Pattern Recognition 136 (2023) 109192

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Laplacian Lp norm least squares twin support vector machine

Xijiong Xie[a,*], Feixiang Sun[a], Jiangbo Qian[a], Lijun Guo[a], Rong Zhang[a], Xulun Ye[a], Zhijin Wang[b]

[a] The school of Information Science and Engineering, Ningbo University, China
[b] College of Computer Engineering, Jimei University, Yinjiang Road 185, Xiamen 361021, China

ARTICLE INFO

Article history:
Received 25 October 2021
Revised 8 November 2022
Accepted 18 November 2022
Available online 21 November 2022

Keywords:
Semi-supervised learning
Laplacian Lp norm least squares twin support vector machine
Lp norm graph regularization
Geometric information

ABSTRACT

Semi-supervised learning has become a hot learning framework, where large amounts of unlabeled data and small amounts of labeled data are available during the training process. The recently proposed Laplacian least squares twin support vector machine (Lap-LSTSVM) is an excellent tool to solve the semi-supervised classification problem. Motivated by the success of Lap-LSTSVM, in this paper, we propose a novel Laplacian Lp norm least squares twin support vector machine (Lap-LpLSTSVM). There are several advantages of our proposed method: (1) The performance of our proposed Lap-LpLSTSVM can be improved by the adjustability of the value of p. (2) The introduced Lp norm graph regularization term can efficiently exploit the geometric information embedded in the data. (3) An efficient iterative strategy is employed to solve the optimization problem. Besides, to demonstrate that our proposed method can make use of unlabeled data effectively, least squares twin support vector machine (LSTSVM) which only uses the same labeled data is used to compare with our proposed method. The experimental results on both synthetic and real-world datasets show that our proposed method outperforms other state-of-the-art methods and can also deal with noisy datasets.

1. Introduction

In many real-world applications such as natural language parsing (NLP) [1], spam filtering [2], it is challenging to acquire enough labeled data to train. However, the acquisition of unlabeled data is accessible. In such situations, the performance of traditional supervised algorithms will deteriorate because of the insufficient labeled data. Therefore, semi-supervised learning (SSL) [3] is introduced to address this problem, which uses a large number of unlabeled data along with a few labeled data to train the model.

Support vector machine (SVM) is an efficient and effective tool for the tasks of classification and regression [4]. After the introduction of SVM, it outperformed a large number of other tools in a wide of applications [5,6]. Recently, Marchetti and Perracchione [7] proposed local-to-global support vector machine (LGSVM), which is based on the approximation theory and local kernel-based models. LGSVM is a global method which is constructed by gluing local SVM contributions by support weights. Meanwhile, LGSVM improves the execution time and does not lose the overall classification capability. However, the computational complexity of the quadratic programming problem (QPP) is enormous. It will take a long training time when the datasets are large. In order to optimize SVM, in recent years, various algorithms have been proposed. Mangasarian and Wild [8] proposed generalized eigenvalue proximal support vector machine (GEPSVM), which relaxes the bounding or proximal planes to be nonparallel in the input space. Li et al. [9] proposed L1-norm proximal nonparallel support vector machine (L1-NPSVM), which uses a justifiable iterative technique to solve a pair of L1-norm optimal problems and has no parameters to be regularized. In the real world, the same object can be described from multiple views. SVM can also be combined with multi-view learning. For example, Sun et al. [10] proposed multi-view GEPSVM, which can effectively combine two views by the introduction of multi-view co-regularization term and the consensus on distinct views is maximized. However, multi-view GEPSVM ignores discriminations between different views and agreement of the same view. Therefore, Cheng et al. [11] proposed improved multi-view GEPSVM via inter-view difference maximization and intra-view agreement minimization, and they also designed an ef-

* Corresponding author.
E-mail addresses: xjxie11@gmail.com (X. Xie), feixiangsun2022@163.com (F. Sun), qianjiangbo@nbu.edu.cn (J. Qian), guolijun@nbu.edu.cn (L. Guo), zhangrong@nbu.edu.cn (R. Zhang), yexlwh@163.com (X. Ye), zhijinecnu@gmail.com (Z. Wang).

https://doi.org/10.1016/j.patcog.2022.109192
0031-3203/

fective iterative algorithm and proved its convergence by the theory and experiment.

