Chinese Journal of Electronics
vol. 33, no. 3, pp. 753–765, May 2024
https://doi.org/10.23919/cje.2022.00.156

RESEARCH ARTICLE

Weighted Linear Loss Large Margin Distribution Machine for Pattern Classification

Ling LIU, Maoxiang CHU, Rongfen GONG, Liming LIU, and Yonghui YANG

School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China

Corresponding author: Rongfen GONG, Email: fx_gong@hotmail.com

Manuscript Received May 31, 2022; Accepted August 7, 2023

Abstract — Compared with support vector machine, large margin distribution machine (LDM) has better generalization performance. The central idea of LDM is to maximize the margin mean and minimize the margin variance simultaneously. But the computational complexity of LDM is high. In order to reduce the computational complexity of LDM, a weighted linear loss LDM (WLLDM) is proposed. The framework of WLLDM is built based on LDM and the weighted linear loss. The weighted linear loss is adopted instead of the hinge loss in WLLDM. This modification can transform the quadratic programming problem into a simple linear equation, resulting in lower computational complexity. Thus, WLLDM has the potential to deal with large-scale datasets. The WLLDM is similar in principle to the LDM algorithm, which can optimize the margin distribution and achieve better generalization performance. The WLLDM algorithm is compared with other models by conducting experiments on different datasets. The experimental results show that the proposed WLLDM has better generalization performance and faster training speed.

Keywords — Support vector machine, Large margin distribution machine, Weighted linear loss, Pattern classification.

Citation — Ling LIU, Maoxiang CHU, Rongfen GONG, *et al.*, “Weighted Linear Loss Large Margin Distribution Machine for Pattern Classification,” *Chinese Journal of Electronics*, vol. 33, no. 3, pp. 753–765, 2024. doi: 10.23919/cje.2022.00.156.

I. Introduction

Support vector machine (SVM) [1] proposed by Vapnik *et al.*, is a kind of machine learning method. It obtains an optimal classification hyperplane by solving a quadratic programming problem (QPP). On the one hand, SVM obeys the structural risk minimization principle, which makes maximum margin be considered. On the other hand, the kernel trick can be directly used for solving the nonlinear classification problem. Owing to these advantages, SVM has been successfully applied in the fields [2]–[4] of pattern classification and regression analysis and many improved models [5]–[8] based on SVM are proposed by various scholars. All algorithms improve the SVM algorithm in efficiency and generalization performance to some extent.

Although SVM has many advantages, there are still two important problems that may affect its application. One problem is that SVM is based on the margin theory, which ignores the effect of margin distribution on the model. Reyzin *et al.*, [9] revealed that the margin distribution is more important for the generalization performance, rather than the minimum margin. Such a conjecture has been researched theoretically and has been proved by Gao *et al.* [10] recently. In addition, Zhou [11] pointed that both the margin mean and variance instead of a single-point margin are more crucial. Inspired by the above results, Zhang *et al.* [12] proposed a large margin distribution machine (LDM). LDM can improve the generalization ability of SVM-type classifiers more effectively. Therefore, LDM has been extensively studied [13]–[15] in pattern recognition field.

The other problem is that SVM and its variants are the high computational complexity, especially for large-scale datasets. As we know that SVM solves a large QPP costing around $O(m^3)$, where m is the number of training samples. With the increase of the number of samples, the computational complexity increases. In order to reduce the computational complexity, some improved algo-

Associate Editor: Prof. De-Chuan ZHAN, Nanjing University.

rithms have been proposed. For instance, Suykens *et al.* [16] proposed a least squares support vector machine (LSSVM) algorithm. LSSVM is implemented by solving a linear equation instead of solving a QPP in SVM, leading to a faster learning speed. Jayadeva *et al.* [17] proposed a twin-type SVM, called twin SVM (TSVM). TSVM is realized by solving two smaller QPPs rather than a QPP, leading to very fast training speed. Shao *et al.* [18], [19] proposed the weighted linear loss support vector machine (WLSVM) model and the weighted linear loss TSVM (WLTSVM) model. Both WLSVM and WLTSVM are constructing by defining a weighted linear loss. They only need to take less time to solve simple linear equations.

Similar to SVM, LDM also needs to solve a large QPP. In addition, the dual form in LDM involves matrix inversion. Thus, LDM is more time-consuming than SVM in theory. In fact, in order to reduce the time cost of LDM, Abe [20] proposed a least squares LDM (LSLDM) algorithm. The framework of LSLDM is built by using the least squares loss, which can speed up the training procedure than LDM. But Abe revealed that LSLDM is not a good formulation because very similar measures are mixed in the model. In this paper, following WLSVM [18] and WLTSVM [19], instead of using the hinge loss or least squares loss in LDM, we formulate a weighted linear loss LDM (WLLDM) algorithm. In WLLDM, the objective function is the sum of the margin mean, the margin variance, the regularization term and the slack variables, and includes the equality constraint term. The following is the discussion about advantages of WLLDM.

i) The weighted linear loss is used to measure the empirical risk of WLLDM. On the one hand, the weighted linear loss can weigh the impact of each sample on classification hyperplane, resulting in a more stable model. On the other hand, the use of weighted linear loss makes WLLDM solve a simple linear equation, rather than a QPP, leading to lower computational complexity.

ii) An additional regularization term is added in WLLDM. This is a common method to guarantee the global solution and stability of optimization problem. Strong convexity plays an important role on the very fast computational times obtained.

iii) The classification hyperplane fulfills the requirement that the positive and negative samples be on one side respectively. Thus, the boundary hyperplanes are not adopted in WLLDM.

iv) Inspired by the theorem in [21], the solution methods of linear WLLDM and nonlinear WLLDM are different.

II. Large Margin Distribution Machine

For a binary classification problem, assume that the training set is $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_m]^{\mathrm{T}}\in\mathbf{R}^{m\times d}$, and the corresponding label is $\boldsymbol{y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_m]^{\mathrm{T}}\in\mathbf{R}^{m}$, where $\boldsymbol{x}_1\in\mathbf{R}^{d}$, $\boldsymbol{y}_i\in\{+1,-1\}$, m is the number of training samples, and d is the dimension of the feature space.

Similar to SVM, LDM also searches for a classification hyperplane

$$\boldsymbol{w}^{\mathrm{T}}x=0 \tag{1}$$

where $\boldsymbol{w}\in\mathbf{R}^{d}$ is the normal vector. Differently, the classification hyperplane is obtained by maximizing the margin mean $\bar{r}$ and minimizing the margin variance $\hat{r}$. The formulas of $\bar{r}$ and $\hat{r}$ are as follows:

$$\bar{r}=\frac{1}{m}\sum_{i=1}^{m}y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i=\frac{1}{m}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w} \tag{2}$$

$$\begin{aligned}\hat{r}&=\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i-y_j\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_j\right)^2\\&=\frac{2}{m^2}(m\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w}-\boldsymbol{w}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{w})\end{aligned} \tag{3}$$

Then, the primal problem of LDM can be formulated as

$$\begin{aligned}\min_{\boldsymbol{w},\xi_i}\ &\frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+\lambda_1\hat{r}-\lambda_2\bar{r}+C\sum_{i=1}^{m}\xi_i\\ \text{s.t. }&y_i\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i\geq 1-\xi_i\\ &\xi_i\geq 0,\ i=1,2,\ldots,m\end{aligned} \tag{4}$$

where ξ_i denotes the slack variable, C, λ_1 and λ_2 are trading-off parameters among the training error, margin mean and margin variance. The primal problem (4) can be explained as the following. The first term implies that the structural risk principle is measured by the term $\frac{1}{2}||\boldsymbol{w}||^2$. The second and third terms are the maximization of margin mean and the minimization of margin variance, respectively, resulting in a better generalization performance. The last term and the inequality constraint are used to construct two boundary hyperplanes $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}=+1$ and $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}=-1$. And the classification hyperplane $\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}=0$ is located in the middle of two boundary hyper-planes. As can be observed, LDM also adopts the hinge loss, leading to high computational complexity.

