{x}_t^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+b^{-}=0.\tag{29}$$

The above kernel-generated hyperplanes are obtained by the following QPPs

$$\begin{aligned}&\min_{w^{+}b^{+}}\frac{1}{2}\sum_{i=1}^{m^{+}}\left(\psi((\boldsymbol{x}_i^{+})^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+b^{+}-1\right)^{2}+c_1^{+}\sum_{k=1}^{m}v_k^{+}\xi_k^{+}+c_2^{+}\hat{\gamma}_{\pm}^{\psi+}\\&s.t.\ \ y_k\left(\psi(\boldsymbol{x}_k^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+b^{+}\right)=-\xi_k^{+},\ k=1,2,...m,\end{aligned}\tag{30}$$

$$\begin{aligned}&\min_{w^{-}b^{-}}\frac{1}{2}\sum_{j=1}^{m^{-}}\left(\psi((\boldsymbol{x}_j^{-})^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+b^{-}+1\right)^{2}+c_1^{-}\sum_{k=1}^{m}v_k^{-}\xi_k^{-}+c_2^{-}\hat{\gamma}_{\pm}^{\psi-}\\&s.t.\ \ y_k\left(\psi(\boldsymbol{x}_k^{T},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+b^{-}\right)=-\xi_k^{-},\ k=1,2,...,m,\end{aligned}\tag{31}$$

where

$$\begin{aligned}\hat{\gamma}_{\pm}^{\psi+}=&2m^{-}(\psi(\boldsymbol{x}^{+},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})^{T}(\psi(\boldsymbol{x}^{+},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})\\&+2m^{+}(\psi(\boldsymbol{x}^{-},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})^{T}(\psi(\boldsymbol{x}^{-},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+})\\&+4(\psi(\boldsymbol{x}^{+},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+\boldsymbol{e}^{+}b^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}(\psi(\boldsymbol{x}^{-},\boldsymbol{x}^{T})\boldsymbol{w}^{+}+\boldsymbol{e}^{-}b^{+}),\end{aligned}\tag{32}$$

$$\begin{aligned}\hat{\gamma}_{\pm}^{\psi-}=&2m^{+}(\psi(\boldsymbol{x}^{-},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+\boldsymbol{e}^{-}b^{-})^{T}(\psi(\boldsymbol{x}^{-},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+\boldsymbol{e}^{-}b^{-})\\&+2m^{-}(\psi(\boldsymbol{x}^{+},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+\boldsymbol{e}^{+}b^{-})^{T}(\psi(\boldsymbol{x}^{+},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+\boldsymbol{e}^{+}b^{-})\\&+4(\psi(\boldsymbol{x}^{-},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+\boldsymbol{e}^{-}b^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}(\psi(\boldsymbol{x}^{+},\boldsymbol{x}^{T})\boldsymbol{w}^{-}+\boldsymbol{e}^{+}b^{-}).\end{aligned}\tag{33}$$

After a similar derivation as in the linear case, the solutions of (30) and (31) are represented as follows:

$$\begin{bmatrix}\boldsymbol{w}^{+}\\b^{+}\end{bmatrix}=\left(\boldsymbol{P}^{+}+\varepsilon\boldsymbol{I}^{+}\right)^{-1}\left(c_1^{+}\boldsymbol{H}^{T}\boldsymbol{Yv}^{+}+(\boldsymbol{H}^{+})^{T}\boldsymbol{e}^{+}\right),\tag{34}$$

$$\begin{bmatrix}\boldsymbol{w}^{-}\\b^{-}\end{bmatrix}=\left(\boldsymbol{P}^{-}+\varepsilon\boldsymbol{I}^{-}\right)^{-1}\left(c_1^{-}\boldsymbol{H}^{T}\boldsymbol{Yv}^{-}-(\boldsymbol{H}^{-})^{T}\boldsymbol{e}^{-}\right),\tag{35}$$

where

$$\begin{aligned}\boldsymbol{P}^{+}=&(\boldsymbol{H}^{+})^{T}\boldsymbol{H}^{+}+4c_2^{+}m^{-}(\boldsymbol{H}^{+})^{T}\boldsymbol{H}^{+}+4c_2^{+}m^{+}(\boldsymbol{H}^{-})^{T}\boldsymbol{H}^{-}\\&+4c_2^{+}(\boldsymbol{H}^{+})^{T}\boldsymbol{e}^{+}(\boldsymbol{e}^{-})^{T}\boldsymbol{H}^{-}+4c_2^{+}(\boldsymbol{H}^{-})^{T}\boldsymbol{e}^{-}(\boldsymbol{e}^{+})^{T}\boldsymbol{H}^{+},\end{aligned}\tag{36}$$

$$\begin{aligned}\boldsymbol{P}^- &= (\boldsymbol{H}^-)^T \boldsymbol{H}^- + 4c_2^- m^- (\boldsymbol{H}^+)^T \boldsymbol{H}^+ + 4c_2^- m^+ (\boldsymbol{H}^-)^T \boldsymbol{H}^- \\ &+ 4c_2^- (\boldsymbol{H}^+)^T \boldsymbol{e}^+ (\boldsymbol{e}^-)^T \boldsymbol{H}^- + 4c_2^- (\boldsymbol{H}^-)^T \boldsymbol{e}^- (\boldsymbol{e}^+)^T \boldsymbol{H}^+.\end{aligned} \tag{37}$$

$$\begin{aligned}\boldsymbol{H}^+ &= [\psi(\boldsymbol{x}^+, \boldsymbol{x}^T) \;\; \boldsymbol{e}^+], \\ \boldsymbol{H}^- &= [\psi(\boldsymbol{x}^-, \boldsymbol{x}^T) \;\; \boldsymbol{e}^-], \\ \boldsymbol{H} &= [\psi(\boldsymbol{x}^+, \boldsymbol{x}^T) \;\; \boldsymbol{e}].\end{aligned} \tag{38}$$

Similarly, a new sample $\boldsymbol{x}_t$ can be decided through the kernel-generated decision hyperplanes. The decision function of nonlinear ETMDM is as follows:

$$class = \arg\min_{l=+,-} | h^{\psi l}(\boldsymbol{x}_t) |, \tag{39}$$

where

$$\begin{aligned}h^{\psi +}(\boldsymbol{x}_t) &= \psi(\boldsymbol{x}_t^T, \boldsymbol{x}^T)\boldsymbol{w}^+ + b^+ - 1, \\ h^{\psi -}(\boldsymbol{x}_t) &= \psi(\boldsymbol{x}_t^T, \boldsymbol{x}^T)\boldsymbol{w}^- + b^- + 1.\end{aligned} \tag{40}$$

The algorithm of nonlinear ETMDM is summarized as follows:

Algorithm 2:

1) Input the training dataset $\boldsymbol{x} = [(\boldsymbol{x}^+)^T \;\; (\boldsymbol{x}^-)^T]^T$.