Jayadeva et al. [12] proposed twin support vector machine (TSVM) which seeks a pair of nonparallel hyperplanes and assigns the label of new data depending on which of the two hyperplanes it is closer to, and it is four times faster than the standard SVM. There are also lots of variants for TSVM, for example, Kumar and Gopal [13] presented smooth twin support vector machine (STSVM), which uses smoothing techniques for TSVM, and converts QPP into unconstrained minimization problem and can be solved by the well-known Newton-Armijo algorithm. Additionally, they also presented least squares twin support vector machine (LSTSVM) [14] in which two modified primal problems are solved instead of two dual problems, and the computational time is lesser than the one of standard TSVM. Nevertheless, TSVM is not efficient enough to solve large-scale data problems and it obtains the optimal hyperparameters by the grid search method which is time-consuming. Therefore, Pan et al. [15] presented a safe screening rule (SSR) for linear TSVM and modified safe screening rule (MSSR) for nonlinear TSVM. Most training data can be deleted by SSR and MSSR and the scale of TSVM is reduced before solving it. Besides, the hyperparameters tuning process can be accelerated by the sequential versions of SSR and MSSR. TSVM can also be combined with multi-view learning, for example, Xie and Sun [16] proposed multi-view twin support vector machine (MvTSVM) which can be solved by a pair of quadratic programming problems (QPPs). Then, they also extended it to semi-supervised learning and proposed multi-view Laplacian twin support vector machine [17] whose dual optimization problems are also QPPs. After that, they also proposed multi-view support vector machines and multi-view twin support vector machines with the consensus and complementarity information [18], which not only can deal with multi-view classification problems, but also combine the consensus and complementarity principles. Chen et al. [19] presented a novel twin support vector machine for multi-label learning which constructs multiple nonparallel hyperplanes to exploit the potential multi-label information embedded in the data.

Manifold regularization (MR) [20] is one of the most efficient strategies for SSL. The MR introduces a regularization term to capture the geometric information embedded in the data for SSL and traditional machine learning model can be combined with MR. For example, Sun and Shawe-Taylor [21] proposed sparse semi-supervised learning using conjugate functions, which uses Fenchel-Legendre conjugates to rewrite a convex insensitive loss including a regularization with unlabeled data. Meanwhile, they also presented a globally optimal iterative algorithm to optimize the problem. Belkin et al. [22] include the intrinsic smoothness penalty term to SVM and extended SVM to semi-supervised field. Qi et al. [23] proposed Laplacian twin support vector machine (Lap-TSVM), which is able to exploit the geometric information of the marginal distribution embedded in the unlabeled data to seek two nonparallel hyperplanes for semi-supervised classification. However, Lap-TSVM needs to solve two QPPs with matrix inversion, which is time-consuming. In order to reduce the computational cost, Chen et al. [24] proposed Laplacian least twin support vector machine (Lap-LSTSVM) for semi-supervised classification, which solves two systems of linear equations and can be efficiently performed with conjugate gradient (CG) algorithm. Besides, they also introduced a meaningful hyperparameter to balance the regularization terms. Liu et al. [25] presented nonparallel support vector machine (NPSVM) with large margin distribution for pattern classification, which firstly reconstructs large margin distribution and then introduces it into NPSVM to process small and medium-scale datasets. Kang et al. [26] proposed structured graph learning framework to preserve the local and global structure. The self-expressiveness of data is used to capture global structure and adaptive neighbor is used to capture local structure. Besides, the rank constraint is considered to achieve graph to have exact c connected components and multiple kernel learning is presented to avoid the extensive search for suitable kernel. The disadvantage of structured graph learning framework is that its time complexity is high. Nie et al. [27] proposed multi-view clustering and semi-supervised classification with adaptive neighbors, which performs clustering/semi-supervised classification and local structure learning simultaneously. There are also lots of applications for semi-supervised learning such as structural health monitoring. For example, Garrett et al. [28] presented a multiresolution classification framework for semi-supervised learning. The multiresolution framework is used to handle nonstationarities in the signals and extract features in each localized time-frequency region. Meanwhile, an adaptive graph filter was also proposed for semi-supervised classification, Bull et al. [29] proposed a semi-supervised Gaussian mixture model for probabilistic damage classification, which improves the classification accuracy significantly. Agrawal and Chakraborty [30] investigated an alternative semi-supervised approach for damage identification. Unlike the graph-based approach, the classification for their proposed method is not based on edge-weight for the data. As a result, two points nearing each other do not necessarily have the same labels.

