IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 30, 2019, accepted May 20, 2019, date of publication May 27, 2019, date of current version June 10, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2919185

LINEX Support Vector Machine for Large-Scale Classification

YUE MA[1,2,3], QIN ZHANG[4], DEWEI LI[1,2,3], AND YINGJIE TIAN[2,3,5]

[1]School of Mathematical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China
[2]Research Center on Fictitions Economy & Data Science, Chinese Academy of Sciences, Beijing 100190, China
[3]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing 100190, China
[4]Data Center for Cloud and Smart Industries Group, Tencent, Shenzhen 518000, China
[5]School of Economics and Management, University of Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Yingjie Tian (tyj@ucas.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71731009, Grant 61472390, Grant 71331005, and Grant 91546201, and in part by the Social Science Research Common Program of Beijing Municipal Commission of Education under Grant SM201710037002.

ABSTRACT Traditional soft margin support vector machine usually uses hinge loss to build a classifier with the ‘‘maximum-margin’’ principle. However, C-SVM depends on support vectors causing the loss of data information. Then, least square support vector machine is proposed with square loss (l_2-loss). It establishes equality constraints instead of inequalities and considering all the instances. However, the square loss is still not the perfect one, since it gives equivalent punishment to the instances at both sides of the center plane. It does not match the reality considering the instances between two center planes deserve heavier penalty than the others. To this end, we propose a novel SVM method with the adoption of the asymmetry LINEX (linear-exponential) loss, which we called it LINEX-SVM. The LINEX loss gives different treatments to instances based on the importance of each point. It gives a heavier penalty to the points between two center planes while drawing light penalty to the points outside of the corresponding center planes. The comprehensive experiments have been implemented to validate the effectiveness of the LINEX-SVM.

INDEX TERMS LINEX loss, large-scale classification, support vector machine (SVM).

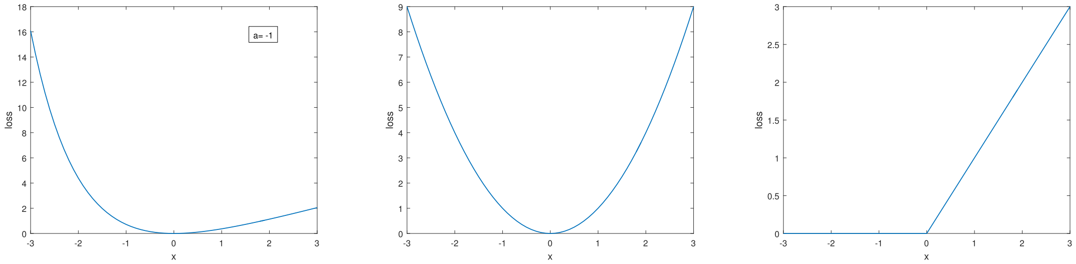
I. INTRODUCTION

Support vector machine, which was introduced in the early 1990s [1], rooted in statistical learning theory (SLT) [2]. It is a powerful tool for classification and regression, widely used in amounts of fields, such as financial forecasting [3], computational biology [4], image annotation [5] and text mining [6]. The fundamental idea is finding a hyperplane to separate the data with the maximization of the distance between the two support planes, which is known as 'max-margin' principle. Usually, the margin maximization is achieved by solving an optimization problem with inequality condition. To avoid over-fitting, the original SVM was extended to the soft-margin SVM (i.e., C-SVM), by bringing in slack variables to relax the constraints and increase a penalty term for the slack variables in the objective function [7]. The loss adopted by C-SVM typically is the hinge loss [8]. In this way, C-SVM depends only on part of the training data, i.e., the support samples, which makes it dramatically sensitive to noise.

Later, the C-SVM was extended for solving function estimation problems, for example, a support vector interpretation of ridge regression [9], which uses equivalent constraints instead of inequalities in C-SVM. In 1999, Suykens considered equality constraints for classification with a formulation in the least squares sense and proposed least squares SVM (LSSVM) [10]. Different from C-SVM in which non-support vectors are not utilized to optimize the classifiers, in LSSVM, the information of all the data points are fully used. For a binary classification problem, LSSVM seeks for two parallel center planes. Each plane is trained to locate at the center of the points in the same class. And the l_2 loss used in LSSVM penalties the points on both sides symmetrically. However, it is more reasonable to give heavier penalties to the points between the planes since these points are normally the incorrectly-classified samples. To this end, we propose a new SVM model that uses asymmetric linear-exponential loss (LINEX) to achieve this goal.

To be intuitive, we illustrate and compare LINEX loss with hinge loss and l_2 loss in Fig.1, where Fig. (a) demonstrates the LINEX loss with the hyper-parameter $a = -1$ and

The associate editor coordinating the review of this manuscript and approving it for publication was Somayeh Sojoudi.



(a) (b) (c)

FIGURE 1. The illustration of LINEX loss (a), squared loss (b), and hinge loss (c). LINEX loss is with severely asymmetric smooth convex function, while hinge loss is with non-smooth convex function and squared loss is with symmetric smooth convex function.

Fig. (b), (c) shows the squared loss and hinge loss, respectively. We can see the left part of LINEX loss is similar to the curve of $y = x^2$ (i.e. the squared loss) while the right part is similar to the straight line $y = kx$ (i.e. the right part of the hinge loss). This characteristic makes LINEX loss could gain the advantage of squared loss and be more suitable to train the model with appreciate sample penalties. Furthermore, LINEX loss is superior to hinge loss since hinge loss is non-smooth.

Further, we introduce LINEX loss to SVM, i.e., LINEX-SVM, for classification problems. This combination brings several advantages: (1) every instance contributes, which will not cause loss of information at any point; (2) it gives heavier penalties to the points near the classification boundary which are with higher probabilities of incorrectly-classifying. This will improve classification performance; (3) it has good generality since LSSVM is a special situation of it. When the small parameter a takes a very small value, LINEX-SVM is approximately proportional to LSSVM. (4) it can be transformed into an unconstrained convex optimization problem, which is easy to solve. Then Nesterov accelerated gradient (NAG) algorithm can be employed to solve it effectively for large-scale data. The numerical experiments illustrate the good performance of the LINEX-SVM model.

The organization of the rest content is as follows. The background, mainly in terms of C-SVM and LSSVM, is introduced in Section II. Section III proposes the new LINEX-SVM model and the related theory analysis. In section IV, various experiment evaluation results are displayed. Finally, Section V ends the paper with conclusions.

II. BACKGROUND

Here, we briefly introduce the background and some related works about LINEX-SVM. To be specific, we introduce the preliminary knowledge about C-SVM, LSSVM and LINEX loss.

A. C-SVM WITH HINGE LOSS

For binary classification problems, the training set is represented as

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)\} \tag{1}$$

where $x_i \in R^n, y_i \in \{-1, 1\}, i = 1, \cdots, l$. The formulation of C-SVM is a constrained convex quadratic programming problem (QPP) show in Problem (2).

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

where $\xi = (\xi_1, \cdots, \xi_l)^{\mathrm{T}}$ is a relaxation variable. $C > 0$ is a penalty parameter to balance the precision and model complexity.

Many loss functions have been presented in SVM to improve its generalization ability. In this framework, hinge loss [2] is utilized. The formulation of hinge loss is defined as

$$L_{hinge}(x) = max\{0, x\}, \quad \forall x \in R \tag{3}$$

And the loss is illustrated in in Fig.1(a). In a more general form of expression, Problem (2) can be rewritten as

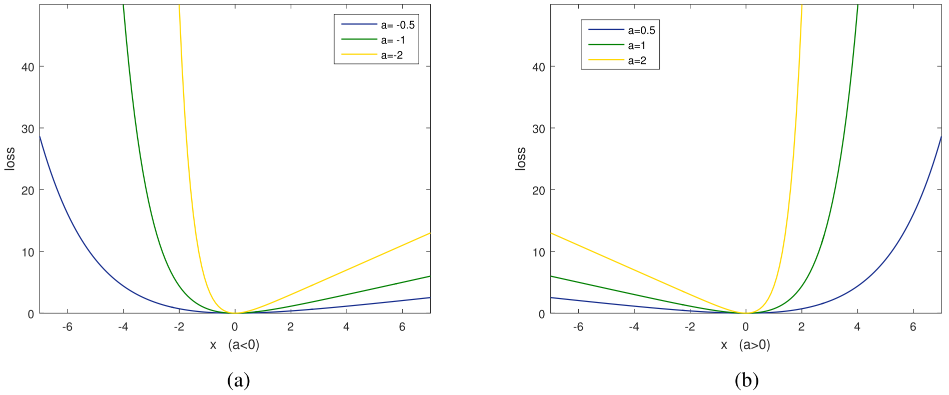
$$\begin{aligned} \xi_i &= max(0, 1 - y_i(w^\top x_i + b)) \\ &= L_{hinge}(1 - y_i(w^\top x_i + b)). \end{aligned} \tag{4}$$

Then the QPP can be rewritten as Eq. (5) by eliminating the constraints.