III. Weighted Linear Loss LDM

Though LDM obtains perfect generalization performance by optimizing margin distribution, it is the high computational complexity. In order to reduce the time cost of LDM under the premise of high accuracy, the WLLDM model is proposed in this section. On the one hand, the weighted linear loss is applied to WLLDM instead of hinge loss, so the QPP can be replaced by a linear equation, resulting in lower computational complexity than LDM. On the other hand, WLLDM is similar in principle to the LDM, which can get better generalization ability than WLSVM and WLTSVM. Thus, WLLDM not only inherits almost all advantages of LDM but also has lower computational complexity.

1. Loss function

In the SVM-type classifiers, the commonly used loss functions include hinge loss, square loss and linear loss. The hinge loss is defined as

$$L_{\text{hinge}}(u) = \max\{0, 1 - yu\}, \quad y = \pm 1 \tag{5}$$

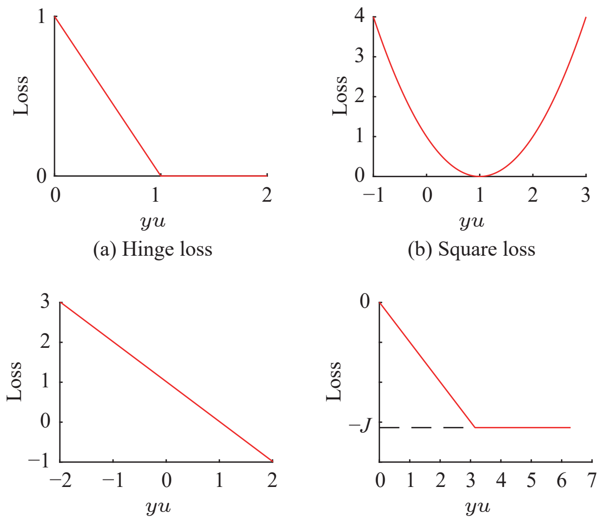
The square loss is defined as

$$L_{\text{s}}(u) = \sum (1 - yu)^2, \quad y = \pm 1 \tag{6}$$

And the linear loss is defined as

$$L_{\text{l}}(u) = 1 - yu, \quad y = \pm 1 \tag{7}$$

where u is the decision function, and y is the corresponding label of a sample $\boldsymbol{x}$. Figure 1 are examples of the above loss functions.



(a) Hinge loss

(b) Square loss

(c) Linear loss

(d) Weighted linear loss

Figure 1 Different loss functions.

Simiar to SVM, to measure the empirical risk, LDM also adopts the hinge loss, leading to high computational complexity. In terms of SVM-type classifiers, there are some models with high computational speed, such as LSSVM. The reason is that they are constructed by using the linear loss, leading to the QPPs with equality constraints be solved. Therefore, one way to solve the problem of low computational efficiency is to apply equality constraints in LDM, such as least squares LDM (LSLDM).

As you can see from Figure 1(c), the optimization problem based on linear loss may suffer the negative infinity problem. So, in this section, the linear loss is limited by weighted parameter v_i, and a weighted linear loss (seen in Figure 1(d)) is defined as

$$L_{wl}(u) = v_i(1 - yu), \quad y = \pm 1 \tag{8}$$

where v_i is formulated as

$$v_i = \begin{cases} 10^{-4}, & |1 - yu| \geq J \\ 1, & |1 - yu| < J \end{cases} \tag{9}$$

where $J \geq 0$ is a constant.

2. Linear WLLDM

In spirt of the characteristics of weighted linear loss and LDM, the linear WLLDM can be formulated as

$$\min_{\boldsymbol{w},b} \frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + b^2) + \lambda_1 \hat{r} - \lambda_2 \bar{r} + c_1 \sum_{i=1}^{m} L_{wl}(f(\boldsymbol{x}_i)) \tag{10}$$

where c_1, λ_1 and λ_2 are positive plenty parameters, b is a bias, $f(\boldsymbol{x}) : \boldsymbol{w}^T\boldsymbol{x} + b = 0$ is the classification hyperplane.

Based on (8), the problem (10) can be further equivalently transformed into

$$\begin{aligned} &\min_{\boldsymbol{w},\xi_i} \frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + b^2) + \lambda_1 \hat{r} - \lambda_2 \bar{r} + c_1 \sum_{i=1}^{m} v_i \xi_i \\ &\text{s.t.}\ \ y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = 0 - \xi_i \end{aligned} \tag{11}$$

where ξ_i is the error variable of sample $\boldsymbol{x}_i$, and v_i can be derived from formula (9). $\bar{r}$ and $\hat{r}$ can be derived from (12) and (13).

$$\bar{r} = \frac{1}{m}\sum_{i=1}^{m} y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) = \frac{1}{m}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y})^{\mathrm{T}}\boldsymbol{w} + \frac{1}{m}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{e}b \tag{12}$$

$$\hat{r} = \frac{1}{m}\sum_{i=1}^{m} \left[y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b) - \bar{r}\right]^2 = \frac{1}{m^2}(\boldsymbol{\Delta_1} - \boldsymbol{\Delta_2}) \tag{13}$$

where $\boldsymbol{\Delta_1} = m(\boldsymbol{X}\boldsymbol{w} + \boldsymbol{e}b)^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} + \boldsymbol{e}b)$ and $\boldsymbol{\Delta_2} = (\boldsymbol{X}\boldsymbol{w} + \boldsymbol{e}b)^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w} + \boldsymbol{e}b))$

Next, the illustration of formula (11) is given. The object function includes four terms. The first one is the regularization term, which can realize the structural risk minimization. The second and third terms represent the margin distribution described by margin variance and margin mean. WLLDM tries to maximize the margin mean and minimize the margin variance to improve the generalization performance. The last term is to minimize the training errors, which means the mis-classification is allowed and the errors should be as small as possible. The weighted information v_i not only be used to avoid the negative infinity problem, but also balance the influence of each sample on the hyperplane $f(\boldsymbol{x})$. Moreover, the inequality constraints are replaced by the equality constraints, improving the computational efficiency.

In order to obtain the optimal solution of (11), the Lagrange function is constructed as the following:

$$\begin{aligned} L = &\frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + b^2) + \frac{\lambda_1}{m^2}(\boldsymbol{\Delta_1} - \boldsymbol{\Delta_2}) \\ &- \frac{\lambda_2}{m}[\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{w} + \boldsymbol{y}^{\mathrm{T}}\boldsymbol{e}b] + c_1\boldsymbol{v}^{\mathrm{T}}\boldsymbol{\xi} \end{aligned} \tag{14}$$

where $\boldsymbol{\xi} = -\boldsymbol{Y}\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}b$, $\boldsymbol{Y} = \mathrm{diag}(\boldsymbol{y})$ and $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2,$

$\ldots, \boldsymbol{v}_m)^{\mathrm{T}}$ is the weight vector.

Based on the KKT conditions, we have

$$\frac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{w} + \frac{2\lambda_1}{m^2}\coprod - \frac{\lambda_2}{m}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} - c_1\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v} = 0 \tag{15}$$

$$\frac{\partial L}{\partial b} = b + \frac{2\lambda_1}{m^2}\prod - \frac{\lambda_2}{m}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y} - c_1\boldsymbol{y}^{\mathrm{T}}\boldsymbol{v} = 0 \tag{16}$$

where $\coprod = m\boldsymbol{X}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}+\boldsymbol{e}b) - \boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}+\boldsymbol{e}b)$ and $\prod = m\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}+\boldsymbol{e}b) - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{X}\boldsymbol{w}+\boldsymbol{e}b)$.

Then, the combined matrix form of (15) and (16) is

$$\begin{bmatrix} \boldsymbol{P} & \boldsymbol{Q} \\ \boldsymbol{Q}^{\mathrm{T}} & \boldsymbol{N} \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix} = \begin{bmatrix} \dfrac{\lambda_2}{m}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + c_1\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v} \\ \dfrac{\lambda_2}{m}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y} + c_1\boldsymbol{y}^{\mathrm{T}}\boldsymbol{v} \end{bmatrix} \tag{17}$$

where $\boldsymbol{P}=\boldsymbol{I}+\frac{2\lambda_1}{m^2}(m\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{X})$, $\boldsymbol{Q}=\frac{2\lambda_1}{m^2}(m\boldsymbol{X}^{\mathrm{T}}\boldsymbol{e}-\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{e})$, $\boldsymbol{N}=1+\frac{2\lambda_1}{m^2}(m\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e}-\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\boldsymbol{e})$ and $\boldsymbol{I}$ is the identity matrices with appropriate dimensions.