However, the L2 norm is sensitive to the noises and outliers [31]. Therefore, in this paper, we propose a novel Laplacian Lp norm least squares twin support vector machine termed Lap-LpLSTSVM for semi-supervised classification, and the architecture of our proposed Lap-LpLSTSVM is illustrated in Fig. 1. In contrast to Lap-SVM, Lap-TSVM, and Lap-LSTSVM, the contributions of this paper are as follows:

- An Lp norm graph regularization term based on the technique of eigenvalue decomposition is introduced to exploit the geometric information embedded in the data, which can boost the generalization ability of our model.
- The Lp norm is introduced instead of the L1 norm or L2 norm, which can achieve the desired performance by choosing the appropriate value of p.
- An efficient iterative strategy is designed to solve the optimization problem.
- The appropriate hyperparameters will be obtained by the strategy of cross-validation, experimental results show that our proposed method outperforms other state-of-the-art methods in the linear and nonlinear cases.

The remaining parts of this paper are organized as follows: Section 2 briefly describes the background of Lap-SVM, Lap-TSVM, and Lap-LSTSVM. Section 3 introduces the details of our proposed Lap-LpLSTSVM in the linear and nonlinear cases. Section 4 shows the experiments we make and the analysis for the experiments. In the last of this paper, we give the conclusions.

2. Related work

In this paper, all the matrices are written in uppercase. The vectors and scalars are written in lowercase. For matrix A and vector e, we denote $A^\top$ and $e^\top$ as their transpose, respectively. Meanwhile, A^{-1} denotes the inverse matrix for matrix A.

A binary semi-supervised classification problem is considered in the d dimensional real space $\mathbb{R}^d$. The total dataset is represented as $M = \{(x_1, y_1), \dots (x_l, y_l), \dots, x_{l+u}\}$, Let $X_l = \{x_i\}_{i=1}^{l} \in \mathbb{R}^{l\times d}$ denote the labeled data, $Y_l = \{y_i\}_{i=1}^{l} \in \{-1, 1\}$ denote the labels, $X_u = \{x_i\}_{i=l+1}^{l+u}$ denote the unlabeled data. We denote $A \in \mathbb{R}^{m_1\times d}$ as the labeled data belonging to "+1" class and $B \in \mathbb{R}^{m_2\times d}$ as the labeled data belonging to "−1" class, where $m_1 + m_2 = l$ indicates the total labeled data.

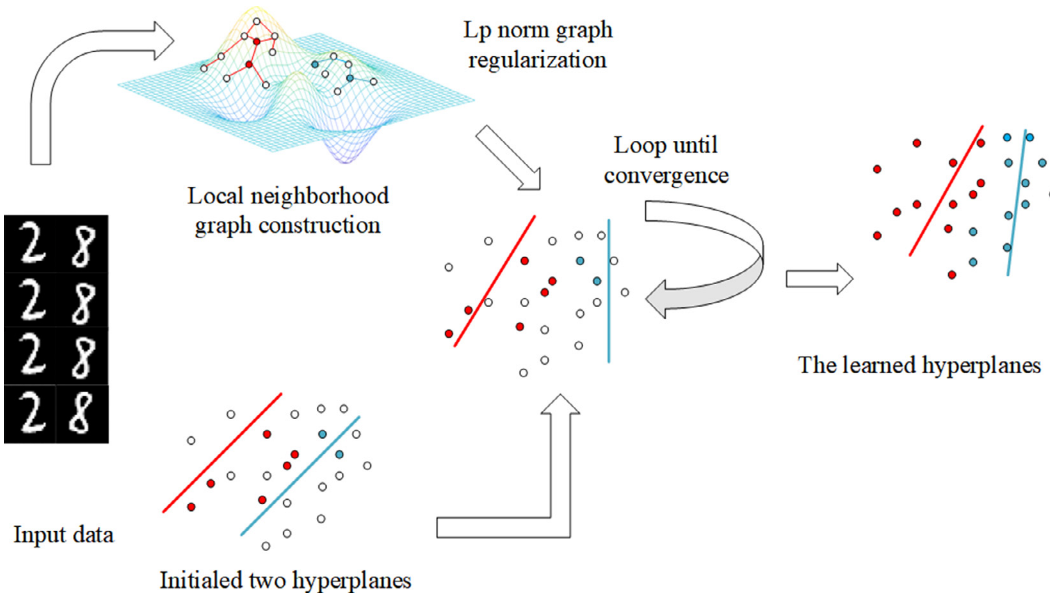


Fig. 1. The framework of our proposed Laplacian Lp norm least squares twin support vector machine (Lap-LpLSTSVM). The red points, blue points and white points represent the positive, negative and unlabeled data respectively. With the input data, local neighborhood graph will be constructed by the k nearest neighbors. Lp norm is used to measure the distance and Lp norm graph regularization term is used to exploit geometric information embedded in the data to obtain the desired performance with the flexible value of p. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