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} L_{hinge}(1 - y_i(w^\top x_i + b)) \tag{5}$$

Then many works arise around the loss function. Vapnik's ε-insensitive loss and Huber's loss have been employed to enhance the robustness of SVM [11], [12]. Twin support vector machine(TWSVM) is proposed using two symmetrical square loss [13]. Different from the Vapnik's SVM, which devotes to maximize the margin, SVM with pinball loss [14] maximize the quantile distance, leading to noise insensitivity. Moreover, the ε-insensitive loss and the ramp loss are applied to derive a robust nonparallel support vector machine (NPSVM) [15], [16]. The general form for SVM with different loss function is

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} L(\xi_i) \tag{6}$$



(a)

(b)

FIGURE 2. The illustration of LINEX loss function with different parameter *a*. The subfigures (a) and (b) display that the parameter *a* controls the direction of the loss function curve and the value |*a*| decides the steepness of the curve. LINEX loss is with a seriously asymmetric smooth convex function, while hinge loss is with a non-smooth convex function and squared loss is with a symmetric smooth convex function.

where $L(\xi_i) = \xi_i(w; x_i, y_i)$ is a loss function and $C > 0$ is a parameter representing the weight of loss. For the mentioned loss functions, the bounds of classification error and their more theoretic properties can be found in [17], [18].

In C-SVM, the loss only penalizes the support vectors, the non-support vectors contribute nothing to the classifier. However, in many real-world applications, where the size of datasets are small, the classification results are easily affected by the outliers.

B. LSSVM WITH SQUARED LOSS

LSSVM [10] uses two parallel hyperplanes to assist in the optimization of the classifier. The formula of LSSVM's problem is the convex QPP with equality constraints, shown in Problem (7).

$$\begin{aligned} &\min_{w,b,\eta} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\xi_i^2 \\ &s.t.\ y_i((w\cdot x_i)+b) = 1-\xi_i, \quad i = 1,\cdots,l \end{aligned} \tag{7}$$

The first term of the objective function, $\frac{1}{2}\|w\|^2$, represents the margin between two hyperplanes $(w \cdot x) + b = 1$ and $(w \cdot x) + b = -1$. The minimization of the second term, $\min \sum_{i=1}^{l} \xi_i^2$, make the two hyperplanes to locate at the center of the corresponding class of points. Different from C-SVM which concentrates only on support vectors, all the instances contribute to the classification hyperplane in LSSVM. Besides, Problem (7) is a simple convex QPP with equality constraints. Solving Problem (7) is equivalent to solving a system of linear equations. Therefore, LSSVM works faster than SVM. However, LSSVM gives the same penalty to the points around the classification boundary and ignores the different contribution of points with different distances. The points at the intersection should receive more attention than the points are far away.

C. LINEX LOSS

To improve the ability of SVM, we introduce the LINEX loss, which was first mentioned in statistics [19]. In the beginning, symmetric quadratic loss, such as the mean square errors (MSE), has widely been used for measuring bias. However, the symmetric loss has limitations in many circumstances, particularly where overestimation and underestimation matter. Then, asymmetric loss, such as LINEX loss, is introduced and explored [20].

Specifically, the function of LINEX loss is defined as:

$$L(x) = \exp(ax) - ax - 1 \tag{8}$$

where $a \neq 0$ is a parameter, determining the steepness of LINEX function. Fig.2 shows some examples of LINEX loss in terms of different values of a. We can see, The sign of a controls the direction of the curve: when $a < 0$, the left of the function is steeper than the right part, which means the negative points would receive heavier penalties than the positive ones even though they have the same modulus. The loss $L(x)$ increases approximately exponentially as $x \to -\infty$, while it increases approximately linearly when $x \to +\infty$. But when $a > 0$, the situation is opposite.

Further, the value of $|a|$ controls the steepness of the curve. The larger value of $|a|$, the steeper the curve is, illustrated in Fig.2. When $|a|$ is very small, $L(x) \approx a^2x^2/2$, is proportional to the l_2 loss. So when parameter a chooses the appropriate value, LINEX loss can degenerate to l_2 loss. The LINEX loss function was adopted in various realistic scenarios: estate price prediction [21], estimation for population [22], [23]. Besides, LINEX loss was also applied for the ridge regression estimator in Statistic [24].

Due to the merits of LINEX loss, we introduce it to SVM and propose a new SVM problem: LINEX-SVM, which will be shown later. As far as we know, this is the first attempt that LINEX loss is applied for classification.

III. METHODOLOGY

In this section, we discuss the details of the LINEX-SVM model. We first introduce its primal problem, then its dual problem. Further, we will give some theoretical analysis and show the algorithm in the next sections.

A. PRIMAL PROBLEM

Considering the binary classification problem, the training set is

$$T = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)\} \tag{9}$$

where $x_i \in R^n, y_i \in \{-1, 1\}, i = 1, \cdots, l$. In order to solve the classification problem, we seek two parallel hyperplanes

$$\begin{aligned} (w \cdot x) + b &= 1 \\ (w \cdot x) + b &= -1 \end{aligned} \tag{10}$$

by solving the following problem

$$\begin{aligned} \min_{w,b,\xi} \ & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(exp(a\xi_i) - a\xi_i - 1) \\ s.t. \ & y_i((w \cdot x_i) + b) = 1 + \xi_i, \quad i = 1, \cdots, l \end{aligned} \tag{11}$$

where $C > 0$ is the weight parameter of penalty error, a is the parameter in LINEX loss function, and $\xi = (\xi_1, \cdots, \xi_l)$ is a slack variables. The solution w^*, b^* to the optimization problem determines the classification hyperplane $(w^* \cdot x) + b^* = 0$.

It is a convex optimization problem with equality constraints. Three criteria are considered here. Firstly, we hope the positive hyperplane $(w \cdot x) + b = 1$ located at the center of positive instances. And the negative hyperplane $(w \cdot x) + b = -1$ located at the center of negative instances. Secondly, we maximize the margin between the above two hyperplanes, which is measured by $\frac{2}{\|w\|}$. Third, if we set $a > 0$, which means we hope the points located between two center planes would be punished heavier since they are more likely to cause mis-classification. Thus, we use LINEX loss to measure the errors $\xi_i, i = 1, \cdots, l$. Based on these three considerations, the primal optimization problem is established.

B. DUAL PROBLEM

Follow the methods of classical SVM, we show the dual problem of LINEX-SVM in this section and introduce the kernel function to it. To get the dual problem, we first derive the Lagrange function of problem (11) as

$$\begin{aligned} L(w, b, \xi_i, \alpha_i) = & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(exp(a\xi_i) - a\xi_i - 1) \\ & - \sum_{i=1}^{l}\alpha_i(y_i((w \cdot \Phi(x_i)) + b) - \xi_i - 1) \end{aligned} \tag{12}$$

where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_l)^\top$ is the Lagrange multiplier and $\Phi(x_i)$ is a function that maps the instance x_i into a higher dimensional space. Solving the Karush-Kuhn-Tucker(KKT) system, we can get the dual problem as follows

$$\begin{aligned} \min_{\alpha_i} \ & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \Phi(x_i)^\top \Phi(x_j) - (1 - \frac{1}{a})\sum_{i=1}^{l}\alpha_i \\ & + \sum_{i=1}^{l}(C - \frac{\alpha_i}{a})ln(\frac{aC - \alpha_i}{aC}) \\ s.t. \ & \sum_{i=1}^{l}\alpha_i y_i = 0, \quad i, j = 1, \cdots, l \end{aligned} \tag{13}$$

The vector form of the dual problem is

$$\begin{aligned} \min_{\alpha_i} \ & \frac{1}{2}\alpha^\top Q\alpha - (1 - \frac{1}{a})e^\top\alpha - (Ce - \frac{1}{a}\alpha)^\top\beta \\ s.t. \ & \alpha^\top y = 0 \end{aligned} \tag{14}$$

where $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_l)^\top, \beta = (\beta_1, \beta_2, \cdots, \beta_l)^\top = (ln(\frac{aC-\alpha_1}{aC}), ln(\frac{aC-\alpha_2}{aC}), \cdots, ln(\frac{aC-\alpha_l}{aC}))^\top$. The dual problem maintains to be a convex problem.

Some papers solve the SVM model through the dual problem while some others try to solve the primal one directly. In many real-world situations, such as document applications, where the data are in high dimension, the dual problem is popular. Because the variable α_i in dual problem is related to the number of samples rather than the dimension. Dual problems with kernel trick bring convenience to SVM in solving nonlinear classification problems. With the kernel function $K(x_i, x_j) = \Phi(x_i)^\top\Phi(x_j)$, SVM can be extended for nonlinear case easily without knowing the exact function $\Phi(x)$. On the other side, in many other scenarios, linear SVM is more popular since its high efficiency, especially in some large-scale scenarios. The reason is that the weight w in primal problem is related to the dimension of instances while the α in dual problem is related to the number of instances.

IV. THEORETICAL ANALYSIS

In this section, we would like to do some theoretical analysis on LINEX-SVM introduced in the last section. We start with Bayes rule, then go to the error bound analysis.