Later, we can obtain the solution $\begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}$ as follows:

$$\begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{P} & \boldsymbol{Q} \\ \boldsymbol{Q}^{\mathrm{T}} & \boldsymbol{N} \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\lambda_2}{m}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y} + c_1\boldsymbol{X}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v} \\ \dfrac{\lambda_2}{m}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y} + c_1\boldsymbol{y}^{\mathrm{T}}\boldsymbol{v} \end{bmatrix} \tag{18}$$

Once the optimal solution is obtained, an unknown sample $\boldsymbol{x}_t$ can be classified by the following decision function:

$$\mathrm{label}(\boldsymbol{x}_t) = \mathrm{sgn}\left\{ [\boldsymbol{x}_t, 1] \begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix} \right\} \tag{19}$$

The detailed process of linear WLLDM is described in Algorithm 1.

Algorithm 1 Lagrange multiplier method solver for problem (11)

```
Require:
  $X$: training samples;
  $y$: label;
  $c_1$, $\lambda_1$, $\lambda_2$: appropriate parameters.
1: Let $v = e = (1, 1, \ldots, 1)^{\mathrm{T}}$;
2: Calculate the initial solution $\begin{bmatrix} w \\ b \end{bmatrix}$ by equation (18);
3: Based on $\begin{bmatrix} w \\ b \end{bmatrix}$, calculate the slack variable $\xi$, that is,
   $\xi = -YXw - yb$;
4: According to the slack variable $\xi$, we can get $J = |\xi|^{\mathrm{mean}}$,
   then the $v_i$ is obtained by equation (9);
5: Based on $v_i$, the optimal solution $\begin{bmatrix} w \\ b \end{bmatrix}$ is calculated by
   (18);
6: According to the solution, the decision function in equation (19) is constructed.
```

3. Nonlinear WLLDM

For nonlinear case, a mapping function $\varPhi(\cdot)$ is considered, which maps the input sample $\boldsymbol{x}$ from the original space to the Hibert space, i.e., $\boldsymbol{x} \to \varPhi(\boldsymbol{x})$. Let $\boldsymbol{X}_{\varPhi} = [\varPhi(\boldsymbol{x}_1), \varPhi(\boldsymbol{x}_2), \ldots, \varPhi(\boldsymbol{x}_m)] \in \mathbf{H}^{d\times m}$ be the training set in Hilbert space. Then, the primal formulation of nonlinear WLLDM can be described as

$$\begin{aligned} &\min_{\boldsymbol{w},\xi_i} \frac{1}{2}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w}+b^2) + \lambda_1\hat{r} - \lambda_2\bar{r} + c_1\sum_{i=1}^{m} v_i\xi_i \\ &\text{s.t. } \ y_i(\boldsymbol{w}^{\mathrm{T}}\varPhi(\boldsymbol{x}_i)+b) = 0 - \xi_i \end{aligned} \tag{20}$$

Inspired by the theorem in reference [21], for nonlinear case, let

$$\boldsymbol{w} = \boldsymbol{X}_{\Phi}^{\mathrm{T}}\boldsymbol{\alpha} \tag{21}$$

then, we have

$$\boldsymbol{X}_{\varPhi}\boldsymbol{w} = \psi(\boldsymbol{X}_{\varPhi}, \boldsymbol{X}_{\varPhi}^{\mathrm{T}})\boldsymbol{\alpha} = \boldsymbol{G}\boldsymbol{\alpha} \tag{22}$$

$$\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} = \boldsymbol{\alpha}^{\mathrm{T}}\psi(\boldsymbol{X}_{\varPhi}, \boldsymbol{X}_{\varPhi}^{\mathrm{T}})\boldsymbol{\alpha} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\alpha} \tag{23}$$

According to equations (22) and (23), the formula (20) can be reformulated as

$$\begin{aligned} &\min_{\boldsymbol{w},\xi_i} \frac{1}{2}(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\alpha}+b^2) + \frac{\lambda_1}{m^2}\boldsymbol{\varOmega} - \frac{\lambda_2}{m}\boldsymbol{\varpi} + c_1\sum_{i=1}^{m} v_i\xi_i \\ &\text{s.t. } \ y_i(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{X}_{\varPhi}\varPhi(\boldsymbol{x}_i)+b) = 0 - \xi_i \end{aligned} \tag{24}$$

where $\boldsymbol{\varTheta} = m(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b)^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b)$, $\odot = (\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b)^{\mathrm{T}}\times\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b)$, $\boldsymbol{\varpi} = \boldsymbol{y}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{y}^{\mathrm{T}}\boldsymbol{e}b$, $\boldsymbol{\varOmega} = \boldsymbol{\varTheta}-\odot$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_m]^{\mathrm{T}}$ is the Lagrange multiplier, $\boldsymbol{G}=\psi(\boldsymbol{X}_{\varPhi}, \boldsymbol{X}_{\varPhi}^{\mathrm{T}})$ and $\psi(\cdot,\cdot)$ is an appropriately kernel.

In order to obtain the optimal solution of (24), the Lagrange function is constructed as forllows:

$$\begin{aligned} L = &\frac{1}{2}(\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{\alpha}+b^2) + \frac{\lambda_1}{m^2}\boldsymbol{\varOmega} \\ &- \frac{\lambda_2}{m}\boldsymbol{\varpi} + c_1\boldsymbol{v}^{\mathrm{T}}(-\boldsymbol{Y}\boldsymbol{G}\boldsymbol{\alpha}-\boldsymbol{y}b) \end{aligned} \tag{25}$$

where $\boldsymbol{Y} = \mathrm{diag}(\boldsymbol{y})$, and $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m)^{\mathrm{T}}$ is the weight vector.

Based on the KKT conditions, we have

$$\frac{\partial L}{\partial \boldsymbol{\alpha}} = \boldsymbol{G}\boldsymbol{\alpha} + \frac{2\lambda_1}{m^2}\boldsymbol{\tau} - \frac{\lambda_2}{m}\boldsymbol{G}^{\mathrm{T}}\boldsymbol{y} - c_1\boldsymbol{G}^{\mathrm{T}}\boldsymbol{Y}\boldsymbol{v} = 0 \tag{26}$$

$$\frac{\partial L}{\partial b} = b + \frac{2\lambda_1}{m^2}\boldsymbol{\varrho} - \frac{\lambda_2}{m}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y} - c_1\boldsymbol{y}^{\mathrm{T}}\boldsymbol{v} = 0 \tag{27}$$

where $\boldsymbol{\tau} = m\boldsymbol{G}^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b) - \boldsymbol{G}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b)$ and $\boldsymbol{\varrho} = m\boldsymbol{e}^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b) - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}(\boldsymbol{G}\boldsymbol{\alpha}+\boldsymbol{e}b)$.

Then, the combined matrix form of formulas (26) and (27) is

$$\begin{bmatrix} \boldsymbol{P1} & \boldsymbol{Q1} \\ \boldsymbol{Q1}^{\mathrm{T}} & \boldsymbol{N1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \dfrac{\lambda_2}{m}\boldsymbol{Gy} + c_1\boldsymbol{GYv} \\ \dfrac{\lambda_2}{m}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y} + c_1\boldsymbol{y}^{\mathrm{T}}\boldsymbol{v} \end{bmatrix} \tag{28}$$

where $\boldsymbol{P1} = \boldsymbol{G} + \frac{2\lambda_1}{m^2}(m\boldsymbol{G}^{\mathrm{T}}\boldsymbol{G} - \boldsymbol{G}^{\mathrm{T}}\boldsymbol{yy}^{\mathrm{T}}\boldsymbol{G})$, $\boldsymbol{Q1} = \frac{2\lambda_1}{m^2} \times (m\boldsymbol{G}^{\mathrm{T}}\boldsymbol{e} - \boldsymbol{G}^{\mathrm{T}}\boldsymbol{yy}^{\mathrm{T}}\boldsymbol{e})$, $\boldsymbol{N1} = 1 + \frac{2\lambda_1}{m^2}(m\boldsymbol{e}^{\mathrm{T}}\boldsymbol{e} - \boldsymbol{e}^{\mathrm{T}}\boldsymbol{yy}^{\mathrm{T}}\boldsymbol{e})$.