2.1. Lap-SVM

Lap-SVM [32] is a semi-supervised extension of SVM, which exploits the geometry of probability distribution with the regularization term. In one word, if two data x_1, x_2 are close in the intrinsic geometry $P_x(x)$, the condition probability distributions $P_{Y|X}(y|x_1)$ and $P_{Y|X}(y|x_2)$ are similar. The regularized minimization function in Lap-SVM is formulated as

$$f^* = \arg\min_{f\in H_k} \frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + c_1\|f\|_K^2 + c_2\|f\|_I^2, \tag{1}$$

where f is the decision function, c_1 controls the complexity of function f in the reproduction kernel Hilbert space (RKHS) $\mathcal{H}_k$, l is the number of labeled data, V is loss function such as squared loss or hinge loss, c_2 controls the complexity of the function in the intrinsic geometry of marginal distribution. $\|f\|_I^2$ is empirically estimated from the labeled and unlabeled data using the graph Laplacian. We have

$$\begin{aligned}\|f\|_I^2 &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(f(x_i)-f(x_j))^2\\ &= \frac{1}{2}(\sum_{i=1}^{n} d_i f^2(x_i) + \sum_{j=1}^{n} d_j f^2(x_j)\\ &\quad -2\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij} f(x_i) f(x_j))\\ &= \sum_{i=1}^{n} d_i f^2(x_i) - \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij} f(x_i) f(x_j)\\ &= f^\top (D-W) f,\end{aligned} \tag{2}$$

where $L = D - W$ is the Laplacian matrix. D is the diagonal matrix and $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$. W can be obtained by k nearest neighbors.

$$W_{ij} = \begin{cases} \exp\left(-\left\|x_i - x_j\right\|^2/2\delta^2\right) & \text{if } x_i, x_j \text{ are neighbors,} \\ 0 & \text{otherwise.}\end{cases} \tag{3}$$

$\|f\|_K^2$ is a regularization term of functions concerning RKHS. Therefore, Lap-SVM is denoted as

$$\min_{f\in\mathcal{H}_k} \sum_{i=1}^{l} \max(1 - y_i f(x_i), 0) + c_1\alpha^\top K\alpha + c_2 f^\top L f, \tag{4}$$

where α is the coefficient over finite dimensional space, K is the kernel matrix with $K(x_i, x_j)$ as its (i, j)th element. By introducing the slack variable ξ_i, the unconstrained problem can be written as a constrained one,

$$\begin{aligned}&\min_{\alpha\in R^n, \xi\in R^l} \sum_{i=1}^{l} \xi_i + c_1\alpha^\top K\alpha + c_2\alpha^\top KLK\alpha\\ &\text{ s.t. } \quad y_i(\sum_{j=1}^{n} \alpha_i k(x_i, x_j) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, l,\\ &\xi_i \geq 0, \quad i = 1, \ldots, l.\end{aligned} \tag{5}$$

Then the dual problem for Eq. (5) is shown as

$$\begin{aligned}&\max_{\beta\in R^l} \sum_{i=1}^{l} \beta_i - \frac{1}{2}\beta^\top Q\beta\\ &\text{ s.t. } \quad \sum_{i=1}^{l} \beta_i y_i = 0,\\ &0 \leq \beta_i \leq 1, \quad i = 1, \ldots, l,\end{aligned} \tag{6}$$

where $Q = YJ_{\mathcal{L}}K(2c_1 I + 2c_2 KL)^{-1}J_{\mathcal{L}}^\top Y$, $J_{\mathcal{L}} \in \mathbb{R}^{l\times n}$ is the matrix $[I, O]$ where $I \in \mathbb{R}^{l\times l}$ is the identity matrix and $O \in \mathbb{R}^{l\times u}$ is a rectangular matrix with all zeros. $Y \in \mathbb{R}^{l\times l}$ is a diagonal matrix composed by labels $y_i, i = 1, \ldots, l$. The Preconditioned Conjugate Gradient (PCG) algorithm [32] can be employed to calculate the parameters of hyperplanes. Then the labels of the unlabeled data can be calculated by

$$y_j = \text{sign}(\sum_{i=1}^{l+u} \alpha_i^* K(x_i, x_j))(j = l+1, \ldots, l+u). \tag{7}$$