A. BAYES RULE

We first show how LINEX loss achieves Bayes rule. Assuming that samples $\{(x_i, y_i)\}_{i=1}^{l}$ are extracted independently from the same probability ρ, the probability ρ is defined on $X \times Y$, where $X \subseteq \mathbb{R}^n$ represents the feature space and $Y = \{-1, 1\}$ is the label space. Further, we assume condition distribution $\rho(y|x)$ is a binomial distribution, given by $P(y = -1|x)$ and $P(y = 1|x)$. The ultimate goal of the classification problem is to get a classifier $\mathcal{C} : X \to Y$ with less error. Define the Bayes classifier as:

$$f_c(x) = \begin{cases} 1, & if \ P(y = 1|x) \geq P(y = -1|x); \\ -1, & if \ P(y = 1|x) < P(y = -1|x); \end{cases} \tag{15}$$

With any loss function L, the expected risk of a classifier $f : X \to \mathbb{R}$ is defined as:

$$\mathcal{R}_{L,\rho}(f) = \int_{X\times Y} L(1 - yf(x))d\rho \tag{16}$$

By minimizing the expected risk on all measurable classification functions, we can get function $f_{L,\rho}$ as:

$$f_{L,\rho}(x) = arg\ min_{s\in\mathbb{R}} \int_{Y} L(1 - y(x)s)d\rho(y|x), \quad \forall x \in X \tag{17}$$

Then we can obtain Theorem 1 demonstrating that Bayes rule holds for LINEX loss L_{linex}. And we show the proof in details further.

Theorem 1: Function $f_{L_{linex},\rho}$, which minimizes the expected $L_{linex} - risk$ over all measurable function $f : X \to Y$, is equal to the Bayes classifier, i.e., $f_{L_{linex},\rho}(x) = f_c(x), \forall x \in X$.

Proof: Simple calculation shows that

$$\begin{aligned}&\int_Y L_{linex}(1 - y(x)s)d\rho(y|x)\\ &\quad = L_{linex}(1-s)P(y=1|x) + L_{linex}(1+s)P(y=-1|x)\\ &\quad = \{exp[a(1-s)] - a(1-s) - 1\}P(y=1|x)\\ &\qquad + \{exp[a(1+s)] - a(1+s) - 1\}P(y=-1|x)\end{aligned}$$

Hence, when $P(y = 1|x) > P(y = -1|x)$, the minimal value is obtained at $s = -1$. when $P(y = 1|x) < P(y = -1|x)$, the minimal value is obtained at $s = 1$. when $P(y = 1|x) = P(y = -1|x)$, the minimal value is obtained at $s = -1 (a > 0)$ or $s = 1 (a < 0)$. Therefore, $f_{L_{linex},\rho}(x)$, which minimizes the expected risk measured by the LINEX loss, satisfies

$$f_{L_{linex},\rho}(x) = \begin{cases}1, & if\ P(y=1|x) \geq P(y=-1|x);\\ -1, & if\ P(y=1|x) < P(y=-1|x);\end{cases} \tag{18}$$

i.e., $f_{L_{linex},\rho}(x) = f_c(x)$. ■

B. ERROR BOUND OF LINEX-SVM

In classification problems, the symbolic function $sgn(f)$ of the function $f : X \to \mathbb{R}$ is usually used as a classifier. In this way, we have the classification error as

$$\mathcal{R}_{L_{mis},\rho}(sgn(f)) = \int_{X\times Y} L_{mis}(1 - yf(x))d\rho \tag{19}$$

where $L_{mis}(x)$ is the mis-classification loss defined as

$$L_{mis}(x) = \begin{cases}1 & x < 0\\ 0 & x \geq 0\end{cases}$$

As the minimal true classification error is given by $\mathcal{R}_{L_{mis},\rho}(f_c)$, the ability of the classifier can be evaluated by $\mathcal{R}_{L_{mis},\rho}(sgn(f)) - \mathcal{R}_{L_{mis},\rho}(f_c)$. Assuming that the i.i.d. condition is satisfied in sampling, we can expect that the value of $\mathcal{R}_{L_{mis},\rho}(sgn(f)) - \mathcal{R}_{L_{mis},\rho}(f_c)$ tends to zero in probability with the increasing number of samples. And the convergence has been extensively studied in [25]. For the hinge loss, the upper bound on the mis-classification error, known as Zhang's inequality, was given in [26].

$$\mathcal{R}_{L_{mis},\rho}(sgn(f)) - \mathcal{R}_{L_{mis},\rho}(f_c) \leq \mathcal{R}_{L_{hinge},\rho}(f) - \mathcal{R}_{L_{hinge},\rho}(f_c)$$

According to $\mathcal{R}_{L_{hinge},\rho}(f) \leq \mathcal{R}_{L_{linex},\rho}(f), \forall f$,which results from $0 \leq L_{hinge}(x) \leq L_{linex}(x) (\exists a, \forall x \in \mathbb{R})$, and the facts that $\mathcal{R}_{L_{linex},\rho}(f_c) = \mathcal{R}_{L_{hinge},\rho}(f_c)$, we can bound the classification error for the LINEX loss:

$$\begin{aligned}&\mathcal{R}_{L_{mis},\rho}(sgn(f)) - \mathcal{R}_{L_{mis},\rho}(f_c)\\ &\quad \leq \mathcal{R}_{L_{hinge},\rho}(f) - \mathcal{R}_{L_{hinge},\rho}(f_c)\\ &\quad \leq \mathcal{R}_{L_{linex},\rho}(f) - \mathcal{R}_{L_{linex},\rho}(f_c)\end{aligned}$$

Theorem 2: $\forall a > 0$, for any probability measure ρ and any measurable function $f : X \to \mathcal{R}$, we have

$$\mathcal{R}_{L_{mis},\rho}(sgn(f)) - \mathcal{R}_{L_{mis},\rho}(f_c) \leq \mathcal{R}_{L_{linex},\rho}(f) - \mathcal{R}_{L_{linex},\rho}(f_c)$$

In conclusion, we illustrate the proof that our LINEX-SVM achieves Bayes rule and give an error bound for it. At last, we should also mention that compared with C-SVM and LSSVM, an additional parameter a has been involved in tuning the degree of penalty. Conventionally, the appreciate parameter a is selected by cross-validation method, and the details are described in Section VI.

V. ALGORITHM

In this part, we introduce the algorithm we adopt for the proposed LINEX-SVM. It is based on an accelerated gradient descent method while a simulated annealing method is combined to tune the learning rates.

Stochastic gradient descent (SGD) algorithm [27], [28] is a well-known algorithm to solve many convex optimization problems. With the prosperity of the researches in deep learning fields, SGD algorithm becomes increasingly valuable [29]. Compared with many other frequently used gradient-related algorithms, for example, dual coordinate descent [30], trust region Newton methods [31], concave-convex procedure [32], alternating direction method of multipliers [33], the SGD method is easily manipulated. The reason is that the unique iteration method of SGD is different from others. At each iteration, SGD updates the classifier with a small step along a random direction which is approximate to the negative gradient direction to make the objective loss decrease. Commonly, during SGD training, one or a small batch of instances are used for each iteration. In this way, it reduces the amount of calculation and works much faster, especially in large scale problems. However, sometimes this manner would get stuck in local optima during its process of convergence since its randomness of the sample chosen. To overcome this drawback, many researchers contribute to improve SGD and propose accelerated variances [34], [35]. Momentum method [36] is an effective approach that helps SGD to achieve lighter oscillation and to convergent faster. The goal is achieved by combining the update gradient direction of the previous step with the current gradient direction. With the momentum method, SGD could escape local optima.

Nesterov accelerated gradient (NAG) is one of the representing methods. NAG method gives an approximate future position to the update parameter at the next time stamp and calculates the gradient of the current position to update the momentum model. Theoretically, NAG algorithm improves the convergence speed from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/k^2)$, where k refers to k steps update [37].

Since the advantage of NAG, we construct the algorithm according to NAG to solve the LINEX-SVM problem. One main problem for NAG is that it is challenging to select an appropriate learning rate during the training. When the learning rate is very small, the convergence speed of the algorithm is very slow. On the contrary, a large learning rate probably causes the algorithm to miss of the optimal point or even non-convergence. An intuitive idea is starting with a slightly large learning rate and then gradually decrease the scale of it according to the predefined schedule during the learning process. So the specific method of decreasing the learning rate is the critical point. Lightly decreasing would waste computation resources and achieve little improvement. Aggressively decreasing would increase the probability for the learning system of falling into local optima, instead of reaching the global optimum. In this paper, inspiring by simulated annealing idea [38] and considering our model with the exponential additional term, we adopt the exponential decay way: $\eta_{new} = \eta_{old}e^{-kt}$, where k is a hyper-parameter controlling the decay degree of learning rate at each iteration and t is the number of current iterations.

Besides, we also follow the idea of the mini-batch method, other than updating the model with one single sample every iteration. In this way, utilizing multiple instances at each iteration can decrease the variance of the gradient and lead to stable convergence.