Later, we can obtain the solution $\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix}$ as follows:

$$\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{P1} & \boldsymbol{Q1} \\ \boldsymbol{Q1}^{\mathrm{T}} & \boldsymbol{N1} \end{bmatrix}^{-1} \begin{bmatrix} \dfrac{\lambda_2}{m}\boldsymbol{Gy} + c_1\boldsymbol{GYv} \\ \dfrac{\lambda_2}{m}\boldsymbol{e}^{\mathrm{T}}\boldsymbol{y} + c_1\boldsymbol{y}^{\mathrm{T}}\boldsymbol{v} \end{bmatrix} \tag{29}$$

Applying the solution $\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix}$, the decision function of nonlinear WLLDM can be obtained as follows:

$$\mathrm{label}(\boldsymbol{x}_t) = \mathrm{sgn}\left\{ \begin{bmatrix} \psi(\boldsymbol{x}_t^{\mathrm{T}}, \boldsymbol{X}_{\Phi}^{\mathrm{T}}) & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix} \right\} \tag{30}$$

The detailed process of nonlinear WLLDM is described in Algorithm 2.

Algorithm 2 Lagrange multiplier method solver for problem (20)

```
Require:
  $\boldsymbol{X}_{\Phi}$: training samples;
  $\boldsymbol{y}$: label;
  $\psi(,)$: appropriate kernel;
  $c_1, \lambda_1, \lambda_2$: model parameters.
1: Let $\boldsymbol{v} = \boldsymbol{e} = (1, 1, \dots, 1)^{\mathrm{T}}$;
2: Calculate the initial solution $\begin{bmatrix} \boldsymbol{w} \\ b \end{bmatrix}$ by equation (29);
3: Based on $\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix}$,calculate the slack variable $\boldsymbol{\xi}$, that is,
   $\boldsymbol{\xi} = -\boldsymbol{YXw} - \boldsymbol{y}b$;
4: According to the slack variable $\boldsymbol{\xi}$, we can get $J = |\boldsymbol{\xi}|^{\mathrm{mean}}$,
   then the $v_i$ is obtained by equation (9);
5: Based on $v_i$, the optimal solution $\begin{bmatrix} \boldsymbol{\alpha} \\ b \end{bmatrix}$ is calculated by
   equaiton (29);
6: According to the solution, the decision function in equation (30) is constructed.
```

IV. Analysis of WLLDM

1. Realationship with SVM, LDM and LSLDM

Compared to standard SVM, LDM, LSLDM and WLLDM show better metrists in two aspects. On the one hand, LDM, LSLDM and WLLDM optimize the margin distribution of training samples. Hence, they are more accurate than SVM. On the other hand, LDM, LSLDM and WLLDM are more robust than SVM. The reason is that all training samples contribute to the classification hyperplane, which makes the hyperplane insensitive to noise.

Different from LDM and LSLDM, WLLDM change the inequality constraint into equality constraint, which is similar to LSLDM. Therefore, they solve a simple linear equation instead of a single QPP, which can lead to the reduction of time cost. Hence, both LSLDM and WLLDM are suitable for large-scale classification.

Although WLLDM and LSLDM solve a linear equation, the losses they used are markedly different. For LSLDM, the quadratic loss is adopted. Reference [20] points out that LSLDM is not a good formula because it mixes very similar measures. As for WLLDM, the weighted linear loss is used, which is built by adding the weights on linear loss. In WLLDM, the training samples in the different positions are given different penalties, which can avoid over-fitting to a certain extent and yield better generalization performance than linear loss.

2. Realationship with LSSVM, WLSVM and WLTSVM

For WLLDM, LSSVM, WLSVM and WLTSVM, what they have in common is that different loss functions are used in them rather than the hinge loss. Thus, they only need to solve simple linear equations, leading to a fast-training speed.

The difference is that, for LSSVM, the least squares loss is employed and the importance of each sample is the same. But some samples corrupted by noises are less significant and should be ignored. LSSVM lacks this kind of ability. However, for WLLDM, WLSVM and WLTSVM, they can weigh the effect of each sample on the hyperplane by the weighted vectors in the weighted linear loss. Compared to LSSVM, WLLDM, WLSVM and WLTSVM are more robust and accurate. Furthermore, compared with WLSVM and WLTSVM, our WLLDM can achieve better generalization performance. The reason is that the influence of the margin distribution on classifier is considered in WLLDM. Theoretical research indicates that the margin distribution is more important than minimum margin for SVM-style classifiers.

3. Computational complexity

Notice that some algorithms may have various ways to optimize. For example, SVM can be solved by stochastic gradient descent, sequential minimal optimization and dual coordinate descent. Hence, to make the comparation fair, in this section, all algorithms are solved by the optimization toolbox QP in MATLAB. From [17], we conclude that the complexity of the usual SVM is no more than $O(m^3)$. For LDM [12], as can be seen from its dual form, LDM needs to solve a vector with m variables. Thus, the time complexity of obtaining the optimal solution can reach to $O(m^3)$. Moreover, the matrix of dimension $d \times d$ needs to be inverted, which costs $O(d^3)$. Thus, the total cost for solving LDM is about $O(m^3+d^3)$. For optimal margin distribution machine (ODM), according to the [22], we found that a vector with $2m$ variables needs to be solved. Meanwhile, there is no matrix inversion in ODM. Thus, ODM costs around $O(8m^3)$. For WLLDM, all samples are used to optimize the QPP with equality constraints. WLLDM used all samples with weighted linear loss. So, the WLLDM has better computational efficiency than ODM, LDM and SVM. However, WLLDM is lower than LSSVM, WLSVM and LSLDM. The one reason is that LSSVM [16], WLSVM [18] and LSLDM [20] also solve the QPP with equality constraint. Also, WLLDM performs more matrix operations than LSSVM, WLSVM and LSLDM, such as $\boldsymbol{Q}$ and $\boldsymbol{P}$. The other reason is that WLLDM needs to train the model twice, which is similar to WLTSVM [19] and WLSVM. Therefore, compared with LSSVM, they are more time-consuming.

V. Experiments

In order to verify the classification performance of the proposed WLLDM, we compare WLLDM with some related algorithms on some different datasets, including UCI datasets [23], NDC dataset [24] and defect dataset [25], [26]. Before the experiments, each feature of all samples is normalized into [0, 1]. All algorithms are implemented in MATLAB 2016b. The software platform is installed on a PC with an Intel I7 processor (3.60 GHz) and 16 GB RAM. For the sake of fairness, all QPPs based on inequality constraints were solved using the built-in “quadprog” function in MATLAB. As for all QPPs based on equality constraints, they are directly implemented by solving linear equations in MATLAB.

The model parameters are important for the generalization performance of the model. In the experiments, the 5-fold cross-validation and the grid search are used to choose the optimal parameters. All penalty parameters in aforementioned algorithms are selected from $\{2^i|i=-7,-6,\dots,7\}$. For nonlinear case, the Gaussian kernel function $\psi(\boldsymbol{x}_i,\boldsymbol{x}_j)=\exp(-||\boldsymbol{x}_i-\boldsymbol{x}_j||^2/2\delta^2)$ is used, and the kernel parameter δ is chosen from $\{2^i|i=-7,-6,\dots,7\}$. For large-scale datasets, due to the long training time, the range of optimization parameters should be reduced accordingly. Since the SVM-type classifiers are binary classifiers for pattern recognition. For learning a multi-class problem, SVM-type classifiers must be extended to multi-class classifiers. For multiple class problems, the SVM-type classifiers are generally combined with “1 vs. 1”, “1 vs. rest”, “binary tree” and “directed acyclic graph”. Binary tree [27] is one of the most commonly used methods in multi-classification technology. Hence, in this paper, the binary tree method is combined with binary classifier to construct multi-class classification model.