2.2. Lap-TSVM

Lap-TSVM extends TSVM to the semi-supervised case. The goal of Lap-TSVM is to exploit the underlying geometric information embedded in the data to generate a better classifier. In order to achieve this purpose, in the linear case of Lap-TSVM, it aims at seeking the following two nonparallel hyperplanes

$$f_1(x) = w_1^\top x + b_1 = 0, \tag{8}$$

$$f_2(x) = w_2^\top x + b_2 = 0, \tag{9}$$

where $w_1, w_2 \in \mathbb{R}^d$ are normal vectors and $b_1, b_2 \in \mathbb{R}$ are bias. The empirical risk of Lap-TSVM is implemented by the following L2 and L1 norm loss functions

$$R_1(f) = \sum_{i=1}^{m_1}(f_1(A_i))^2 + c_1\sum_{i=1}^{m_2}\max(0, 1 + f_1(B_i)), \tag{10}$$

$$R_2(f) = \sum_{i=1}^{m_2}(f_2(B_i))^2 + c_1\sum_{i=1}^{m_1}\max(0, 1 - f_2(A_i)), \tag{11}$$

where $c_1 > 0$ is the penalty parameter to trade off the loss terms in Eqs. (10) and (11). Consider the RKHS regularization terms

$$\|f_1\|_{\mathcal{H}}^2 = \tfrac{1}{2}(\|w_1\|^2 + b_1^2), \tag{12}$$

$$\|f_2\|_{\mathcal{H}}^2 = \tfrac{1}{2}(\|w_2\|^2 + b_2^2), \tag{13}$$

and manifold regularization terms

$$\|f_1\|_{\mathcal{M}}^2 = \tfrac{1}{2}(Mw_1 + eb_1)^\top L(Mw_1 + eb_1), \tag{14}$$

$$\|f_2\|_{\mathcal{M}}^2 = \tfrac{1}{2}(Mw_2 + eb_2)^\top L(Mw_2 + eb_2), \tag{15}$$

therefore, the formulation for Lap-TSVM can be expressed as

$$\begin{aligned} &\min_{w_1,b_1,\xi}\ \tfrac{1}{2}\|Aw_1 + e_1b_1\|^2 + c_1e_2^\top\xi + c_2(\|w_1\|^2 + b_1^2) \\ &+c_3(Mw_1 + eb_1)^\top L(Mw_1 + eb_1) \\ &\ \text{s.t.}\quad -(Bw_1 + e_2b_1) + \xi \geq e_2, \quad \xi \geq 0, \end{aligned} \tag{16}$$

and

$$\begin{aligned} &\min_{w_2,b_2,\eta}\ \tfrac{1}{2}\|Bw_2 + e_2b_2\|^2 + c_1e_1^\top\eta + c_2(\|w_2\|^2 + b_2^2) \\ &+c_3(Mw_2 + eb_2)^\top L(Mw_2 + eb_2) \\ &\ \text{s.t.}\quad Aw_2 + e_1b_2 + \eta \geq e_1, \quad \eta \geq 0, \end{aligned} \tag{17}$$

where $c_1, c_2, c_3 > 0$ are the hyperparameters of regularization term, $\xi \in \mathbb{R}^{m_2}$, $\eta \in \mathbb{R}^{m_1}$ are the slack vectors and $\|\cdot\|$ denotes the L2 norm. e_1, e_2 and e are the vectors of ones with appropriate dimensions. $A \in \mathbb{R}^{m_1\times d}$ and $B \in \mathbb{R}^{m_2\times d}$ are the positive and negative training data, respectively. The dual problems of QPPs for Eqs. (16) and (17) can be formulated as

$$\begin{aligned} &\max_\alpha\ e_2^\top\alpha - \tfrac{1}{2}\alpha^\top G(H^\top H + c_2I + c_3J^\top LJ)^{-1}G^\top\alpha \\ &\ \text{s.t.}\quad 0 \leq \alpha \leq c_1e_2, \end{aligned} \tag{18}$$

$$\begin{aligned} &\max_\beta\ e_1^\top\beta - \tfrac{1}{2}\beta^\top H(G^\top G + c_2I + c_3J^\top LJ)^{-1}H^\top\beta \\ &\ \text{s.t.}\quad 0 \leq \beta \leq c_1e_1, \end{aligned} \tag{19}$$

where $H = [A\,e_1]$, $G = [B\,e_2]$ and $J = [M\,e]$. Once the Eqs. (18) and (19) are solved, the two nonparallel hyperplanes can be obtained by

$$\begin{aligned} v_1 &= -(H^\top H + c_2I + c_3J^\top LJ)^{-1}G^\top\alpha, \\ v_2 &= (G^\top G + c_2I + c_3J^\top LJ)^{-1}H^\top\beta, \end{aligned} \tag{20}$$

where $v_k = [w_k^\top\ b_k]^\top (k = 1, 2)$. Each hyperplane is closer to one class and as far as possible from the other class. When two nonparallel hyperplanes are obtained, a new data $x \in \mathbb{R}^d$ can be assigned to the corresponding class " + 1" or " - 1" depending on which of the two hyperplanes it is closer to, i.e.