Specifically, the details of the algorithm is shown in Algorithm 1. For simplify, we use the denotation: $[b, w] \rightarrow [w]$ and $[1, x] \rightarrow [x]$. Moreover, for the hyper-parameters in the algorithm, we use the grid search method and manual parameter adjustment to select appropriate parameters.

VI. EXPERIMENTS

First, we make a comparison between our method and some other popular SVM methods on 12 small-size datasets and 5 large-size datasets. Then, we expend our experiments to the multi-class scenario. Further, the kernel method are adopted to evaluate the performance of LINEX-SVM as a nonlinear classifier. Besides, we also do the parameter study to analyze the effect of the main parameters and the robust analysis of LINEX-SVM. Finally, we give an intuitive case study to show the connection and the difference between LINEX-SVM and LSSVM. All experiments are operated in MATLAB 2015 on a PC equipped with Intel Core I5 3.10GHz processor, 4 GB RAM and 64-bits operating system of windows.

Algorithm 1 Nesterov Accelerated Gradient(NAG) for the LINEX-SVM

```
Require:
    The instances set: $(x_1, y_1), (x_2, y_2) \ldots, (x_l, y_l)$;
    The parameters: penalty weight $C$, LINEX loss parameter $a$, maximum iteration number $T$, error tolerance $e$ mini-batch size $m$; learning rate decay factor $k$, momentum parameter $r$,
    Initialize: model parameter $w_0$, learning rate $\eta_0$, velocity $v_0$;
Ensure:
    The classifier parameter: $w$;
1: Randomly choosing $k$ samples: $(x_1, y_1) \ldots, (x_k, y_k)$;
2: Temporary update: $\widetilde{w}_t = w_t + rv_t$
3: Computing
   $grad(\widetilde{w}_t) = \frac{\widetilde{w}_t}{l} + \frac{aC}{k}\sum_{i=1}^{k} y_i x_i\{exp[a(y_i(\widetilde{w}_t^\top x_i) - 1)] - 1\}$
4: Updating velocity: $v_t = rv_t - \eta_t grad(\widetilde{w}_t)$;
5: Updating model parameters: $w_{t+1} = w_t + v_t$;
6: Updating learning rate: $\eta_{t+1} = \eta_t e^{-kt}$;
7: Updating current iteration number: $t = t + 1$
8: Repeating 1,2,3,4 until convergence or up to the maximum iteration number $T$;
9: return $w$;
```

TABLE 1. Characteristics of UCI datasets in the experiments.

	Dataset	Size	Attribute	Class
Small-size	Australian	690	15	2
	Bupa_liver	345	7	2
	CMC	1473	10	2
	German	1000	21	2
	Hepatitis	155	20	2
	Ionosphere	351	35	2
	Sonar	208	61	2
	Votes	435	17	2
	WPBC	198	34	2
	Pima	768	9	2
	Arrhythmia	452	280	2
	Svmguide	1242	23	2
Large-size	a8a	32561	124	2
	a9a	48842	123	2
	ijcnn1	141691	23	2
	codrna	59535	9	2
	rcv1	20242	47237	2
Multi-class	Wine	178	13	3
	Seeds	210	7	3
	Vehiabc	282	19	4
	Iris	150	4	3
	Dermatology	366	34	6

A. EXPERIMENT ON SMALL-SIZE DATASETS

In this section, we estimate the performance of LINEX-SVM on 12 small-scale UCI datasets [45]. The characteristics of the datasets are summarized in the first section of Table 1.

TABLE 2. The fivefold cross-validation accuracy on binary datasets with linear classifiers.

Dataset	LINEX-SVM			C-SVM	LSSVM	TWSVM	NPSVM
	Accuracy	Parameter					
		a	C				
Australian	**87.10** ± **3.81**	-4	-2	85.51 ± 3.90	86.94 ± 3.85	86.66 ± 4.05	86.51 ± 3.90
Bupa_liver	**69.56** ± **3.06**	-8	-8	68.12 ± 3.69	66.67 ± 3.97	66.99 ± 7.42	68.90 ± 6.61
CMC	67.68 ± 1.93	-4	-1	67.44 ± 1.83	67.62 ± 1.97	**67.96** ± **1.69**	66.74 ± 3.06
German	**76.70** ± **2.25**	-3	1	76.30 ± 1.48	75.50 ± 2.57	71.10 ± 3.19	73.70 ± 3.67
Hepatitis	**85.80** ± **8.71**	-3	1	**85.81** ± **8.10**	85.45 ± 7.70	79.68 ± 11.31	85.16 ± 7.43
Ionosphere	87.46 ± 4.99	-3	1	87.31 ± 3.12	86.04 ± 4.68	89.74 ± 4.35	**90.02** ± **3.37**
Sonar	79.81 ± 4.68	-5	-4	78.35 ± 3.06	75.94 ± 6.02	76.88 ± 6.92	**80.23** ± **4.25**
Votes	94.71 ± 1.31	-7	-7	**95.86** ± **1.54**	95.40 ± 1.15	95.17 ± 1,26	95.63 ± 1.26
WPBC	**81.35** ± **3.95**	-4	-1	78.79 ± 2.80	77.76 ± 6.13	75.79 ± 4.71	78.39 ± 8.50
Pima	**77.47** ± **2.63**	-4	-1	65.11 ± 3.44	64.46 ± 1.16	76.57 ± 3.75	77.47 ± 3.33
Arrhythmia	**76.67** ± **5.53**	-1	-8	76.19 ± 5.13	75.85 ± 7.15	76.43 ± 5.77	76.53 ± 7.19
Svmguide	**80.84** ± **1.70**	-6	-4	78.43 ± 1.98	78.42 ± 2.20	76.17 ± 2.28	80.12 ± 2.13

TABLE 3. The fivefold cross-validation AUC on binary datasets with linear classifiers.

Dataset	LINEX-SVM			C-SVM	LSSVM	TWSVM	NPSVM
	AUC	Parameter					
		a	C				
Australian	**86.92** ± **2.90**	-6	-7	86.39 ± 3.16	86.91 ± 2.64	86.59 ± 3.07	86.39 ± 3.16
Bupa_liver	**66.34** ± **2.57**	-8	-7	66.08 ± 5.97	65.25 ± 5.61	63.87 ± 6.85	60.38 ± 6.52
CMC	**65.77** ± **1.57**	-4	-1	65.65 ± 2.22	65.23 ± 2.65	64.98 ± 2.36	62.92 ± 3.16
German	**69.11** ± **3.56**	-4	-1	67.07 ± 2.74	66.07 ± 2.04	70.17 ± 2.97	57.77 ± 3.21
Hepatitis	**73.45** ± **16.71**	-8	-9	72.81 ± 21.55	73.27 ± 15.64	72.08 ± 9.00	70.44 ± 12.68
Ionosphere	84.57 ± 3.83	-6	-6	85.81 ± 4.17	82.12 ± 4.47	47.15 ± 4.26	**88.45** ± **4.45**
Sonar	**79.89** ± **7.80**	-8	-9	79.46 ± 4.82	75.48 ± 8.21	76.04 ± 8.97	79.81 ± 5.33
Votes	93.91 ± 1.17	-6	-5	96.14 ± 1.25	95.03 ± 1.90	95.40 ± 1.17	**96.27** ± **0.90**
WPBC	68.73 ± 9.98	-4	-1	69.88 ± 7.31	72.94 ± 8.87	**79.64** ± **6.92**	65.39 ± 7.09
Pima	**73.47** ± **6.03**	-1	9	53.02 ± 3.16	55.70 ± 3.59	76.27 ± 2.16	71.67 ± 7.00
Arrhythmia	**74.41** ± **4.12**	-1	-9	74.16 ± 4.44	71.50 ± 1.04	66.86 ± 7.70	74.39 ± 5.75
Svmguide	**63.66** ± **3.31**	-6	-4	56.70 ± 2.48	58.13 ± 3.87	49.82 ± 2.36	59.99 ± 3.13

The data features are normalized into [0,1] before training. For the benchmarks, we compare LINIX-SVM with another four SVM-based classical classifiers: C-SVM [7], LSSVM [10], TWSVM [13], NPSVM [15]. Among these SVM models, LINEX-SVM, TWSVM and NPSVM have extra hyper-parameters, in addition to the hyper-parameter C. All the hyper-parameters are chosen through the grid search method and manual adjustment. And the five-fold cross-validation evaluation method is also employed. The details of the search range and the stride are as follows: (1) penalty parameter C is chosen from 2^{-10} to 2^{10}, with the step of 2^1. (2) LINEX loss parameter a is chosen from -1 to -10, with the step -1. (3) Parameter ϵ in NPSVM is chosen from 0 to 0.5, with the step 0.1. The parameters for NAG algorithm are set experimentally as: (1) Initial learning rate $\eta_0 = 0.01$. (2) Learning rate decay parameter $k = 0.1$. (3) Initial weight $w_0 = 0$. (4) Initial momentum $v_0 = 0$ (5) Momentum parameter $r = 0.6$ (6) Error tolerance $e = 10^{-8}$. (7) Mini-batch size $m = 100$. (8) Maximum iteration number $T = 5000$.