2) Set $\boldsymbol{v}^+ = \boldsymbol{e}^+$ and $\boldsymbol{v}^- = \boldsymbol{e}^-$. Choose the kernel function $\psi(,)$ and the appropriate parameters c_1^+, c_2^+, c_1^- and c_2^- by using 5-fold cross validation method.

3) Calculate $\boldsymbol{w}^+$, b^+, $\boldsymbol{w}^-$ and b^- by (34) and (35). Next, calculate ξ_k^+ and ξ_k^- by (30) and (31).

4) Calculate the mean ξ_{meam}^+ and ξ_{meam}^- from ξ_k^+ and ξ_k^- $(k = 1, 2, \cdots, m)$, respectively. Then, obtain $\boldsymbol{v}^+$ and $\boldsymbol{v}^-$ by (13) and (15).

5) Calculate the solutions $\boldsymbol{w}^+$, b^+, $\boldsymbol{w}^-$ and b^- of (34) and (35) with the new $\boldsymbol{v}^+$ and $\boldsymbol{v}^-$.

6) Determines the class label of a new sample $\boldsymbol{x}_t$ by (39).

D. Analysis of algorithms

Compared with other state-of-the-art algorithms, the ETMDM has some advantages and disadvantages. They are stated as follows:

(1) Although both TSVM and ETMDM solve for two smaller QPPs, their computational efficiency is different. For TSVM, the quadratic loss and the hinge loss are used in each QPP. The solving speed for the target samples with quadratic loss is fast, but the solving speed for the other samples with hinge loss is slow. This is because the samples with hinge loss are used to optimize the QPP with inequality constraints. For ETMDM, all samples are used to optimize the QPP with equality constraints. ETMDM used all samples with weighted linear loss. So, the ETMDM has better computational efficiency than TSVM. Furthermore, the margin mean and margin variance are reconstructed to generate the margin hyperplanes. And the reconstructed hyperplanes are optimized by ETMDM to investigate the margin distribution information of all samples. As a result, the ETMDM has the margin distribution optimization, which results in better generalization performance than TSVM.

(2) The margin distribution is considered for LDM and ETMDM. LDM maximizes the margin mean and minimizes the margin variance. ETMDM uses the same optimization approach. But the difference is that the margin distribution is trimmed. In the case of positive samples, the terms $\hat{\gamma}_-^+$ and $\hat{\gamma}_+^+$ in the margin distribution are abandoned. Maximizing the margin mean $\bar{\gamma}^+$ is equivalent to the optimization problem with linear loss. Further, the linear loss is weighted to avoid suffering from the possible negative infinity problem. As a result, the ETMDM achieves better performance than LDM. In addition, ETMDM has the twin structure. So, it is more suitable for the classification of cross-planes data than LDM. Of course, LDM is far inferior to ETMDM in computational efficiency. This is because LDM solves a large QPP with inequality constraints using all samples.

(3) In terms of computational efficiency, although ETMDM is higher than TSVM and LDM, it is lower than LSSVM and LSTSVM. The reason is that LSSVM and LSTSVM also solve the QPPs with equality constraints. Also, ETMDM performs more matrix operations than LSSVM and LSTSVM. This can be seen from $\boldsymbol{Q}^+$, $\boldsymbol{Q}^-$ $\boldsymbol{P}^+$ and $\boldsymbol{P}^-$. Nevertheless, ETMDM is better than LSSVM and LSTSVM in generalization performance. LSSVM and LSTSVM employ the term with quadratic loss, which can replace some terms of margin variance. In the case of positive samples, the term with quadratic loss can tends to be the same as $\hat{\gamma}_+^+$. But they do not take into account $\hat{\gamma}_\pm^+$. Moreover, they do not embed the optimization of the margin mean. So, compared with the ETMDM, they are not sufficient in mining the margin distribution information.

IV. Experiments

To demonstrate the performance of ETMDM, some comparative experiments are performed on different types of datasets, including UCI datasets [36], large-scale NDC datasets [37] and industrial NEU dataset [38]. And our ETMDM is compared with SVM, LDM, TSVM, LSSVM and LSTSVM. All models are programmed and implemented in MATLAB 2016b on a PC with Inter Core I7 CPU and 16GB RAM. Moreover, SVM, LDM and TSVM with inequality constraints are solved by the QP solver in MATLAB. For nonlinear case, the radial basis kernel function $\psi(\boldsymbol{x}_{t1}, \boldsymbol{x}_{t2}) = \exp(-\| \boldsymbol{x}_{t1} - \boldsymbol{x}_{t2} \|^2 / \sigma^2)$ is adopted for all models. For brevity's sake, the trade-off parameters are set to the same ($c_1^+ = c_1^-$ and $c_2^+ = c_2^-$) for two QPPs in ETMDM. The same process is done for TSVM and LSTSVM. The trade-off parameters in all models are selected from $\{10^i \mid i = -4, -2, \cdots, 4\}$ and the kernel radius σ is chosen from $\{2^i \mid i = -2, -1, \cdots, 6\}$. In order to determine the optimal parameters, the five-fold cross validation technique and grid searching method are used for parameter selection. Once the optimal parameters are determined, they are used to train the final decision functions.

A. UCI datasets

Tables I and II report the brief information of binary and multiclass UCI datasets [36]. The binary datasets are used to testing the performance of six models for linear and nonlinear cases. The multiclass datasets are used to further demonstrate the robustness of ETMDM in generalization performance. In order to realize multiclass classification, the binary classification models are combined with binary tree model.