$$f(x) = \text{sign}\left(\frac{w_1^\top x + b_1}{\|w_1\|} - \frac{w_2^\top x + b_2}{\|w_2\|}\right). \tag{21}$$

2.3. Lap-LSTSVM

In this section, we introduce the formulation of Laplacian least squares twin support vector machine. First of all, the empirical risk for Lap-LSTSVM is written as

$$R_1(f) = \sum_{i=1}^{m_1}(f_1(A_i))^2 + c_1\sum_{i=1}^{m_2}(f_1(B_i) + 1)^2, \tag{22}$$

$$R_2(f) = \sum_{i=1}^{m_2}(f_2(B_i))^2 + c_1\sum_{i=1}^{m_1}(f_2(A_i) - 1)^2, \tag{23}$$

the formulation of Lap-LSTSVM can be expressed as

$$\begin{aligned} \min_{w_1,b_1}\mathcal{L}_1 = &\ \frac{1}{2}\|Aw_1 + e_1b_1\|^2 + c_1\|Bw_1 + e_2b_1 + e_2\|^2 \\ &+c_2(\|w_1\|^2 + b_1^2) + c_3(Mw_1 + eb_1)^\top L(Mw_1 + eb_1), \end{aligned} \tag{24}$$

and

$$\begin{aligned} \min_{w_2,b_2}\mathcal{L}_2 = &\ \frac{1}{2}\|Bw_2 + e_2b_2\|^2 + c_1\|Aw_2 + e_1b_2 - e_1\|^2 \\ &+c_2(\|w_2\|^2 + b_2^2) + c_3(Mw_2 + eb_2)^\top L(Mw_2 + eb_2), \end{aligned} \tag{25}$$

where $c_1, c_2, c_3 > 0$ are hyperparameters. $A \in \mathbb{R}^{m_1\times d}$ and $B \in \mathbb{R}^{m_2\times d}$ are positive and negative training data, respectively. $M \in \mathbb{R}^{(l+u)\times d}$ is all the training data. e_1, e_2 and e are the vectors of ones with appropriate dimensions. Take the partial derivatives of Eq. (24) with respect to w_1 and b_1 and set them to zero, the following equations can be obtained

$$\begin{aligned} \nabla_{w_1}\mathcal{L}_1 = &\ A^\top(Aw_1 + e_1b_1) + c_1B^\top(Bw_1 + e_2b_1 + e_2) + c_2w_1 \\ &+c_3M^\top L(Mw_1 + eb_1) = 0, \end{aligned} \tag{26}$$

and

$$\begin{aligned} \nabla_{b_1}\mathcal{L}_1 = &\ e_1^\top(Aw_1 + e_1b_1) + c_1B^\top(Bw_1 + e_2b_1 + e_2) + c_2b_1 \\ &+c_3e^\top L(Mw_1 + eb_1) = 0. \end{aligned} \tag{27}$$

The joint matrix form of Eqs (26) and (27) is formulated as

$$H^\top Hv_1 + c_1G^\top Gv_1 + c_1G^\top e_2 + c_2v_1 + c_3J^\top LJv_1 = 0, \tag{28}$$

where $H = [A\,e_1], G = [B\,e_2], J = [M\,e], v_1 = [w_1^\top\ b_1]^\top$. The solution of Eq. (28) can be shown as

$$Pv_1 = -c_1G^\top e_2, \tag{29}$$

where $P = H^\top H + c_1G^\top G + c_2I + c_3J^\top LJ$. Similarly, the solution of Eq. (25) can be obtained by

$$Qv_2 = c_1H^\top e_1, \tag{30}$$

where $v_2 = [w_2^\top\ b_2]^\top$ and $Q = G^\top G + c_1H^\top H + c_2I + c_3J^\top LJ$. A powerful conjugate gradient algorithm is employed to solve Eqs. (29) and (30).