The experimental results and the selected parameters for our method are listed in Table 2. The best results are shown in boldface. Besides accuracy, we also illustrate the average five-fold cross validation AUC and F1-score in Table 3 and Table 4, respectively. From the experiment outcomes, we can learn that LINEX-SVM model outperforms other methods on the majority of the 12 datasets (8 out of 12 datasets in terms of accuracy and F1-score, 9 out of 12 datasets in terms of AUC). This proves the effectiveness of LINEX-SVM. Another interesting scenario is LINEX-SVM achieves better results than LSSVM on all the datasets and the metrics. This is because LINEX-SVM can degenerate into LSSVM and the experiment results are consistent with the theory.

TABLE 4. The fivefold cross-validation F1-score on small-scale binary datasets with linear classifiers.

Dataset	LINEX-SVM			C-SVM	LSSVM	TWSVM	NPSVM
	F1-score	Parameter					
		a	C				
Australian	**85.83** ± **5.26**	-7	-8	84.95 ± 4.18	85.28 ± 4.46	85.78 ± 4.50	84.95 ± 4.18
Bupa_liver	**61.53** ± **6.43**	-7	-5	55.86 ± 7.94	53.60 ± 7.03	58.84 ± 13.51	31.65 ± 19.86
CMC	**56.32** ± **2.05**	-3	1	54.83 ± 3.44	55.69 ± 3.32	54.61 ± 3.94	50.00 ± 4.88
German	**84.19** ± **1.94**	-1	6	84.18 ± 1.15	83.71 ± 1.88	82.72 ± 2.33	83.83 ± 2.43
Hepatitis	**66.24** ± **17.97**	-6	-5	63.89 ± 18.89	65.25 ± 15.68	53.55 ± 10.81	60.82 ± 16.06
Ionosphere	91.00 ± 4.29	-8	-10	91.03 ± 3.60	89.60 ± 4.86	85.25 ± 4.84	**92.08** ± **4.49**
Sonar	**76.95** ± **5.84**	-3	-1	76.92 ± 5.05	74.47 ± 6.09	73.65 ± 7.35	76.55 ± 5.75
Votes	93.40 ± 3.04	-1	7	**94.45** ± **2.79**	93.87 ± 2.20	94.22 ± 1.51	94.32 ± 1.99
WPBC	**88.92** ± **2.55**	-2	1	86.41 ± 4.51	87.44 ± 3.76	87.24 ± 4.97	87.40 ± 3.54
Pima	**64.54** ± **5.83**	-2	5	55.20 ± 4.40	29.75 ± 6.58	60.05 ± 5.13	62.16 ± 9.69
Arrhythmia	77.99 ± 6.18	-7	-8	**81.66** ± **3.81**	76.41 ± 3.88	75.07 ± 5.12	81.60 ± 4.18
Svmguide	39.97 ± 3.95	-1	8	36.66 ± 12.09	31.96 ± 9.65	**53.02** ± **3.28**	39.31 ± 6.69

TABLE 5. Classification accuracy for large-scale datasets.

Dataset		LINEX-SVM		C-SVM		LSSVM	
Name	Size	Acuracy(%)	Time(s)	Acuracy(%)	Time(s)	Acuracy(%)	Time(s)
a8a	32561*124	**84.51** ± 0.40	**24.51**	84.42 ± 0.31	5923.36	84.39 ± 0.46	963.08
a9a	48842*124	**84.93** ± 0.50	**40.45**	84.76 ± 0.17	12073.04	84.44 ± 0.18	4009.40
ijcnn1	141691*23	**91.78** ± 0.05	**59.29**	91.41 ± 0.05	15073.41	91.52 ± 0.16	7432.00
codrna	59535*9	**88.10** ± 0.41	**67.06**	87.90 ± 0.36	1149.33	86.11 ± 0.55	942.27
rcv1	20242*47237	94.48 ± 0.32	**72.93**	**96.91** ± 0.43	19842.68	94.48 ± 0.59	17129.87

B. EXPERIMENT ON LARGE-SIZE DATASETS

In order to validate the classification efficiency of LINEX-SVM solved by NAG, we conduct comparisons between the methods and two most related methods (C-SVM, LSSVM) on 5 large-size UCI datasets: 'a9a', 'a8a', 'ijcnn1', 'codrna' and 'rcv1'. The statistic details of these datasets are shown in the middle part of Table 1. The mini-batch size is set as 10 experimentally, and the other parameters are chosen as the same as in the last two sections. Table 5 illustrates the accuracy and the running time results of the methods. The best ones are highlighted in bold. We can see, firstly, LINEX-SVM outperforms C-SVM and LSSVM on 4 datasets out of the total 5 ones in terms of accuracy. Secondly, LINEX-SVM achieves a significant improvement in terms of running time compared with C-SVM and LSSVM. For example, on dataset ijcnn1, which contains 141,691 instances in the dimension of 23, LINEX-SVM achieves the best classification performance with a 91.78% accuracy, and it only needs around 59 seconds, while CSVM needs around $15,073$ seconds and LSSVM needs around $7,432$ seconds. On these five datasets, the average running time for LINEX-SVM is 52.85 seconds, while for C-SVM the average running time is 10812.36 seconds and for LSSVM is 6095.32 seconds. It means, on average, LINEX-SVM achieves around 204 times faster compared with C-SVM and around 115 times faster compared with LSSVM.

C. EXPERIMENTS ON MULTI-CLASS DATASETS

In this section, we conduct comparisons on 5 multi-class UCI datasets: 'Wine', 'Seeds', 'Vehiabc', 'Iris' and 'Dermatology', to evaluate the effectiveness of LINEX-SVM on multi-classification problems. The statistic details of these datasets are shown in the bottom part of Table 1. The One-vs-all strategy is adopted in this experiment and the test results are shown in Table 9.

The results indicate that LINEX-SVM obtains best results on 4 datasets out of 5. It achieves an average of 2.02% improvement on accuracy compared with C-SVM, an average 1.39% improvement compared with LSSVM, an average 1.36% improvement compared with TWSVM and an average 0.13% improvement compared with NPSVM. This proves the superior of LINEX-SVM in solving multi-class problems.

D. EXPERIMENTS ON SVMS WITH KERNEL

In Section III-B, we have derived the dual formulation eq. (13) and its vector form eq. (14). The dual problem is also a convex optimization problem, so NAG algorithm can be applied easily on the dual problem. We expand the experiments on nonlinear classification problems by introducing

TABLE 6. The fivefold cross-validation accuracy on small-scale binary datasets with RBF kernel.

Dataset	LINEX-SVM			C-SVM	LSSVM	TWSVM	NPSVM
	Accuracy	Parameter					
		a	C				
Australian	**87.25 ± 3.72**	-5	-3	85.65 ± 4.08	85.51 ± 3.24	85.07 ± 3.49	85.80 ± 2.54
Bupa_liver	**66.96 ± 4.74**	-2	5	64.64 ± 4.30	66.28 ± 3.46	58.84 ± 7.43	66.46 ± 4.91
CMC	68.36 ± 1.83	-3	1	**71.63 ± 2.42**	69.12 ± 2.37	70.43 ± 5.57	71.36 ± 2.52
German	**76.80 ± 1.79**	-5	-2	76.40 ± 1.43	76.50 ± 1.41	75.20 ± 2.08	76.00 ± 2.57
Hepatitis	**88.39 ± 6.29**	-1	8	85.81 ± 7.77	87.10 ± 3.23	84.71 ± 11.06	85.81 ± 7.77
Ionosphere	92.87 ± 3.65	-1	8	94.58 ± 2.13	94.30 ± 2.26	93.75 ± 4.35	**94.58 ± 1.87**
Sonar	**85.53 ± 4.00**	-2	7	84.07 ± 7.75	76.36 ± 4.04	83.21 ± 5.16	85.48 ± 2.12
Votes	93.56 ± 1.92	-8	-7	95.63 ± 1.50	95.17 ± 1.89	92.87 ± 3.85	**95.63 ± 1.26**
WPBC	**79.34 ± 2.08**	-5	-1	78.88 ± 4.09	77.80 ± 5.36	78.24 ± 3.89	79.32 ± 1.13
Pima	**78.12 ± 2.51**	-8	-6	65.11 ± 3.44	71.37 ± 2.72	65.11 ± 3.44	77.39 ± 3.15
Arrhythmia	72.78 ± 3.97	-8	-7	**77.40 ± 6.32**	73.01 ± 1.54	55.94 ± 4.75	76.06 ± 6.53
Svmguide	**82.13 ± 0.72**	-2	4	79.23 ± 2.32	81.45 ± 1.32	76.49 ± 2.75	81.79 ± 1.44

TABLE 7. The fivefold cross-validation AUC value on small-scale binary datasets with RBF kernel.