The testing results are shown in Tables IIII, IV and V. The "accuracy" represents the mean and standard deviation of the five-fold accuracies. The "time" represents the total CPU time for training and testing. The win/tie/loss results are used to compare the performance between the ETMDM and one of the other five models. The highest accuracies are emphasized in bold. In addition, Fig. 2 shows that the performance comparison of linear and nonlinear ETMDMs in accuracy and time. And according to Table V, the ranks of six models in terms of accuracy and CPU time are listed in Table VI.

TABLE I
THE BRIEF INFORMATION OF BINARY UCI DATASETS

Dataset	Features	Samples
Australian (Da)	14	690
Blood (Db)	5	748
Diabetes (Dc)	8	768
German (Dd)	24	1000
Heart (De)	12	270
Haberman (Df)	3	306
Ionosphere (Dg)	34	351
Liverdisorder (Dh)	6	345
Sonar (Di)	60	208
Vote (Dj)	16	435
Wdbc (Dk)	30	569
Wpbc (Dl)	33	198

TABLE II
THE BRIEF INFORMATION OF MULTICLASS UCI DATASETS

Dataset	Features	Samples	Classes
Glass (D0)	10	214	6
Wine (D1)	13	178	3
Zoo (D2)	17	101	7
Air (D3)	65	359	3
Balance (D4)	5	625	3
Iris (D5)	5	150	3
Libras_movement (D6)	90	360	15
Soybean (D7)	36	47	4
Vowel (D8)	11	528	11
Vehicle (D9)	19	846	4

It can be seen from Table III that the linear ETMDM wins on six datasets in terms of accuracy. Compared with SVM and TSVM, the ETMDM achieves better generalization performance through the reconstructed margin hyperplanes. The advantage of the margin hyperplanes is that they investigate the margin distribution information of all samples. LDM does not outperform ETMDM in generalization performance, because it does not adopt the twin structure. LSSVM and LSTSVM are not sufficient in mining the margin distribution information, resulting in lower accuracies than ETMDM. The win/tie/loss results further validate the superiority of ETMDM. It can also obtain from Table III that the ETMDM is much shorter than LDM, SVM, and TSVM in terms of CPU time for all binary datasets. This is because that ETMDM solves two small QPPs with equality constraints, while LDM, SVM, and TSVM address the large QPP with inequality constraints. Compared with LSSVM and LSTSVM, the ETMDM doesn't have the advantage in CPU time. The reason is that LSSVM and LSTSVM also solve the QPPs with equality constraints. Also, the ETMDM performs more matrix operations than LSSVM and LSTSVM. Even so, the CPU time of ETMDM is also very short. In summary, our linear ETMDM has good classification accuracy and computational efficiency.

From Table IV, it can be seen that the accuracies of nonlinear ETMDM are obviously better than LDM, TSVM, SVM, LSTSVM and LSSVM on most binary datasets. Combining the results in linear and nonlinear cases, it is confirmed that the ETMDM has better generalization performance. The CPU time of nonlinear ETMDM is also shorter than that of LDM, TSVM and SVM, but longer than that of LSTSVM and LSSVM. This is consistent with the conclusion in linear case.

TABLE III
THE TESTING RESULTS OF SIX LINEAR MODELS ON BINARY UCI DATASETS

Dataset	ETMDM Accuracy (%) Time (s)	LDM Accuracy (%) Time (s)	TSVM Accuracy (%) Time (s)	SVM Accuracy (%) Time (s)	LSTSVM Accuracy (%) Time (s)	LSSVM Accuracy (%) Time (s)
Da	86.66±2.03 0.0095	85.22±1.98 0.3138	86.66±1.77 0.0717	85.51±1.64 0.6762	**86.81±2.04** 0.0048	85.07±1.77 0.0120
Db	77.68±0.87 0.0091	76.20±0.25 0.1149	77.01±1.12 0.1551	76.20±0.25 0.1465	77.45±0.98 0.0048	**77.81±1.18** 0.0119
Dc	76.43±2.32 0.0088	65.75±1.65 0.0992	76.43±1.76 0.1767	75.90±3.63 0.5993	**76.69±1.62** 0.0043	65.75±2.93 0.0143
Dd	**77.00±2.35** 0.0117	76.00±0.84 0.3661	76.60±2.18 0.2516	74.80±3.64 0.6664	76.80±2.04 0.0040	75.90±2.13 0.0222
De	74.84±2.43 0.0072	73.53±0.48 0.0266	75.49±2.49 0.0249	73.53±0.48 0.0241	74.52±2.59 0.0037	**75.54±3.10** 0.0045
Df	**82.96±4.60** 0.0064	80.74±4.16 0.0330	81.48±4.54 0.0238	80.37±3.23 0.0382	**82.96±4.74** 0.00029	81.11±3.19 0.0044
Dg	89.17±3.80 0.0069	79.20±2.37 0.0564	**90.30±3.07** 0.0364	88.88±2.48 0.1191	88.87±4.10 0.0030	81.19±3.35 0.0055
Dh	**70.14±5.47** 0.0064	58.26±1.08 0.0488	69.86±3.09 0.0258	66.96±9.09 0.1003	**70.14±3.25** 0.0038	66.96±5.97 0.0054
Di	**78.85±2.74** 0.0069	75.44±4.89 0.0228	75.93±6.49 0.0225	78.39±5.15 0.0290	77.85±5.26 0.0030	75.47±4.67 0.0032
Dj	**94.71±3.42** 0.0078	93.12±2.95 0.0329	94.50±3.08 0.0753	93.58±3.18 0.1547	94.50±3.08 0.0030	94.04±3.17 0.0055
Dk	97.71±0.71 0.0082	90.32±4.17 0.0557	96.31±1.29 0.0811	**97.89±1.31** 0.3273	97.71±0.43 0.0033	93.31±2.15 0.0079
Dl	**82.35±4.95** 0.0071	79.31±2.78 0.0207	80.34±7.34 0.0251	76.28±0.82 0.0297	80.35±3.40 0.0031	79.85±2.76 0.0046
Win/Tie/Loss	-	12/0/0	8/2/2	11/0/1	7/3/2	9/0/3