Once the solutions (w_1, b_1) and (w_2, b_2) are obtained. A new data $x \in \mathbb{R}^d$ is assigned to the corresponding class depending on which of two hyperplanes it is closer to, i.e.,

$$\text{class}\ i = \arg\min_{k=1,2}\frac{|w_k^\top x + b_k|}{\|w_k\|}. \tag{31}$$

3. Proposed method

3.1. Linear Laplacian Lp norm least squares twin support vector machine

In this section, we elaborate on the formulation of our proposed Laplacian Lp norm least squares twin support vector machine. Since Laplacian matrix L is real symmetric, it can be decomposed into the following formulation by the technique of eigenvalue decomposition

$$L = UVU^\top, \tag{32}$$

where U is a matrix composed of eigenvectors and V is a diagonal matrix whose diagonal elements are eigenvalues corresponding to the eigenvectors. Then the graph regularization term can be written as

$$(JZ_k)^\top UVU^\top(JZ_k) = \|PZ_k\|^2, \tag{33}$$

where $Z_k = [w_k^\top \, b_k]^\top (k = 1, 2)$, $J = [M \; e]$, $M \in \mathbb{R}^{(l+u)\times d}$ includes all the labeled and unlabeled data, $P = V^{\frac{1}{2}} U^\top J$. To promote the performance, Lp norm is used instead of L2 norm. The proposed Lp norm graph regularization term can be formulated as

$$\mathcal{G}(Z_k) = \|PZ_k\|_p^p. \tag{34}$$

Then, the proposed Laplacian Lp norm least squares twin support vector machine can be formulated as

$$\min_{w_1,b_1} \mathcal{L}_1 = \frac{1}{2}\|Aw_1 + e_1 b_1\|_p^p + c_1\|w_1\|_p^p + c_1|b_1|^p + c_2\|Bw_1 + e_2 b_1 + e_2\|_p^p + c_3\|PZ_1\|_p^p, \tag{35}$$

$$\min_{w_2,b_2} \mathcal{L}_2 = \frac{1}{2}\|Bw_2 + e_2 b_2\|_p^p + c_1\|w_2\|_p^p + c_1|b_2|^p + c_2\| -Aw_2 - e_1 b_2 + e_1\|_p^p + c_3\|PZ_2\|_p^p, \tag{36}$$

where $c_1, c_2, c_3 > 0$ are regularization hyperparameters, $A \in \mathbb{R}^{m_1 \times d}$, $B \in \mathbb{R}^{m_2 \times d}$ are the positive and negative data, respectively. e_1, e_2 and e are vectors of ones with appropriate dimensions. We set

$$H = [A \; e_1], G = [B \; e_2], \tag{37}$$

then Eqs. (35) and (36) can be rewritten as

$$\min_{Z_1} \mathcal{L}_1 = \tfrac{1}{2}\|HZ_1\|_p^p + c_1\|Z_1\|_p^p + c_2\|GZ_1 + e_2\|_p^p + c_3\|PZ_1\|_p^p, \tag{38}$$

$$\min_{Z_2} \mathcal{L}_2 = \tfrac{1}{2}\|GZ_2\|_p^p + c_1\|Z_2\|_p^p + c_2\| -HZ_2 + e_1\|_p^p + c_3\|PZ_2\|_p^p, \tag{39}$$

an efficient iterative algorithm is designed to solve the optimization problem. After introducing the diagonal matrix, Eq. (38) can be rewritten as

$$\min_{Z_1} \mathcal{L}_1 = \frac{1}{2} Z_1^\top H^\top D_1 H Z_1 + c_1 Z_1^\top D_2 Z_1 + c_2 (GZ_1 + e_2)^\top D_3 (GZ_1 + e_2) + c_3 Z_1^\top P^\top D_4 P Z_1, \tag{40}$$

where

$$\begin{aligned}
D_1 &= \mathrm{diag}\Big(\tfrac{1}{|H_1 Z_1|^{2-p}}, \tfrac{1}{|H_2 Z_1|^{2-p}}, \cdots, \tfrac{1}{|H_{m_1} Z_1|^{2-p}}\Big),\\
D_2 &= \mathrm{diag}\Big(\tfrac{1}{|Z_{11}|^{2-p}}, \tfrac{1}{|Z_{12}|^{2-p}}, \cdots, \tfrac{1}{|Z_{1(d+1)}|^{2-p}}\Big),\\
D_3 &= \mathrm{diag}\Big(\tfrac{1}{|G_1 Z_1 + 1|^{2-p}}, \tfrac{1}{|G_2 Z_1 + 1|^{2-p}}, \cdots, \tfrac{1}{|G_{m_2} Z_1 + 1|^{2-p}}\Big),\\
D_4 &= \mathrm{diag}\Big(\tfrac{1}{|P_1 Z_1|^{2-p}}, \tfrac{1}{|P_2 Z_1|^{2-p}}, \cdots, \tfrac{1}{|P_m Z_1|^{2-p}}\Big),
\end{aligned}$$