Dataset	LINEX-SVM			C-SVM	LSSVM	TWSVM	NPSVM
	AUC	Parameter					
		a	C				
Australian	**87.04 ± 2.65**	-9	-10	86.57 ± 3.36	83.57 ± 3.57	86.15 ± 3.46	85.58 ± 4.26
Bupa_liver	64.79 ± 5.09	-7	-5	66.31 ± 3.43	67.40 ± 7.48	59.44 ± 4.79	**70.87 ± 5.37**
CMC	**65.79 ± 1.85**	-6	-5	65.27 ± 3.35	65.69 ± 3.19	65.07 ± 1.82	65.39 ± 3.19
German	**69.05 ± 2.29**	-1	10	68.54 ± 3.75	68.86 ± 3.42	68.56 ± 2.19	62.08 ± 2.84
Hepatitis	**79.20 ± 14.71**	-8	-9	77.15 ± 19.91	66.51 ± 10.92	70.42 ± 11.55	73.82 ± 10.25
Ionosphere	92.46 ± 2.54	-1	8	92.85 ± 1.80	93.22 ± 3.06	91.73 ± 3.56	**94.28 ± 1.89**
Sonar	**88.10 ± 6.90**	-4	1	83.85 ± 9.57	66.10 ± 10.80	84.62 ± 7.15	83.01 ± 10.56
Votes	93.54 ± 0.97	-4	1	96.27 ± 0.90	95.48 ± 1.61	**96.29 ± 1.36**	92.19 ± 2.56
WPBC	**69.49 ± 13.86**	-5	-2	67.49 ± 11.72	60.51 ± 10.45	67.38 ± 14.85	63.73 ± 8.93
Pima	**73.85 ± 5.62**	-1	9	61.38 ± 4.80	66.23 ± 4.16	72.41 ± 3.22	71.89 ± 7.24
Arrhythmia	**73.37 ± 0.99**	-8	-9	73.18 ± 5.35	69.80 ± 1.75	72.92 ± 7.14	72.40 ± 2.87
Svmguide	**67.47 ± 2.11**	-2	4	57.93 ± 3.35	67.01 ± 5.51	67.04 ± 2.58	72.32 ± 2.96

kernel to the dual problem, where RBF kernel is adopted. For the parameter setting, the parameter of the RBF kernel is set as 2 through many times manual adjustment and the other parameters are set as the same as the linear case.

Specifically, the experimental results are listed in Table 6, Table 7 and Table 8, in terms of accuracy, AUC and F1-score, respectively. The related values of parameters a and C in LINEX-SVM are also illustrated in the tables. From them, we can see LINEX-SVM achieves the best classification on majority datasets, compared with the baselines. This demonstrates that LINEX-SVM performance well on nonlinear classification problems by using kernels.

E. PARAMETER STUDY

In order to explore the effect of the parameters, we conduct a study on several critical parameters. In LINEX-SVM, two parameters are involved: parameter a in LINEX loss function controlling the steepness of the function curve and parameter m controlling the size of mini-batch in SGD algorithm. For simplicity, we choose four small-size UCI datasets we used in Section VI-A as representations, i.e., 'German', 'Hepatitis', 'Ionosphere' and 'Pima', to study the parameters.

First, we study the choice of parameter a. The results are shown in Fig.3. We can see the accuracy remains relatively high level while a is negative and the accuracy decreases dramatically when a become positive. This outcome is consistent with the theory. This proves the correctness of our hypothesis that the instances between the two center hyperplanes should have heavier penalties, with practical results. Therefore, the value of parameter a in LINEX-SVM do not need to be positive, and we can find the best value in a small range of negative values using the grid search. In addition, we also did an experiment on the mini-batch size. Experiment results are also shown in Fig.3, while the black curves in the

TABLE 8. The fivefold cross-validation F1-score on small-scale binary datasets with RBF kernel.

Dataset	LINEX-SVM			C-SVM	LSSVM	TWSVM	NPSVM
	F1-score	Parameter					
		a	C				
Australian	**86.06 ± 3.87**	-9	-10	85.07 ± 4.36	81.88 ± 3.71	85.70 ± 4.66	84.19 ± 4.59
Bupa_liver	54.89 ± 8.41	-7	-5	64.18 ± 7.77	59.82 ± 9.71	59.62 ± 5.71	**64.22 ± 7.98**
CMC	**58.20 ± 1.17**	-8	-7	58.11 ± 5.46	57.81 ± 4.35	56.16 ± 2.02	56.48 ± 5.22
German	**84.27 ± 2.35**	-7	-4	84.12 ± 1.17	84.07 ± 1.45	76.04 ± 2.68	83.09 ± 1.69
Hepatitis	**72.57 ± 16.68**	-8	-9	65.45 ± 20.02	59.84 ± 11.85	57.29 ± 6.52	68.31 ± 11.42
Ionosphere	94.25 ± 3.56	-1	8	95.61 ± 2.45	95.14 ± 2.58	91.86 ± 5.33	**95.62 ± 2.55**
Sonar	**83.87 ± 4.84**	-4	1	83.24 ± 7.80	80.43 ± 13.20	82.21 ± 5.81	83.27 ± 9.88
Votes	91.09 ± 3.21	-7	-8	**94.32 ± 1.99**	93.59 ± 3.69	94.11 ± 2.22	91.80 ± 1.96
WPBC	**87.36 ± 2.67**	-1	8	87.16 ± 2.79	87.12 ± 3.51	80.00 ± 6.13	87.26 ± 1.80
Pima	**65.79 ± 4.57**	-1	9	55.35 ± 4.62	54.29 ± 6.92	64.54 ± 4.54	62.09 ± 7.71
Arrhythmia	**76.59 ± 3.38**	-8	-9	76.00 ± 5.57	72.25 ± 5.77	74.81 ± 9.43	76.06 ± 4.09
Svmguide	51.32 ± 3.95	-2	4	31.51 ± 9.57	53.96 ± 6.53	50.20 ± 5.74	**60.13 ± 4.18**

TABLE 9. Classification accuracy on multi-class UCI datasets.

Dataset	LINEX-SVM	C-SVM	LSSVM	TWSVM	NPSVM
Wine	96.05 ± 1.61	96.66 ± 2.29	96.13 ± 2.25	**98.33** ± 2.66	96.78 ± 7.14
Seeds	**91.90 ± 1.30**	91.71 ± 1.99	90.95 ± 4.26	91.81 ± 5.04	91.80 ± 8.99
Vehiabc	**64.05 ± 4.84**	61.77 ± 8.26	63.12 ± 8.75	63.35 ± 5.21	63.92 ± 4.74
Iris	**96.00 ± 4.35**	89.33 ± 5.48	92.00 ± 9.55	92.67 ± 7.34	95.17 ± 9.32
Dermatology	**97.56 ± 5.93**	97.27 ± 0.97	97.27 ± 2.17	93.42 ± 3.52	97.32 ± 2.84

middle figures show how the accuracy changes in terms of batch size and the red curves in the bottom figures show how the running time changes in terms of batch size. We can see when the size of mini-batch increases, the accuracy increases as well in the beginning then remains stable. And the running time increases approximately linearly when the batch size increases. This suggests that in the scenarios that we care more about the accuracy, we should choose a relatively medium batch size (it does not need to be large), while in the other scenarios that we care more about the efficiency, then we should choose a relatively small batch size.