TABLE IV
THE TESTING RESULTS OF SIX NONLINEAR MODELS ON BINARY UCI DATASETS

Dataset	ETMDM Accuracy (%) Time (s)	LDM Accuracy (%) Time (s)	TSVM Accuracy (%) Time (s)	SVM Accuracy (%) Time (s)	LSTSVM Accuracy (%) Time (s)	LSSVM Accuracy (%) Time (s)
Da	**87.24±2.29** 0.0432	**87.24±2.75** 0.2692	86.95±2.89 0.1307	86.38±1.68 0.3720	87.10±2.67 0.0312	86.23±2.71 0.0192
Db	**79.42±3.26** 0.0503	76.74±0.74 0.1083	79.15±2.75 0.1834	76.60±0.71 0.3537	79.41±2.42 0.0275	78.88±2.36 0.0156
Dc	**77.73±1.93** 0.0524	74.87±0.56 0.2276	77.21±2.21 0.1883	76.69±2.12 0.4752	77.61±1.98 0.0330	76.95±3.01 0.0178
Dd	76.00±3.38 0.1022	72.50±1.41 0.1919	76.00±3.27 0.3514	75.20±3.46 0.8754	75.00±2.81 0.0712	**76.60±2.80** 0.0294
De	**75.83±3.08** 0.0127	73.87±1.62 0.0200	75.17±2.30 0.0267	74.18±2.17 0.0642	75.15±2.53 0.0073	74.19±2.56 0.0054
Df	82.59±4.16 0.0115	80.00±4.60 0.0360	**82.96±3.59** 0.0262	80.74±4.48 0.0400	82.22±4.48 0.0086	81.11±4.89 0.0052
Dg	**96.01±1.67** 0.0154	87.46±2.13 0.0585	95.43±3.05 0.0671	94.87±1.15 0.0767	93.15±1.68 0.0165	92.02±1.72 0.0050
Dh	**74.78±3.38** 0.0134	59.13±1.42 0.0233	74.49±4.16 0.0391	72.17±5.90 0.0727	74.49±3.38 0.0095	73.62±2.96 0.0055
Di	**90.41±3.34** 0.0117	83.67±3.10 0.0213	89.44±2.36 0.0292	88.93±2.00 0.0267	**90.41±3.34** 0.0127	90.36±2.74 0.0044
Dj	**95.62±2.34** 0.0239	92.88±2.73 0.0741	95.40±2.05 0.0598	94.50±3.08 0.1347	94.94±2.55 0.0146	94.72±2.34 0.0055
Dk	**98.07±0.36** 0.0354	95.25±1.08 0.0662	97.89±1.31 0.1499	97.71±0.90 0.2795	97.54±1.04 0.0332	97.89±0.90 0.0099
Dl	80.82±1.06 0.0100	79.34±3.42 0.0224	**81.37±3.03** 0.0263	81.33±1.85 0.0255	80.39±4.45 0.0075	81.33±1.85 0.0049
Win/Tie/Loss	-	11/1/0	9/1/2	11/0/1	11/1/0	10/0/2

TABLE V
THE TESTING RESULTS OF SIX MODELS ON MULTICLASS UCI DATASETS

Dataset	ETMDM Accuracy (%) Time (s)	LDM Accuracy (%) Time (s)	TSVM Accuracy (%) Time (s)	SVM Accuracy (%) Time (s)	LSTSVM Accuracy (%) Time (s)	LSSVM Accuracy (%) Time (s)
D0	**70.12±7.61** 0.0490	55.11±8.94 0.0656	68.22±4.36 0.0932	65.42±5.20 0.0795	62.79±4.36 0.0267	62.15±0 0.0225
D1	**98.87±1.38** 0.0243	93.83±2.06 0.0330	**98.87±1.38** 0.0412	97.19±2.52 0.0324	**98.87±1.38** 0.0122	97.75±0 0.0101
D2	**96.00±5.83** 0.0293	95.00±5.48 0.0523	**96.00±5.83** 0.0833	94.05±4.91 0.0548	95.00±5.48 0.0335	95.00±5.48 0.0297
D3	97.77±1.88 0.0477	82.15±2.53 0.0988	96.09±1.88 0.0991	96.11±1.02 0.1205	95.83±1.75 0.0414	**98.33±1.62** 0.0267
D4	**91.84±0.57** 0.0867	90.09±1.73 0.1314	91.52±0.93 0.5111	**91.84±0.29** 0.5378	**91.84±0.57** 0.0414	91.52±0.93 0.0224
D5	**98.67±1.63** 0.0144	96.00±3.27 0.0262	98.00±1.63 0.0386	97.33±2.49 0.0317	97.33±3.89 0.0125	97.33±2.49 0.0104
D6	**87.49±2.72** 0.2518	70.27±3.68 0.4490	86.36±3.48 0.5643	86.08±3.08 0.5599	86.90±3.29 0.2283	87.19±2.97 0.1459
D7	**100±0** 0.0181	**100±0** 0.0289	**100±0** 0.0377	**100±0** 0.0216	**100±0** 0.0160	**100±0** 0.0137
D8	95.83±0.47 0.1465	21.79±2.38 0.4932	93.56±1.96 0.9405	63.76±14.31 0.9437	97.17±1.17 0.1094	**97.93±1.20** 0.0786
D9	84.40±2.05 0.1548	60.75±1.42 0.5612	83.22±1.76 0.5423	80.03±3.28 1.0063	84.63±1.83 0.1022	**85.46±1.19** 0.0596
Win/Tie/Loss	-	9/1/0	7/3/0	8/2/0	5/3/2	5/2/3

In addition, as shown in Fig. 2(a), the CPU time of nonlinear ETMDM is longer than that of linear one, because the nonlinear kernel requires extra computing time. However, it can be observed from Fig. 2(b) that the accuracies of nonlinear ETMDM are better than those of the linear one on nine of twelve binary datasets. This means that the ETMDM with nonlinear kernel has more outstanding performance.

In view of the above experimental conclusions, Table V only shows the testing results of nonlinear models with better performance for multiclass datasets. In terms of accuracy, we observe from Tables V and VI that the ETMDM performs better on seven of ten datasets and achieves the best average rank. The results of win/tie/loss show that the ETMDM wins compared to any of the other nonlinear models. Thus, we can confirm that the ETMDM is superior in accuracy for multi-class datasets. In terms of CPU time, the ETMDM is also excellent, but inferior to LSSVM and LSTSVM with equality constraints. For the average rank of CPU time, the same conclusions are obtained. This is because it requires more matrix operations to obtain higher accuracy. Combining the test results in accuracy and CPU time for two-class datasets and multi-class datasets, we draw the conclusion that the ETMDM is robust in generalization performance and strong in computational performance.