1. UCI datasets

In order to verify the influence of margin distribution on performance, twenty-two benchmark datasets were selected from UCI datasets [23], including 12 two-category datasets and 10 multi-category datasets. Their attributes are listed in Table 1. Since the SVM-type classifiers are used to binary classification, for multiple class classification problem, the binary tree method is utilized. In the experiments, WLLDM is compared with SVM, LDM, ODM and LSLDM.

Table 1 The attributes of twenty-one benchmark datasets in UCI datasets

Datasets	Samples	Features	Classes
Australian	690	14	2
Blood	748	5	2
Diabetes	768	8	2
German	24	1000	2
Haberman	306	3	2
Heart	270	12	2
Ionosphere	351	34	2
Liverdisorder	345	6	2
Sonar	208	60	2
Vote	435	16	2
Wdbc	569	31	2
Wpbc	198	34	2
Air	359	65	3
Balance	625	5	3
Glass	214	10	6
Iris	150	5	3
Libras_Movement	360	90	15
Soybean	47	36	4
Vehicle	846	19	4
Vowel	528	11	11
Wine	178	13	3
Zoo	101	17	7

1) Binary classification

In this section, we conducted a series of experiments on twelve binary datasets. For each dataset, we randomly split the data into five subsets, one for testing and the others for training. This process is repeated five times until all of the five subsets have been set to be a testing set. Finally, the testing result is represented by the average result taken by five experiments. In the experiments, both the linear and nonlinear are considered. The testing results of five classifiers are recorded in Tables 2 and 3, including the accuracy and running time. “Accuracy” refers to the average accuracy with standard deviation taken by five-fold cross-validation. “Time” indicates the total time of training and testing.

In terms of time cost, some conclusions can be drawn from Tables 2 and 3. Firstly, WLLDM and LSLDM are faster than SVM, LDM and ODM. The reason is that they solve a simple linear equation instead of a QPP. Secondly, WLLDM is slower than LSLDM on some datasets. The reason is that WLLDM needs to

Table 2 The results of five linear classifiers on binary class UCI datasets

Dataset	WLLDM accuracy(%) Time(s)	LSLDM accuracy(%) Time(s)	ODM accuracy(%) Time(s)	LDM accuracy(%) Time(s)	SVM accuracy(%) Time(s)
Australian	**87.82±1.26** 0.0314	87.10±1.53 0.0094	87.68±1.5884 1.9338	87.54±1.54 0.1870	86.81±1.95 0.1573
Blood	**77.95±4.11** 0.0356	60.70±3.98 0.0076	64.04±3.71 3.3388	64.31±5.27 0.4329	63.91±4.94 0.4862
Diabetes	**77.47±3.84** 0.0369	73.96±3.02 0.0094	74.09±3.00 3.9478	74.09±2.73 0.3277	73.57±2.70 0.3470
German	**76.80±3.00** 0.0667	70.00±4.89 0.0088	69.90±5.12 4.0830	70.00±5.26 0.3618	69.20±4.80 0.2792
Haberman	**75.50±3.66** 0.0056	71.25±6.24 0.0065	71.59±6.59 0.4059	74.19±4.87 0.0660	74.19±4.87 0.0641
Heart	**85.19±3.31** 0.0064	84.44±3.81 0.0064	84.81±2.96 0.2680	84.44±3.81 0.0244	84.07±3.01 0.0358
Ionosphere	88.34±4.97 0.0120	86.91±4.23 0.0082	88.91±3.09 0.6296	**89.19±3.36** 0.0985	**89.19±3.36** 0.0936
Liverdisorder	**66.67±5.27** 0.0072	62.61±3.48 0.0072	63.77±3.66 0.5533	62.32±4.58 0.0628	62.03±4.62 0.0610
Sonar	77.46±7.20 0.0057	76.98±7.94 0.0073	77.46±8.09 0.1491	**77.47±8.49** 0.0171	76.93±5.51 0.0232
Vote	**94.71±0.59** 0.0087	93.33±1.87 0.0079	92.87±1.56 0.7652	94.48±0.46 0.1027	94.25±0.73 0.1260
Wdbc	97.36±1.11 0.0209	97.18±1.04 0.0070	98.25±1.11 1.5178	**98.59±0.90** 0.2436	**98.59±1.32** 0.2795
Wpbc	**81.85±2.67** 0.0066	59.67±4.02 0.0069	61.58±4.00 0.2479	61.09±3.58 0.0130	56.54±3.14 0.0317
W-T-L	–	12-0-0	9-1-2	9-0-3	10-0-2

train the model twice. Thirdly, LDM is even faster than SVM on some datasets. There are two reasons for this result. On the one hand, on some datasets, LDM needs less iterations than SVM to get the optimal value. On the other hand, for the regular-scale dataset, the matrix inversion will not consume too much time. Hence, LDM is even faster than SVM. This also reveals that the introduction of margin distribution can make the "quadprog function" easier to converge in the optimization process. Lastly, the results show that ODM is nearly ten times slower than LDM. The reason is that the introduction of square term makes the QPP of ODM with more inequality constraints. This result leads to that ODM needs to optimize a larger dual problem than LDM.

In terms of accuracy, we can see that LDM, ODM, LSLDM and WLLDM obtain higher accuracy than SVM on most datasets. The reason is that almost all datasets have hidden margin distribution. It can enhance the accuracy by optimizing the margin distribution of training samples. In addition, it can be observed that WLLDM is superior to LSLDM, ODM and LDM in accuracy. Cause is that the use of weighted linear loss is more reasonable. The training samples in the different positions are given different penalties, which can avoid over-fitting to a certain extent and achieve better learning performance. This also reveals that our WLLDM, whose solution is obtained by solving a linear system, performs comparable or better than LDM. In addition, from the last row of Tables 2 and 3, we can observe that our WLLDM gets the best classification accuracy in most cases. The result further indicates that WLLDM exceeds other four classifiers in classification accuracy.

For further fair and objective comparison of the five classifiers on the UCI datasets, the Friedman test [28] is resorted. First, the classification accuracy of five classifiers on all datasets is ranked, and the results are shown in Tables 4 and 5. The lower the rank is, the better the performance of the classifier is. Obviously, the average rank of WLLDM is the lowest among the five classifiers in both linear and nonlinear cases.

Next, let l represent the number of comparable algorithms and n be the number of datasets. Based on the average rank of five classifiers, the Friedman statistic parameter can be calculated as

$$\chi_F^2 = \frac{12n}{l(l+1)}\left[\sum_j \mathrm{Rank}_j^2 - \frac{l(l+1)^2}{4}\right] \tag{31}$$

where Rank_j denotes the average rank of the jth algorithm. And

Table 3 The results of five nonlinear classifiers on binary class UCI datasets

Dataset	WLLDM accuracy(%) Time(s)	LSLDM accuracy(%) Time(s)	ODM accuracy(%) Time(s)	LDM accuracy(%) Time(s)	SVM accuracy(%) Time(s)
Australian	87.10±1.84 0.0352	86.81±1.33 0.0330	86.96±1.59 2.2485	**87.39±1.32** 0.2210	86.96±1.34 0.2093
Blood	79.81±2.71 0.0408	79.81±2.41 0.0352	**79.95±2.11** 2.4497	78.61±2.75 0.3062	78.48±2.49 0.3186
Diabetes	**78.13±3.09** 0.0431	77.86±3.37 0.0389	77.60±3.13 3.0151	77.87±4.09 0.2421	77.47±3.70 0.4419
German	**77.70±2.29** 0.0885	76.80±2.06 0.0804	76.60±1.88 5.3144	76.10±2.27 0.5570	75.90±2.22 0.5012
Haberman	**76.15±1.87** 0.0082	74.84±1.53 0.0113	75.17±2.13 0.3503	75.17±1.153 0.0372	74.18±1.22 0.0505
Heart	**85.19±3.31** 0.0068	84.44±4.16 0.0096	84.44±4.16 0.3247	84.44±3.43 0.0999	84.07±3.81 0.0403
Ionosphere	**95.45±2.88** 0.0175	92.33±3.72 0.0166	91.47±3.47 0.5596	91.75±3.81 0.0634	91.47±3.43 0.0731
Liverdisorder	**73.62±4.44** 0.0092	72.17±3.60 0.0111	71.59±2.35 0.4803	71.59±4.64 0.0754	71.59±4.64 0.0570
Sonar	**87.04±3.51** 0.0092	**87.04±4.13** 0.0117	**87.04±3.20** 0.1921	86.56±3.83 0.0216	85.60±3.97 0.0252
Vote	**96.09±1.18** 0.0123	95.86±1.18 0.0151	**96.09±0.95** 0.8215	95.63±0.87 0.0823	95.40±0.73 0.0874
Wdbc	98.24±1.12 0.0402	98.06±1.41 0.0324	**98.59±1.64** 1.4944	98.41±1.97 0.2470	98.24±2.31 0.1875
Wpbc	**80.93±7.04** 0.0070	80.88±6.48 0.0095	80.90±5.52 0.1778	79.89±5.07 0.0218	78.85±4.03 0.0178
W-T-L	–	10-2-0	8-2-2	10-0-2	11-1-0