F. ROBUST ANALYSIS

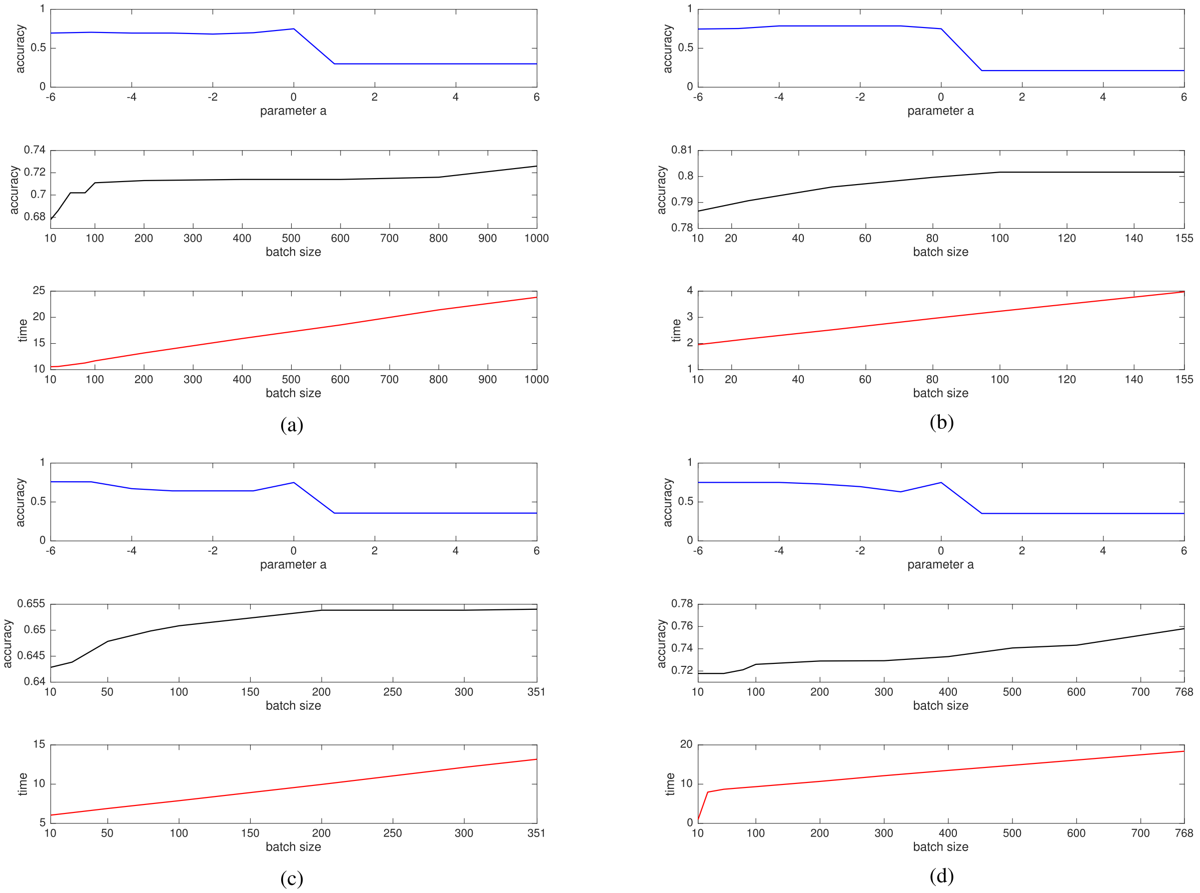
To illustrate the performance of LINEX-SVM intuitively, we generate two two-dimensional datasets and explore the robustness of LINEX-SVM on them. The data in dataset 1 are from two Gaussian distributions with equal probability: $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$, $\boldsymbol{x}_j \sim \mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$, where $\boldsymbol{\mu}_1 = [0.5, -3]^T$, $\boldsymbol{\mu}_2 = [-0.5, 3]^T$, and $\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}$, $\boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 3 \end{bmatrix}$. We display the training set and the classifier in Fig.4(a). The positive points are marked as red while the negative points are marked as blue. And the number of points in both class is 100. The blue line is the classification boundary. Then we add noise into dataset 1 to get the dataset 2. The positions of noise points locate at the edge of all the data where is far from the classification boundary. These noise points come from two Gaussian distributions: $\boldsymbol{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_3, \boldsymbol{\Sigma}_3)$, $\boldsymbol{x}_j \sim \mathcal{N}(\boldsymbol{\mu}_4, \boldsymbol{\Sigma}_4)$, where $\boldsymbol{\mu}_3 = [1, -7]^T$, $\boldsymbol{\mu}_4 = [-1, 7]^T$ and $\boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.2 & 0 \\ 0 & 3 \end{bmatrix}$, $\boldsymbol{\Sigma}_4 = \boldsymbol{\Sigma}_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 3 \end{bmatrix}$. The dataset with noise is shown in Fig.4(b). The labels of the noise points are contrary to surrounding points' labels. We add 10 noise points to two class respectively. That is to say that noise accounts for 10% of the total data. For intuition, we circle the noise points in the picture.

Comparing two classification boundary, we can find that such noise does not seriously affect the classification results. The noise only brings minor changes to the classifier. This experiment shows that the model is insensitive to the noise far from the classifier. As we introduced, LINEX-SVM gives different penalties to different points. It punishes the points around classification boundary heavily while punishes the points far from classification boundary slightly. Therefore, the model is insensitive for the noise distributed away from the classification boundary.

G. CASE STUDY

At last, we give an intuitive case, to show the differences between LINEX-SVM and LSSVM.

In LSSVM, minimizing $\frac{1}{2}\|w\|^2$ realizes the maximal margin between the two center hyperplanes $(w \cdot x_i) + b = 1$



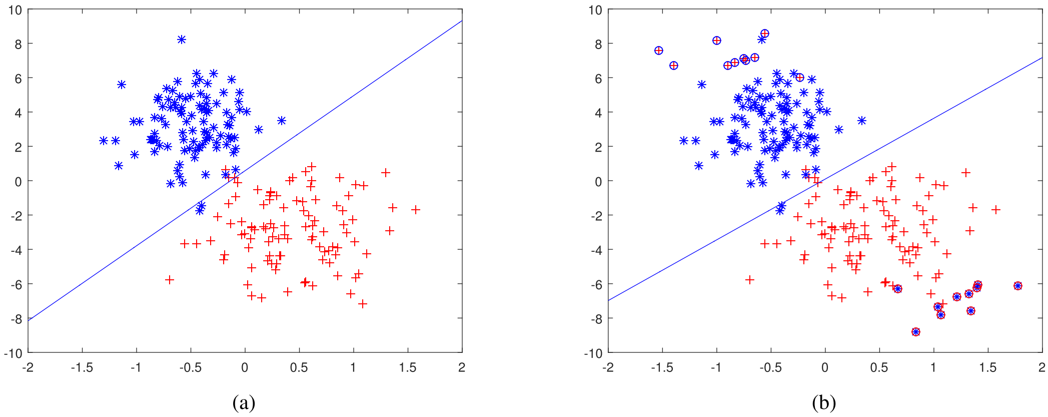
(a)

(b)

(c)

(d)

FIGURE 3. Parameter study on the size of mini-batch and loss parameter *a*. The experiment results are respectively on dataset 'German', 'Hepatitis', 'Ionosphere', and 'Pima'. (a) German(1000*21). (b) Hepatitis (155*20). (c) Ionosphere (351*35). (d) Pima (768*9).



(a)

(b)

FIGURE 4. The performance on data with noise. Positive points and negative points are marked as red crosses and blue stars, respectively. The blue line is the classification boundary. (a)Dataset 1: 100 positive points and 100 negative points. (b)Dataset 2: besides samples in dataset 1, another ten noise samples are added to each class. The results illustrate that the model is robust for such noise.

and $(w \cdot x_i) + b = -1$, while minimizing the square loss $\sum_{i=1}^{l} \eta_i^2$ is to make the two hyperplanes close to the center of positive and negative points, respectively. However, in LINEX-SVM, based on the idea that the points closed to the hyperplane deserve heavier punishment than the far-side points, the asymmetric LINEX loss function is adopted. The minimizing of $\sum_{i=1}^{l}(exp(a\xi_i) - a\xi_i - 1)$ implies not only making the two straight blue lines close to the center of points respectively but also paying more attention to the points between the two straight blue lines.

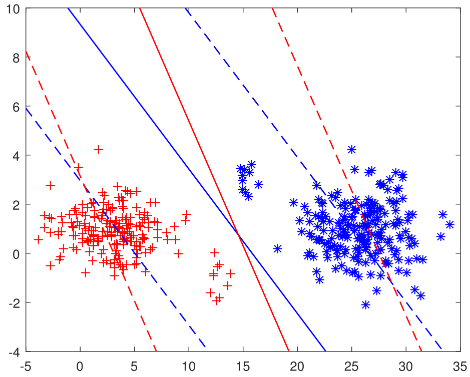


FIGURE 5. Case study on the comparison between LINEX-SVM and LSSVM, in the linear case. Points in the same color share the same label. The hyperplanes obtained by LINEX-SVM are shown in blue while the one obtained by LSSVM are showed in red. The decision hyperplane $(\boldsymbol{w} \cdot \boldsymbol{x}_i) + b = 0$ is marked with solid lines, while the center hyperplanes $(\boldsymbol{w} \cdot \boldsymbol{x}_i) + b = \pm 1$ are represented with dash lines. Clearly, in the picture, the LINEX-SVM model pays more attention to the two cluster minority points distributed around the decision hyperplane.

In Fig.5, we compare LINEX-SVM and LSSVM with a demo. First, we generate two kinds of points that satisfy normal distribution to observe the differences between LINEX-SVM and LSSVM. We can see LINEX-SVM obtain a more reasonable classification hyperplane since LINEX-SVM concentrates on the points closed to the hyperplane. This result suggests that the LINEX loss may achieve better classification results than LSSVM by paying more attention to the points around the decision hyperplanes.

In conclusion, for LINEX-SVM, the experiments in this section demonstrate that: (1) LINEX loss is very useful in improving classification performance. It is a convex asymmetric loss function and is superior compared with the baseline loss functions in many scenarios. (2) No mater on small-size datasets or large-size datasets, or in multi-class problem, or for nonlinear classification, LINEX-SVM performs well and achieves best results on most of the datasets. (3) LINEX-SVM is extremely efficient. It is much faster than C-SVM and LSSVM on large-size datasets and achieves an obvious scale of decrease of running times compared with the baselines. (4) NAG algorithm is efficient in solving the convex optimization problems, since it adopts an adaptive learning rate according to the simulated annealing method. Compared with the common SGD algorithm, NAG algorithm normally convergent faster.

VII. CONCLUSION

In this paper, we bring the LINEX loss into SVM for classification and propose the LINEX-SVM classifier. LINEX-SVM makes use of all the points and penalizes the points around the classification boundary heavier to achieve better classification performance. Further, we made a theoretical analysis of our method, i.e. 1) the expected risk minimizing of LINEX-SVM satisfies the Bayes rule and 2) the classification error bound has a theoretical upper bound. Then the NAG algorithm is adopted to solve LINEX-SVM, which has a significant advantage in handling large-scale classification problem. In order to verify the performance of LINEX-SVM, we evaluate it on some UCI standard datasets in terms of both accuracy, AUC value, F1-score and the training time. The experimental results demonstrate that our method is competitive and efficient compared with other baselines.