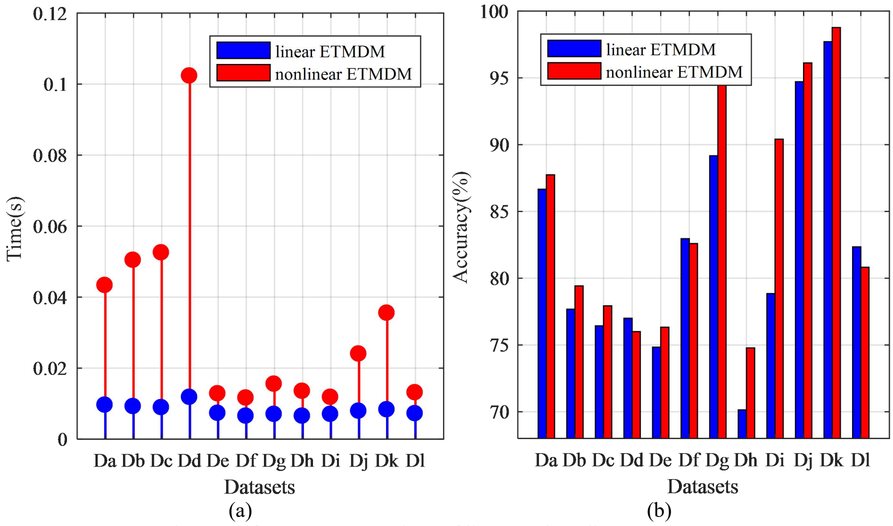


Fig. 2. Performance comparison of linear and nonlinear ETMDMs.

TABLE VI
THE RANKS OF SIX MODELS ON MULTICLASS UCI DATASETS

Dataset	Ranks (Accuracy / CPU time)					
	ETMDM	LDM	TSVM	SVM	LSTSVM	LSSVM
D0	1/3	6/4	2/6	3/5	4/2	5/1
D1	2/3	6/5	2/6	5/4	2/2	4/1
D2	1.5/1	4/4	1.5/6	6/5	4/3	4/2
D3	2/3	6/4	4/5	3/6	5/2	1/1
D4	2/3	6/4	4.5/5	2/6	2/2	4.5/1
D5	1/3	6/4	2/6	4/5	4/2	4/1
D6	1/3	6/4	4/6	5/5	3/2	2/1
D7	3.5/3	3.5/5	3.5/6	3.5/4	3.5/2	3.5/1
D8	3/3	6/4	4/5	5/6	2/2	1/1
D9	3/3	6/5	4/4	5/6	2/2	1/1
Average rank	2/2.8	5.55/4.3	3.15/5.5	4.15/5.2	3.15/2.1	3/1.1

B. *NDC datasets*

In this subsection, the experiments on large-scale datasets are performed. The NDC datasets [37] with different numbers of data points are used to explore the computational efficiency and classification accuracy. The NDC datasets are divided into training and testing samples according to Table VII.

TABLE VII
THE BRIEF INFORMATION OF NDC DATASETS

Dataset	Training samples	Testing samples	Features
NDC-500	500	100	32
NDC-1K	1000	200	32
NDC-2K	2000	400	32
NDC-3K	3000	600	32
NDC-4K	4000	800	32
NDC-5K	5000	1000	32
NDC-10K	10000	2000	32

The testing accuracies for six nonlinear models are presented in Table VIII. It can be seen that the ETMDM achieves the best accuracies on six of seven NDC datasets. Moreover, the accuracy of ETMDM is better than the other five models on NDC-500, NDC-1K and NDC-10K datasets. For NDC-4K dataset, the accuracy of ETMDM is only lower than that of LDM. The win/tie/loss results are recorded in the last row of Table VIII. It is easy to see that the ETMDM wins compared to any of the other models. Therefore, we can conclude that our ETMDM has advantages over other models for large-scale datasets in terms of classification accuracy. There are some reasons to explain this conclusion. Firstly, the ETMDM explores the margin distribution information of samples more comprehensively. Secondly, the ETMDM adopts the twin structure model to improve the adaptability of data distribution. Thirdly, the weighted linear loss is used to avoid suffering from the possible negative infinity problem.

It can also be observed form Table IX that the CPU time of ETMDM, LSSVM and LSTSVM is significantly shorter than that of LDM, SVM and TSVM. The reason is that the LDM, SVM and LDM are solved by using QP solver, while the ETMDM, LSSVM and LSTSVM are solved by solving a linear equation system. Compared with LSSVM and LSTSVM, the ETMDM needs to spend more time to calculate the margin distribution information. Thus, the CPU time of ETMDM is longer than that of LSSVM and LSTSVM.

Table X shows that the ranks of six models in terms of accuracy and CPU time. It can be seen that the ETMDM achieves the best average rank in terms of accuracy. In terms of CPU time, the ETMDM is also excellent, but inferior to LSSVM and LSTSVM with equality constraints. Fig. 3 shows the CPU time of six models for different numbers of data in NDC datasets. It can be seen that the larger the number of data, the longer the CPU time of each model becomes. Moreover, with the increase of the number of data, the CPU time of SVM, LDM and TSVM increases more significantly. This means that the models with inequality constraints are not suitable for large-scale dataset in terms of computational efficiency. Because LSSVM, LSTSVM and ETMDM employ equality constraints, they are suitable for large datasets.