Table 4 Average rank of linear classifiers

Datasets	WLLDM	LSLDM	ODM	LDM	SVM
Australian	1	4	2	3	5
Blood	1	5	3	2	4
Diabetes	1	4	2.5	2.5	5
German	1	2.5	4	2.5	5
Haberman	1	5	4	2.5	2.5
Heart	1	3.5	2	3.5	5
Ionosphere	4	5	3	1.5	1.5
Liverdisorder	1	3	2	4	5
Sonar	2.5	4	2.5	1	5
Vote	1	4	5	2	3
Wdbc	4	5	3	1.5	1.5
Wpbc	1	4	2	3	5
Average rank	1.63	4.08	2.92	2.42	3.96

Table 5 Average rank of nonlinear classifiers

Datasets	WLLDM	LSLDM	ODM	LDM	SVM
Australian	2	5	3.5	1	3.5
Blood	2.5	2.5	1	4	5
Diabetes	1	3	4	2	5
German	1	2	3	4	5
Haberman	1	4	2.5	2.5	5
Heart	1	3	3	3	5
Ionosphere	1	2	4.5	3	4.5
Liverdisorder	1	2	4	4	4
Sonar	2	2	2	4	5
Vote	1.5	3	1.5	4	5
Wdbc	3.5	5	1	2	3.5
Wpbc	1	3	2	4	5
Average rank	1.54	3.04	2.67	3.13	4.63

$$F_F = \frac{(n-1)\chi_F^2}{n(l-1)-\chi_F^2} \tag{32}$$

is distributed according to F-distribution with $(l-1)$ and $(l-1)(n-1)$ degrees of freedom.

Table 6 records the Friedman test results on linear and nonlinear cases. As can be seen from Table 6, whether linear case or nonlinear case, F_F is much greater than the critical value. Therefore, we reject the null-hypothesis. The result means that there are significant differences between the five classifiers. Because the null-hypothesis is rejected, the Nemenyi post-hoc test [29] is used. The critical difference (CD) of the Friedman test with Nemenyi test is calculated by equation (33) as below. The results are listed in Table 6.

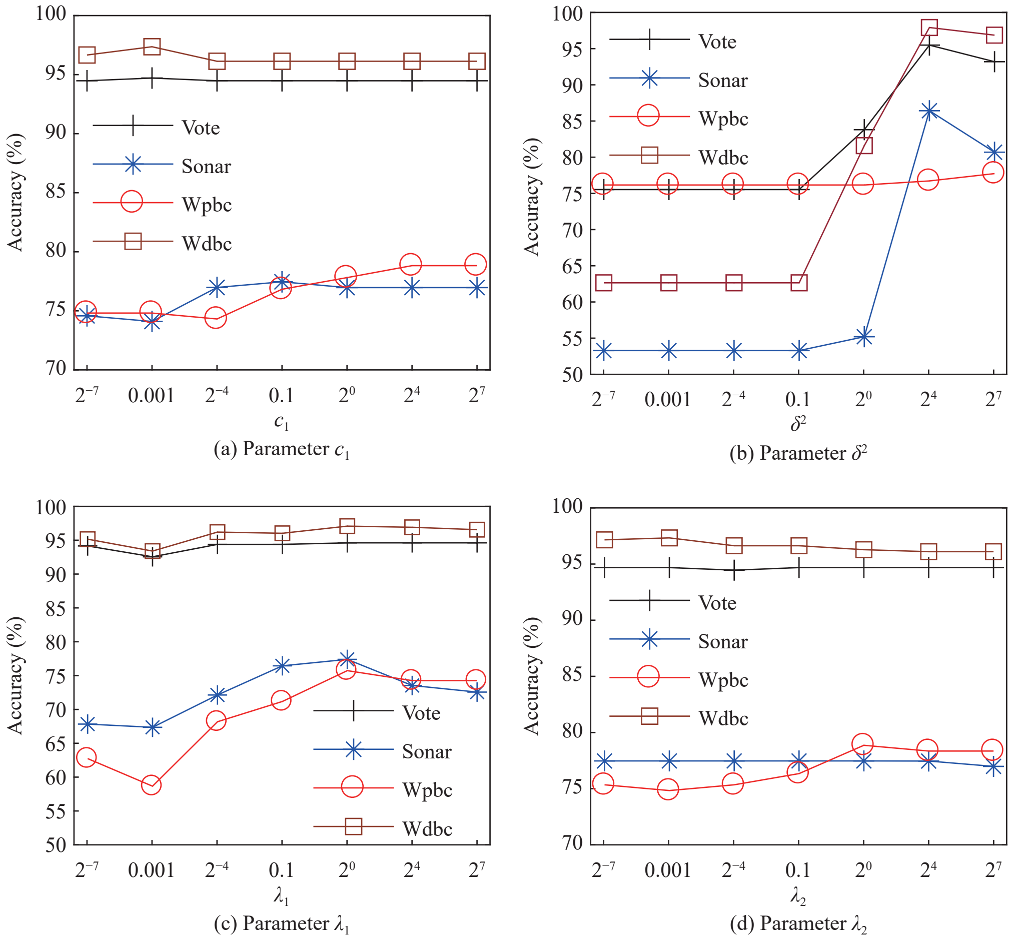
Table 6 The results in the Friedman test on UCI datasets

Case	Significant level	χ_F^2	F_F	Critical value	q_α	CD
Linear	$\alpha = 0.1$	20.96	8.53	2.584	2.728	1.761
	$\alpha = 0.05$	20.96	8.53	2.077	2.459	1.587
Nonlinear	$\alpha = 0.05$	23.88	10.89	2.584	2.728	1.761
	$\alpha = 0.1$	23.88	10.89	2.077	2.459	1.587

$$\text{CD} = q_\alpha \sqrt{\frac{l(l+1)}{6 \times n}} \tag{33}$$

If the difference between the ranks of two algorithms is larger than CD, their performance is considered to be significantly different. For the linear case, when $\alpha = 0.05$ and $\alpha = 0.1$, the performance of WLLDM is always significantly different from LSLDM and SVM algorithms. As for the nonlinear case, obviously, when $\alpha = 0.05$, only the difference between the ranks of WLLDM and SVM is larger than CD. When $\alpha = 0.1$, WLLDM performs significantly better than LDM and SVM.

Subsequently, in order to study the influence of penalty parameters $c_1, \lambda_1, \lambda_2$ and kernel parameter δ on the performance of WLLDM, the comparative experiments are carried out on some datasets by fixing other parameters. These datasets are Wpbc, Sonar, Wdbc and Vote. Figure 2 plots the accuracy of WLLDM with different parameters, where the x-axis represents the range of parameters, and the y-axis represents the classification accuracy. As can be seen from Figure 2, the values of $c_1, \lambda_1, \lambda_2$ and δ impact on the accuracy heavily for most cases, indicating that it is crucial to select the suitable parameters in the experiments.



(a) Parameter c_1

(b) Parameter δ^2

(c) Parameter λ_1

(d) Parameter λ_2

Figure 2 Parameters influence on some UCI datasets.