In further work, we would like to consider a truncated LINEX loss to make the model more robust. And the LINEX loss can also be used in other machine learning approaches to solve different tasks.

ACKNOWLEDGMENT

(Yue Ma and Qin Zhang contributed equally to this work.)

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[2] V. N. Vapnik and V. Vapnik, *Statistical Learning Theory*, vol. 1. New York, NY, USA: Wiley, 1998.

[3] T. B. Trafalis and H. Ince, "Support vector machine for regression and applications to financial forecasting," in *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 6, Jul. 2000, pp. 348–353.

[4] W. S. Noble, B. Scholkopf, K. Tsuda, and J. P. Vert, "Support vector machine applications in computational biology," in *Proc. Kernel Methods Comput. Biol.*, 2004, pp. 71–92.

[5] K.-S. Goh, E. Y. Chang, and B. Li, "Using one-class and two-class SVMs for multiclass image annotation," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 10, pp. 1333–1346, Oct. 2005.

[6] D. Isa, L. H. Lee, V. P. Kallimani, and R. RajKumar, "Text document preprocessing with the Bayes formula for classification using the support vector machine," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1264–1272, Sep. 2008.

[7] J. Bi and T. Zhang, "Support vector classification with input data uncertainty," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 161–168.

[8] L. Rosasco, E. Vito, A. Caponnetto, M. Piana, and A. Verri, "Are loss functions all the same?" *Neural Comput.*, vol. 16, no. 5, pp. 1063–1076, May 2004.

[9] C. Saunders, A. Gammerman, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proc. 15th Int. Conf. Mach.*, Jul. 1998, pp. 515–521.

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

[11] V. Vapnik, "The support vector method of function estimation," in *Proc. Nonlinear Model.*, vol. 55, 1998, p. 55–85.

[12] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 35, no. 1, pp. 73–101, 1964.

[13] R. Khemchandani and S. Chandra, "Twin support vector machines for pattern classification," *Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 5, pp. 905–910, May 2007.

[14] X. Huang, L. Shi, and J. A. K. Suykens, "Support vector machine classifier with pinball loss," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 984–997, May 2014.

[15] Y. Tian, Z. Qi, X. Ju, Y. Shi, and X. Liu, "Nonparallel support vector machines for pattern classification," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1067–1079, Jul. 2014.

[16] D. Liu, Y. Shi, and Y. Tian, "Ramp loss nonparallel support vector machine for pattern classification," *Knowl.-Based Syst.*, vol. 85, pp. 224–233, Sep. 2015.

[17] A. Christmann and I. Steinwart, "On robustness properties of convex risk minimization methods for pattern recognition," *J. Mach. Learn. Res.*, vol. 5, pp. 1007–1034, Aug. 2004.

[18] F. Cucker and D.-X. Zhou, *Learning Theory: An Approximation Theory Viewpoint*, vol. 24. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[19] H. R. Varian, "A Bayesian approach to real estate assessment," in *Studies in Bayesian Econometric and Statistics in honor of Leonard J. Savage*, 1975, pp. 195–208.

[20] A. Zellner, "Bayesian estimation and prediction using asymmetric loss functions," *J. Amer. Stat. Assoc.*, vol. 81, no. 394, pp. 446–451, Jun. 1986.

[21] M. Cain and C. Janssen, "Real estate price prediction under asymmetric loss," *Ann. Inst. Stat. Math.*, vol. 47, no. 3, pp. 401–414, Sep. 1995.

[22] G. Zou, "Admissible estimation for finite population under the linex loss function," *J. Stat. Planning Inference*, vol. 61, no. 2, pp. 373–384, Jun. 1997.

[23] A. Parsian and N. S. Farsipour, "Estimation of the mean of the selected population under asymmetric loss function," *Metrika*, vol. 50, no. 2, pp. 89–107, Dec. 1999.

[24] K. Ohtani, "Generalized ridge regression estimators under the LINEX loss function," *Stat. Papers*, vol. 36, no. 1, pp. 99–110, Dec. 1995.

[25] I. Steinwart and A. Christmann, *Support Vector Machine*. Berlin, Germany: Springer, 2008.

[26] T. Zhang, "Statistical analysis of some multi-category large margin classification methods," *J. Mach. Learn. Res.*, vol. 5, pp. 1225–1251, Oct. 2004.

[27] T. Zhang, "Solving large scale linear prediction problems using stochastic gradient descent algorithms," in *Proc. 21th Int. Conf. Mach. Learn.*, Jun. 2004, p. 116.

[28] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal estimated sub-gradient solver for SVM," *Math. Program.*, vol. 127, no. 1, pp. 3–30, Mar. 2011.

[29] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2013, pp. 1139–1147.

[30] C. J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proc. 25th Int. Conf. Mach. Learn.*, Jul. 2008, pp. 408–415.

[31] C.-J. Lin, R. C. Weng, and S. S. Keerthi, "Trust region newton methods for large-scale logistic regression," in *Proc. 24th Int. Conf. Mach. Learn.*, Jun. 2007, pp. 561–568.

[32] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Comput.*, vol. 15, no. 4, pp. 915–936, Apr. 2003.

[33] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jul. 2011.

[34] R. Johnson and T. Zhang, "Accelerating stochastic gradient descent using predictive variance reduction," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 315–323.

[35] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proc. COMPSTAT*, 2010, pp. 177–186.

[36] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.

[37] Y. Nesterov, "A method for unconstrained convex minimization problem with the rate of convergence o($1/k^2$)," in *Proc. Doklady AN USSR*, vol. 269, 1983, pp. 543–547.

[38] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[39] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *Proc. Int. Conf. Mach. Learn.*, Feb. 2013, pp. 71–79.

[40] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 689–696.

[41] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, Apr. 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[42] M. Collins, A. Globerson, T. Koo, X. Carreras, and P. L. Bartlett, "Exponentiated gradient algorithms for conditional random fields and max-margin markov networks," *J. Mach. Learn. Res.*, vol. 9, pp. 1775–1822, Aug. 2008.

[43] K.-W. Chang, C.-J. Hsieh, and C.-J. Lin, "Coordinate descent method for large-scale L2-loss linear support vector machines," *J. Mach. Learn. Res.*, vol. 9, pp. 1369–1398, Jun. 2008.

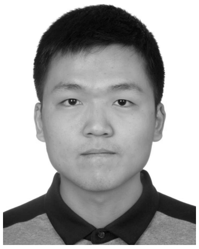
[44] K. De Brabanter *et al.*, "LS-SVM lab toolbox user's guide, version 1.8," Katholieke Univ. Leuven, Leuven, Belgium, 2011.

[45] M. Lichman. (2013). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml

[46] Y.-H. Shao, C.-H. Zhang, X.-B. Wang, and N.-Y. Deng, "Improvements on twin support vector machines," *IEEE Trans. Neural Netw.*, vol. 22, no. 6, pp. 962–968, Jun. 2011.

YUE MA received the B.S. degree in applied mathematics from Zhengzhou University, Zhengzhou, China, in 2015. She is currently pursuing the Ph.D. degree with the School of Mathematical Science, University of Chinese Academy of Sciences, Beijing, China. Her current research interests include optimization, data mining, and machine learning.

QIN ZHANG received the Ph.D. degree from the University of Technology Sydney, Australia, in 2018, the master's degree from the University of Chinese Academy of Sciences, China, in 2014. She is currently a Senior Researcher with the Data Center of Cloud and Smart Industries Group (CSIG), Tencent, China. Her main research interests include NLP, sequence data learning, and network analysis by using various deep learning and optimization methods. She has published several qualified research papers in top journals and top conferences including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (TPAMI) and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE). She has published in top conferences including the International Joint Conferences on Artificial Intelligence (IJCAI) and the IEEE International Conference on Data Mining (ICDM). She has also served as a Reviewer (subreviewer) for KDD, NIPS, ICDM, IJCAI, AAAI, and SDM.



DEWEI LI received the bachelor's and master's degrees from the Renmin University of China, in 2012 and 2015, respectively. He is currently pursuing the degree with the Chinese Academy of Sciences, Beijing, China. He is also pursuing the Ph.D. degree with the School of Mathematical Sciences, University of Chinese Academy of Sciences. His research interests include metric learning and support vector machine.

YINGJIE TIAN received the bachelor's degree in mathematics from Shandong Normal University, Jinan, China, in 1994, and the master's degree in applied mathematics from the Beijing Institute of Technology, Beijing, China, in 1997, and the Ph.D. degree in management science and engineering from China Agricultural University, Beijing, China. He is currently a Professor with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences. He has published four books on SVMs, one of which has been cited over 1,500 times. His research interests include support vector machines, optimization theory and its applications, data mining, intelligent knowledge management, and risk management.