TABLE VIII
THE ACCURACIES AND RANKS OF SIX MODELS ON NDC DATASETS

Dataset	ETMDM Accuracy (%)	LDM Accuracy (%)	TSVM Accuracy (%)	SVM Accuracy (%)	LSTSVM Accuracy (%)	LSSVM Accuracy (%)
NDC-500	**97.00**	92.00	92.00	92.00	96.00	96.00
NDC-1K	**99.00**	96.50	97.50	97.00	98.00	97.00
NDC-2K	**99.75**	99.25	**99.75**	**99.75**	**99.75**	**99.75**
NDC-3K	**99.17**	97.67	99.00	98.83	99.00	**99.17**
NDC-4K	99.12	**99.25**	99.12	99.12	99.12	98.25
NDC-5K	**99.50**	97.90	99.10	98.20	**99.50**	99.40
NDC-10K	**99.60**	99.10	99.30	99.20	99.50	99.50
Win/Tie/Loss	-	6/0/1	5/2/0	5/2/0	4/3/0	5/2/0

TABLE IX
THE CPU TIME OF SIX MODELS ON NDC DATASETS

Dataset	ETMDM Time (s)	LDM Time (s)	TSVM Time (s)	SVM Time (s)	LSTSVM Time (s)	LSSVM Time (s)
NDC-500	0.0337	0.0780	0.1209	0.2333	0.0369	0.0135
NDC-1K	0.1454	0.2971	0.3977	0.6195	0.1308	0.0466
NDC-2K	0.8075	1.4022	2.1526	4.0086	0.4873	0.2303
NDC-3K	2.1445	3.4338	4.5345	12.0435	1.0591	0.6092
NDC-4K	4.4963	8.3894	7.0301	16.3773	1.9264	1.4276
NDC-5K	8.1525	16.8471	21.0774	37.1381	5.5806	2.5196
NDC-10K	39.7305	96.6157	67.7052	208.7702	28.7949	17.3366

TABLE X
THE RANKS OF SIX MODELS ON NDC DATASETS

Dataset	Ranks (Accuracy / CPU time)					
	ETMDM	LDM	TSVM	SVM	LSTSVM	LSSVM
NDC-500	1/2	5/4	5/5	5 /6	2.5/3	2.5/1
NDC-1K	1/3	6/4	3/5	4.5/6	2/2	4.5/1
NDC-2K	3/3	6/4	3/5	3/6	3/2	3/1
NDC-3K	1.5/3	6/4	3.5/5	5/6	3.5/2	1.5/1
NDC-4K	3.5/3	1/5	3.5/4	3.5/6	3.5/2	6/1
NDC-5K	1.5/3	6/4	4/5	5/6	1.5/2	3/1
NDC-10K	1/3	6/5	4/4	5/6	2/2	3/1
Average rank	1.8/2.9	5.1/4.3	3.7/4.7	4.4/6	2.6/2.1	3.4/1

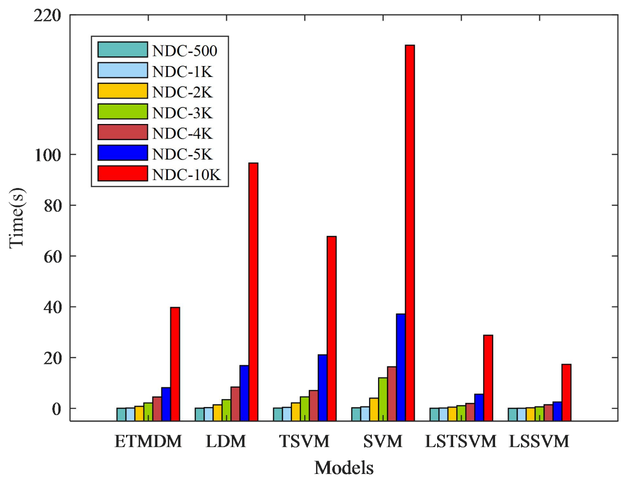


Fig. 3. The CPU time of six models for different numbers of data on NDC datasets.

C. NEU dataset

In this subsection, we further investigate the ETMDM for real data collected from industrial production lines. The NEU dataset from Northeastern University surface defect database [38] is selected. There are six kinds of typical surface defects of the hot-rolled steel strip in NEU dataset, i.e., crazing (CZ), inclusion (IN), patches (PH), pitted-surface (PE), rolled-in-scale (RD) and scratches (SH). Moreover, these defects are described by 35-dimensional features. 3781 defect samples are used in this experiment and are randomly divided into training samples and testing samples. And Training samples and testing samples accounted for 75% and 25% of the total, respectively. In addition, the binary tree technique is used to solve the multi-class classification problem of NEU dataset.

To highlight the testing result of each type of defect, the confusion matrices are presented for six models, as shown in Fig. 4. The testing accuracy of each type of defect can be seen from the diagonal of the confusion matrix. The ETMDM achieves the best accuracies for SH, IN and RD defects. Although the ETMDM is not the best for PH, PE and CZ defects, the difference in accuracy between it and other models is very small. However, it can be seen from Table XI that the average accuracy rank of ETMDM is the best. This shows that the ETMDM has advantage in total classification accuracy compared with other models. Fig. 5 shows the total accuracies and total CPU time for all models on NEU dataset. The total accuracies of ETMDM are much higher than those of LDM, SVM and LTSVM. And in terms of total CPU time, the ETMDM has obvious advantages over LDM and SVM. Compared with TSVM, the ETMDM is significantly better in total CPU time, along with slightly better in total accuracy. The ETLDM and LSSVM have their own advantage and disadvantage in performance. In summary, our ETMDM also has excellent performance for industrial dataset.

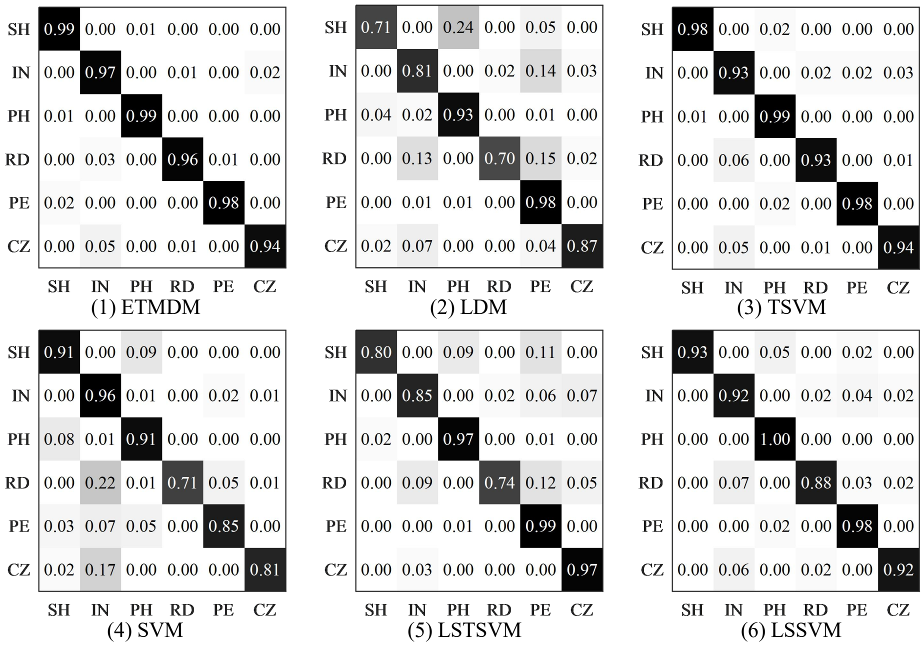


Fig. 4. The confusion matrices for six models.