2) Multiple class classification

In order to further verify the generalization performance of WLLDM, ten multiple class UCI datasets are selected for experiments. In the experiments, only the nonlinear classifiers are compared. The testing results are listed in Table 7, where the best accuracy is highlighted

Table 7 The results of five nonlinear classifiers on multiple class UCI datasets

Dataset	WLLDM accuracy(%) Time(s)	LSLDM accuracy(%) Time(s)	ODM accuracy(%) Time(s)	LDM accuracy(%) Time(s)	SVM accuracy(%) Time(s)
Air	97.77±1.11 0.0383	**98.32±1.05** 0.0380	97.22±0.87 0.8726	97.77±0.69 0.0900	97.77±0.69 0.0898
Balance	**91.52±0.78** 0.0546	91.36±0.88 0.0485	91.05±1.11 3.3217	91.05±1.11 0.4839	90.89±1.33 0.4467
Glass	74.77±4.47 0.0268	74.76±5.40 0.0366	**75.69±4.83** 0.3936	72.43±4.76 0.0827	71.03±3.13 0.0792
Iris	**98.67±1.63** 0.0081	98.00±1.63 0.0160	97.33±1.33 0.1797	96.67±2.11 0.0293	96.67±2.11 0.0319
Libras_Movement	86.67±1.40 0.1717	**86.95±2.24** 0.2064	86.67±2.71 3.0851	86.39±2.38 0.4337	86.39±2.38 0.4241
Soybean	**100±0** 0.0094	**100±0** 0.0204	**100±0** 0.0828	**100±0** 0.0317	**100±0** 0.0299
Vehicle	81.80±2.37 0.1119	**83.33±2.49** 0.1028	77.06±2.47 5.8933	76.59±2.27 0.6113	76.23±2.35 0.6366
Vowel	97.73±0.96 0.1100	**98.48±0.76** 0.1278	97.73±1.14 5.2982	97.35±2.12 0.4980	96.96±2.79 0.4679
Wine	**98.89±2.22** 0.0119	97.75±2.76 0.0160	96.62±2.11 0.2273	96.06±1.39 0.0425	95.51±1.37 0.0365
Zoo	**97.14±5.71** 0.0237	97.04±3.86 0.0405	96.11±4.84 0.1948	**97.14±5.71** 0.0580	**97.14±5.71** 0.0587
Average rank	**1.80**	1.95	2.75	3.45	4.05
W-T-L	–	5-1-4	6-1-3	7-3-0	8-2-0

in boldface.

From Table 7, some conclusions can be obtained. Firstly, WLLDM has the best accuracy in 5 out of 10 datasets. On the rest datasets, WLLDM is the second best with comparable performance. Secondly, it can be observed that ODM, LSLDM and LDM can perform better than SVM, indicating that the margin distribution is important for generalization performance of the model. Thirdly, in terms of execution time, WLLDM and LSLDM cost less time than LDM and ODM. The conclusion is the same as that of binary classification. Last but not least, according to the W-T-L summarization, it can be observed that the accuracy of WLLDM is better than that of LSLDM, LDM, ODM and SVM on most datasets.

2. NDC dataset

In order to further verify the computational complexity of WLLDM, we perform experiments on the NDC dataset [24]. In the experiments, we compare WLLDM with WLTSVM, WLSVM, LSSVM, LDM and SVM. Due to the limitation of memory and running time, the maximum size of NDC dataset is limited to 10K in this experiments. Table 8 shows the attributes of the NDC dataset. Each dataset is randomly divided into a training set and a testing set.

In the experiments, only the nonlinear algorithms are compared. The experimental result is recorded in Table 9, including the classification accuracy and running time. As can be seen from Table 9, our WLLDM, WLTSVM, WLSVM and LSSVM are faster than LDM and SVM. Cause is that they only need to solve simple linear equations rather than a QPP in LDM and SVM. Meanwhile, the learning speed of LSSVM is the fastest among the all algorithms. The reason is that WLLDM, WLTSVM and WLSVM need spend more time on training model twice. In addition, it is not surprising that WLLDM is slightly slower than WLTSVM and WLSVM, due to the introduction of margin distribution information. But the accuracy of our WLLDM on most datasets exceeds that of WLTSVM and WLSVM.

Table 8 The attributes of NDC dataset

Dataset	Training samples	Testing samples	Feature
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

3. Steel surface defect dataset

In this sub-section, we discuss the application of the proposed WLLDM to the steel surface defects classification problem. In the experiments, six types of defect images are chosen from the Northeastern University (NEU) surface defect database [25], [26], including the inclusion (IN), scratches (SH), patches (PH), crazing (CG), pitted-

Table 9 The experimental results of nonlinear classifiers on NDC dataset

Dataset	WLLDM accuracy(%) Time(s)	WLTSVM accuracy(%) Time(s)	WLSVM accuracy(%) Time(s)	LSSVM accuracy(%) Time(s)	LDM accuracy(%) Time(s)	SVM accuracy(%) Time(s)
NDC-500	**96.00** 0.0398	82.00 0.0370	94.00 0.0470	92.00 0.0368	93.00 0.1387	92.00 0.0906
NDC-1K	**99.00** 0.1481	95.50 0.1228	98.50 0.1531	96.50 0.0818	97.50 0.5775	97.00 0.5106
NDC-2K	**100** 0.7219	96.75 0.5597	99.00 0.6373	99.50 0.3221	99.75 2.9733	99.75 2.5047
NDC-3K	**98.67** 1.7813	97.00 1.3611	97.33 1.4420	98.17 0.7329	98.17 5.4340	98.17 4.2322
NDC-4K	99.25 3.9095	96.75 2.6763	98.75 2.8360	99.25 1.5911	**99.38** 11.5682	**99.38** 8.6171
NDC-5K	**99.00** 9.2003	97.40 4.4961	98.60 7.6191	98.60 5.1860	98.80 23.6666	98.80 18.1502
NDC-10K	**99.45** 50.1920	98.70 31.0330	99.35 41.4642	99.40 24.9432	99.40 136.9389	99.40 100.4095

surface (PE) and rolled-in-scale (RE). The typical images are shown in Figure 3. Before the experiments, the defect images are preprocessed [30]–[32] and each defect is described with a feature vector with 35 dimensions. Finally, these feature vectors make up the defect dataset. In the experiments, 2974 samples are regarded as training samples and the other samples are testing samples.

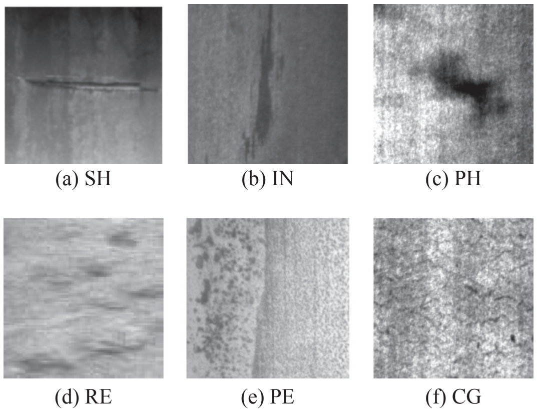


Figure 3 The typical defect images.

In general, the dataset collected from real-world scenarios cannot be separable linearly. Thus, the nonlinear classifiers are compared in the experiments. In order to realize the multi-class classification, the binary tree method is combined with the SVM-type classifiers. The accuracy and the running time of five nonlinear classifiers are shown in Figure 4.

From Figure 4, we observe that WLLDM is more accurate than the other four models. One reason is that the margin distribution of defect samples is considered in WLLDM. Compared with margin theory, the margin distribution is more important for the generalization performance of the model. Another reason is that WLLDM gives different penalties to training samples, which enhances the robustness of the model. In addition, the execution time of WLLDM, LSLDM and LSSVM is relatively short in comparison to LDM and SVM. These observations help us conclude that WLLDM is an important reference value for real-life applications.

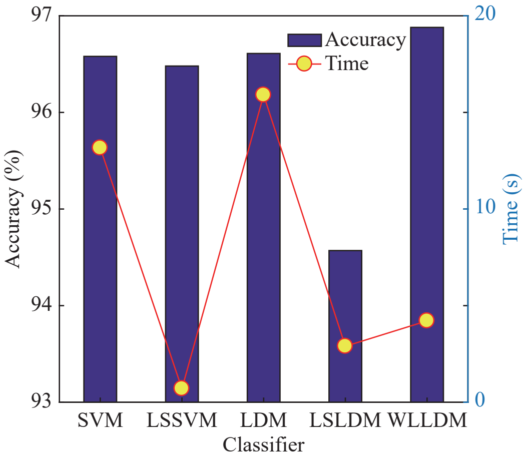


Figure 4 The results of five classifiers on defect dataset.