TABLE XI
THE ACCURACY RANKS OF SIX MODELS ON NEU DATASET

Defects	Ranks					
	ETMDM	LDM	TSVM	SVM	LSTSVM	LSSVM
SH	1	6	2	4	5	3
IN	1	6	3	2	5	4
PH	2.5	5	2.5	6	4	1
RD	1	6	2	5	4	3
PE	3.5	3.5	3.5	6	1	3.5
CZ	2.5	5	2.5	6	1	4
Average rank	1.9	5.3	2.6	4.8	3.3	3.1

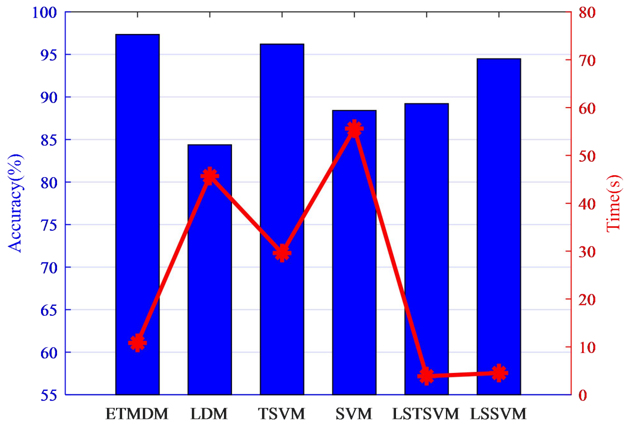


Fig. 5. Total accuracies and total CPU time for six models on NEU dataset.

D. *Face datasets*

To further verify the robustness of ETMDM, the experiments are carried out on two face datasets. These two face datasets are ORL[39] and Yale[40]. The ORL dataset contains 40 people, with 10 face images for each person, totaling 400 images. They have different facial details, light brightness and facial expressions. The size of each image is 112*92. In the experiment, we randomly select 5 images for each person as training samples, and the remaining ones are used as test samples. Therefore, there are 200 images for training and 200 images for testing respectively. The Yale dataset contains 15 people, with 11 face images for each person, totaling 165 images. They have different postures and different lighting conditions. The size of each image is 137*147. In the experiment, we randomly select 6 images for each person as training samples, and the rest are used as test samples. Therefore, there were 90 and 75 images for training and testing respectively. Furthermore, the binary tree technique is also used for multi-category classification. the experiments are conducted on ETMDM and the other five comparison models, and the final results are shown in Table XII. The experimental results show that on two face datasets, the ETMDM algorithm proposed in this paper achieves the highest classification accuracy compared with the other five algorithms. It further indicates that our model has strong generalization performance.

TABLE XII
THE ACCURACIES OF SIX MODELS ON FACE DATASETS

Models	ORL	Yale
ETMDM	**98.33**	**93.91**
LDM	97.99	92.81
TSVM	98.03	90.96
SVM	97.71	92.31
LSTSVM	97.28	91.37
LSSVM	96.32	92.56

E. *USPS dataset*

The USPS dataset [41] consists of grayscale handwritten digit images ranging from 0 to 9. Each number contains 1,100 images, and the size of each image is 16*16. Here we select 6 different groups of images and conduct a binary classification experiment using the one-vs-one method. In the experiment, 550 samples were used for training and the rest for testing. The classification results are recorded in Table XIII. It can be seen from the experimental results that in most cases, ETMDM is better than the other five algorithms. This indicates that our ETMDM has robust classification performance.

TABLE XIII
THE ACCURACIES OF SIX MODELS ON USPS DATASETS

Models	0 vs 1 (%)	1 vs 2 (%)	3 vs 8 (%)	4 vs 7 (%)	5 vs 8 (%)	6 vs 9 (%)
ETMDM	**99.82**	**98.91**	97.70	99.03	**99.82**	**99.94**
LDM	99.76	98.67	**97.76**	98.67	99.64	99.88
TSVM	99.70	98.55	97.52	**99.15**	99.76	99.76
SVM	99.39	98.61	96.97	98.73	99.58	98.67
LSTSVM	99.03	98.65	96.97	98.46	99.33	98.70
LSSVM	98.73	98.11	97.64	98.06	99.17	98.97

V. CONCLUSIONS

In this paper, we present a novel model with equality constraints, named as ETMDM, for pattern classification. The ETMDM inherits the twin structure of TSVM. Instead of boundary hyperplanes, the margin hyperplanes are adopted. The advantage is that the ETMDM no longer requires inequality constraints. Moreover, the margin hyperplanes are used to capture the margin distribution information. Although the margin mean and margin variance in LDM are used to generate the margin hyperplanes, they are reconstructed. For example, the weighted linear loss of margin mean is employed to avoid the negative infinity problem and some redundant terms of margin variance are eliminated. For these reasons, the ETMDM is an optimized model with excellent generalization performance and high computational efficiency. The experiments on UCI datasets show that the ETMDM is robust in generalization performance. The experiments on NDC datasets indicate that the ETMDM is suitable for large datasets. The testing results on NEU dataset confirm that the ETMDM has excellent performance for industrial data. In terms of computational efficiency, the ETMDM does not outperform the models with equality constraints. This is because it takes more time to calculate the margin distribution information to obtain higher accuracy. So, it is interesting to further improve the computational efficiency of ETMDM in the future.

REFERENCES

[1] M. Mohammadi, T.A. Rashid, S.H.T. Karim, A.H.M. Aldalwie, Q.T. Tho, M. Bidaki, A.M. Rahmani, M. Hosseinzadeh, A comprehensive survey and taxonomy of the SVM-based intrusion detection systems, Journal of Network and Computer Applications 178 (2021) 102983.

[2] T. Cuong-Le, T. Nghia-Nguyen, S. Khatir, P. Trong-Nguyen, S. Mirjalili, K.D. Nguyen, An efficient approach for damage identification based on improved machine learning using PSO-SVM, Engineering with Computers 38 (2022) 3069–3084.

[3] R. Loganathan, S. Latha, Optimization of support vector machines performance using OCT images, IAENG International Journal of Computer Science, 51 (12) (2024) 1996-2009.

[4] M.X. Chu, Y. Feng, Y.H. Yang, X. Deng, Multi-class classification method for steel surface defects with feature noise, Journal of Iron and Steel Research International 28 (2021) 303-315.

[5] A. Manoharan, K.M. Begam, V.R. Aparow, D. Sooriamoorthy, Artificial neural networks, gradient boosting and support vector machines for electric vehicle battery state estimation: A review, Journal of Energy Storage 55 (2022) 105384.

[6] Z. Rao, C. Feng, Automatic identification of chords in noisy music using temporal correlation support vector machine, IAENG International Journal of Computer Science, 50 (2) (2023) 813-819.

[7] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20 (3) (1995) 273-297.

[8] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.

[9] L. Tang, Y. Tian, W. Li, P.M. Pardalos, Structural improved regular simplex support vector machine for multiclass classification, Applied Soft Computing 91 (2020) 106235.

[10] H. Wang, Y. Shao, S. Zhou, C. Zhang, N. Xiu, Support vector machine classifier via L-0/1 soft-margin loss, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (10) (2022) 7253-7265.

[11] X. Peng, A spheres-based support vector machine for pattern classification, Neural Computing and Applications 31 (2019) 379-396.

[12] X. Huang, L. Shi, J.A.K. Suykens, Support vector machine classifier with pinball loss, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (5) (2014) 984-997.

[13] F. Marchetti, E. Perracchione, Local-to-global support vector machines (LGSVMs), Pattern Recognition 132 (2022) 108920.

[14] C.F. Lin, S.D. Wang, Fuzzy support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 464-471.

[15] B. Chen, Y. Fan, W. Lan, J. Liu, C. Cao, Y. Gao, Fuzzy support vector machine with graph for classifying imbalanced datasets, Neurocomputing 514 (2022) 296-312.

[16] Jayadeva, R. Khemchandani, S. Chandra. Twin support vector machines for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (5) (2007) 905-910.

[17] Y.H. Shao, C.H. Zhang, X.B. Wang, N.Y. Deng, Improvements on twin support vector machines, IEEE Transactions on Neural Networks 22 (6) (2011) 962-968.

[18] Y. Tian, Z. Qi, X. Ju, Y. Shi, X. Liu, Nonparallel support vector machines for pattern classification, IEEE Transactions on Cybernetics 44 (7) (2014) 1067-1079.

[19] R. Khemchandani, P. Saigal, S. Chandra, Angle-based twin support vector machine, Annals of Operations Research 269 (2018) 387-417.

[20] H. Wang, Y. Xu, Z. Zhou, Twin-parametric margin support vector machine with truncated pinball loss, Neural Computing and Applications 33 (2021) 3781-3798.

[21] X. Hua, S. Xu, J. Gao, S. Ding, L1-norm loss-based projection twin support vector machine for binary classification, Soft Computing 23 (2019) 10649-10659.

[22] Y. An, H. Xue, Indefinite twin support vector machine with DC functions programming, Pattern Recognition 121 (2022) 108195.

[23] Z. Liang, L. Zhang, Uncertainty-aware twin support vector machines, Pattern Recognition 129 (2022) 108706.

[24] W. Gao, Z.H. Zhou, On the doubt about margin explanation of boosting, Artificial Intelligence 203 (2013) 1-18.

[25] L. Breiman, Prediction games and arcing classifiers, Neural Computation 11 (7) (1999) 1493-1517.

[26] T. Zhang, Z.H. Zhou, Large margin distribution machine, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 313–322.

[27] F. Cheng, J. Zhang, C.Wen, Cost-sensitive large margin distribution machine for classification of imbalanced data, Pattern Recognition Letters 80 (2016) 107-112.

[28] L. Liu, M. Chu, Y. Yang, R. Gong, Twin support vector machine based on adjustable large margin distribution for pattern classification, International Journal of Machine Learning and Cybernetics 11 (2020) 2371-2389.

[29] S. Abe, Unconstrained large margin distribution machines, Pattern Recognition Letters 98 (2017) 96-102.

[30] J. Zhou, Y. Tian, J. Luo, Q. Zhai, Laplacian large margin distribution machine for semi-supervised classification, Journal of the Operational Research Society 73 (8) (2022) 1889-1904.

[31] B.B. Hazarika, D. Gupta, Improved twin bounded large margin distribution machines for binary classification, Multimedia Tools and Applications 82 (2023) 13341-13368.

[32] J.A.K. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Processing Letters 9 (1999) 293-300.

[33] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Systems with Applications 36 (4) (2009) 7535-7543.

[34] U. Gupta, D. Gupta, Least squares large margin distribution machine for regression, Applied Intelligence 51 (2021) 7058-7093.

[35] Y.H. Shao, W.J. Chen, Z. Wang, C.N. Li, N.Y. Deng, Weighted linear loss twin support vector machine for large-scale classification, Knowledge-Based Systems 73 (2015) 276-288.

[36] D. Dua, E.K. Taniskidou, UCI machine learning repository, 2017. [Online]. Available: http://archive.ics.uci.edu/ml/.

[37] D.R. Musicant, NDC: Normally Distributed Clustered Datasets, 1998. [Online]. Available: http://www.cs.wisc.edu/dmi/svm/ndc/.

[38] Y. He, K. Song, Q. Meng, Y.Yan, An end-to-end steel surface defect detection approach via fusing multiple hierarchical features, IEEE Transactions on Instrumentation and Measuremente 69 (4) (2020) 1493-1504.

[39] F.S. Samaria, A.C. Harter, ORL Database of Faces, 1994. [Online]. Available: https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[40] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Yale Face Database, 1997. [Online]. Available: http://vision.ucsd.edu/content/yale-face-database.

[41] J.J. Hull, USPS Handwritten Digit Database, 1994. [Online]. Available: http://www.cs.nyu.edu/~roweis/data.html.