VI. Conclusions

In this paper, we mainly focus on reducing the computational complexity of LDM under the premise of high accuracy. Based on the studies of loss function and LDM, we propose a weighted linear loss LDM (WLLDM) classifier. On the one hand, WLLDM inherits the all the advantages of LDM. On the other hand, the weighted linear loss is adopted in WLLDM, which can avoid overfitting to a certain extent and yield better generalization performance than linear loss. In addition, WLLDM solves a simple linear equation instead of solving a single QPP, which can speed up the training process and is suitable for large-scale classification. Lastly, the perfect generalization performance and greater efficiency of WLLDM are proved by a series of experiments on different datasets.

There are three penalty parameters and a kernel parameter in WLLDM model, these parameters will affect the classification accuracy of the model. Thus, the future work mainly focuses on how to obtain the optimal parameters more efficiently to improve the classification performance of the model.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 71771112), the Natural Science Foundation of Liaoning Province (Grant No. 2022-MS-353), and the Basic Scientific Research Project of Education Department of Liaoning Province (Grant No. LJKMZ20220640).

References

[1] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999.

[2] A. Dixit, A. Mani, and R. Bansal, "CoV2-Detect-Net: Design of COVID-19 prediction model based on hybrid DE-PSO with SVM using chest X-ray images," *Information Sciences*, vol. 571, pp. 676–692, 2021.

[3] P. Negri, S. Cumani, and A. Bottino, "Tackling age-invariant face recognition with non-linear PLDA and pairwise SVM," *IEEE Access*, vol. 9, pp. 40649–40664, 2021.

[4] M. B. Abidine and B. Fergani, "Activity recognition from smartphone data using weighted learning methods," *Intelligenza Artificiale*, vol. 15, no. 1, pp. 1–15, 2021.

[5] S. Mehrkanoon, X. L. Huang, and J. A. K. Suykens, "Non-parallel support vector classifiers with different loss functions," *Neurocomputing*, vol. 143, pp. 294–301, 2014.

[6] C. F. Lin and S. D. Wang, "Fuzzy support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 464–471, 2002.

[7] L. Tang, Y. J. Tian, and P. M. Pardalos, "A novel perspective on multiclass classification: Regular simplex support vector machine," *Information Sciences*, vol. 480, pp. 324–338, 2019.

[8] X. L. Yang, Q. Song, and Y. Wang, "A weighted support vector machine for data classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 21, no. 5, pp. 961–976, 2007.

[9] L. Reyzin and R. E. Schapire, "How boosting the margin can also boost classifier complexity," in *Proceedings of the 23rd International Conference on Machine Learning*, Pittsburgh, PA, USA, pp. 753–760, 2006.

[10] W. Gao and Z. H. Zhou, "On the doubt about margin explanation of boosting," *Artificial Intelligence*, vol. 203, pp. 1–18, 2013.

[11] Z. H. Zhou, "Large margin distribution learning," in *Proceedings of the 6th IAPR Workshop on Artificial Neural Networks in Pattern Recognition*, Montreal, Canada, pp. 1–11, 2014.

[12] T. Zhang and Z. H. Zhou, "Large margin distribution machine," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, pp. 313–322, 2014.

[13] L. M. Liu, M. X. Chu, R. F. Gong, *et al.*, "Nonparallel support vector machine with large margin distribution for pattern classification," *Pattern Recognition*, vol. 106, pp. 107374, 2020.

[14] F. Y. Cheng, J. Zhang, Z. Y. Li, *et al.*, "Double distribution support vector machine," *Pattern Recognition Letters*, vol. 88, pp. 20–25, 2017.

[15] U. Gupta and D. Gupta, "Least squares large margin distribution machine for regression," *Applied Intelligence*, vol. 51, no. 10, pp. 7058–7093, 2021.

[16] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Processing Letters*, vol. 9, no. 3, pp. 293–300, 1999.

[17] Jayadeva, R. Khemchandani, and S. Chandra, "Twin support vector machines for pattern classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 5, pp. 905–910, 2007.

[18] Y. H. Shao, Z. Wang, Z. M. Yang, *et al.*, "Weighted linear loss support vector machine for large scale problems," *Procedia Computer Science*, vol. 31, pp. 639–647, 2014.

[19] Y. H. Shao, W. J. Chen, Z. Wang, *et al.*, "Weighted linear loss twin support vector machine for large-scale classification," *Knowledge-Based Systems*, vol. 73, pp. 276–288, 2015.

[20] S. Abe, "Unconstrained large margin distribution machines," *Pattern Recognition Letters*, vol. 98, pp. 96–102, 2017.

[21] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2002.

[22] T. Zhang and Z. H. Zhou, "Optimal margin distribution machine," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1143–1156, 2020.

[23] D. Dua and C. Graff, "UCI machine learning repository," Available at: http://archive.ics.uci.edu/ml.

[24] D. R. Musicant, "NDC: Normally distributed clustered datasets," Available at: http://www.cs.wisc.edu/dmi/svm/ndc.

[25] Y. Q. Bao, K. C. Song, J. Liu, *et al.*, "Triplet-graph reasoning network for few-shot metal generic surface defect segmentation," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 5011111, 2021.

[26] Y. He, K. C. Song, Q. G. Meng, *et al.*, "An end-to-end steel surface defect detection approach via fusing multiple hierarchical features," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493–1504, 2020.

[27] B. Fei and J. B. Liu, "Binary tree of SVM: A new fast multiclass training and classification algorithm," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 696–704, 2006.

[28] J. Liu and Y. B. Xu, "T-friedman test: A new statistical test for multiple comparison with an adjustable conservativeness measure," *International Journal of Computational Intelligence Systems*, vol. 15, no. 1, pp. 29, 2022.

[29] A. Benavoli, G. Corani, and F. Mangili, "Should we really use post-hoc tests based on mean-ranks?," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 152–161, 2016.

[30] C. H. Lin, J. S. Tsai, and C. T. Chiu, "Switching bilateral filter with a texture/noise detector for universal noise removal," *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2307–2320, 2010.

[31] Z. Y. He and L. N. Sun, "Surface defect detection method for glass substrate using improved Otsu segmentation," *Applied Optics*, vol. 54, no. 33, pp. 9823–9830, 2015.

[32] M. X. Chu, R. F. Gong, S. Gao, *et al.*, "Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 140–150, 2017.

Ling LIU was born in 1998. She received the B.S. degree in measurement and control technology and instruments from University of Science and Technology Liaoning, China, in 2021. She is currently working towards the M.S. degree at University of Science and Technology Liaoning, China. Her current research interests include pattern recognition and machine learning.

(Email: ll15566271785@163.com)

Maoxiang CHU was born in 1978. He is with the School of Electronic and Information Engineering in University of Science and Technology Liaoning. He received the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University in 2015. His current research interests include pattern recognition, machine learning, image processing, especially the pattern clasification.
(Email: chu522004@163.com)

Rongfen GONG was born in 1979. She is with the School of Electronic and Information Engineering in University of Science and Technology Liaoning, Anshan, China. She received the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, Shenyang, China, in 2020. Her current research interests include pattern recognition and machine learning.
(Email: fx_gong@hotmail.com)

Liming LIU was born in 1994. She received the M.S. degree in control science and engineering from University of Science and Technology Liaoning in 2019. She is a Ph.D. candidate at School of Electronic and Information Engineering in University of Science and Technology Liaoning. Her current research interests include pattern recognition and machine learning.
(Email: llm06101021@hotmail.com)

Yonghui YANG was born in 1971. He received the Ph.D. degree from University of Science and Technology Liaoning in 2018. Now he is an Associate Professor and Doctoral Supervisor at School of Electronic and Information Engineering in University of Science and Technology Liaoning. His research interests include intelligent control, pattern recognition and machine learning.
(Email: yangyh2636688@163.com)