H_i, G_i denote the ith row data of matrix H and G, respectively. P_i represents the ith row data of matrix P. Z_{1i} denotes the ith element of Z_1. m_1, m_2 are the number of positive and negative data, respectively. m is the number of total training data. By taking the derivative of Eq. (40) with respect to Z_1 and setting it to zero, we can obtain

$$\nabla_{Z_1} \mathcal{L}_1 = H^\top D_1 H Z_1 + c_1 D_2 Z_1 + c_2 G^\top D_3 (GZ_1 + e_2) + c_3 Z_1^\top P^\top D_4 P Z_1 = 0, \tag{41}$$

then we can obtain

$$Z_1 = -c_2 (H^\top D_1 H + c_1 D_2 + c_2 G^\top D_3 G + c_3 P^\top D_4 P)^{-1} \cdot G^\top D_3 e_2, \tag{42}$$

similarly, for Z_2, we can obtain

$$Z_2 = c_2 (G^\top E_1 \; G + c_1 E_2 + c_2 H^\top E_3 H + c_3 P^\top E_4 P)^{-1} \cdot H^\top E_3 e_1, \tag{43}$$

where

$$\begin{aligned}
E_1 &= \mathrm{diag}\Big(\tfrac{1}{|G_1 Z_2|^{2-p}}, \tfrac{1}{|G_2 Z_2|^{2-p}}, \cdots, \tfrac{1}{|G_{m_2} Z_2|^{2-p}}\Big),\\
E_2 &= \mathrm{diag}\Big(\tfrac{1}{|Z_{21}|^{2-p}}, \tfrac{1}{|Z_{22}|^{2-p}}, \cdots, \tfrac{1}{|Z_{2(d+1)}|^{2-p}}\Big),\\
E_3 &= \mathrm{diag}\Big(\tfrac{1}{|H_1 Z_2 + 1|^{2-p}}, \tfrac{1}{|H_2 Z_2 + 1|^{2-p}}, \cdots, \tfrac{1}{|H_{m_1} Z_2 + 1|^{2-p}}\Big),\\
E_4 &= \mathrm{diag}\Big(\tfrac{1}{|P_1 Z_2|^{2-p}}, \tfrac{1}{|P_2 Z_2|^{2-p}}, \cdots, \tfrac{1}{|P_m Z_2|^{2-p}}\Big).
\end{aligned}$$

The algorithm for linear Lap-LpLSTSVM is formulated in Algorithm 1. After we obtain the two nonparallel hyperplanes, Eq. (21) can be used to obtain the label of data.

Algorithm 1 Linear Laplacian Lp norm least squares twin support vector machine.

Input:the data matrix M, the weight parameter δ_1 for k nearest neighbor,the parameters $c_1, c_2, c_3$of Lap-LpLSTSVM (chosen by five-fold cross-validation),convergence constant ϵ and maximum iterations $k_{\max}$.
Initial: initialize Z_1 and Z_2 with the vectors of ones.
Obtain the adjacency matrix by Eq. (3), and then the graph Laplacian can be obtained by $L = D - W$.
Repeat
    update Z_1 by the Eq. (42).
    update Z_2 by the Eq. (43).
    $t = t + 1$.
Until $t == k_{\max}$ or $(\max(\mathrm{abs}(Z_1^t - Z_1^{t+1})) \le \epsilon$ and $\max(\mathrm{abs}(Z_2^t - Z_2^{t+1})) \le \epsilon)$.
Obtain the final label of test data by Eq. (21).

3.2. Nonlinear Laplacian Lp norm least squares twin support vector machine

In this section, we extend linear Lap-LpLSTSVM to the nonlinear case. The decision function can be written as $f_\pm(x) = w_\pm \cdot \phi(x) + b_\pm$, where $\phi(\cdot)$ is the nonlinear mapping from the low dimensional space to higher dimensional Hilbert space $\mathcal{H}$. The two kernel-generated nonparallel hyperplanes are formulated as

$$\begin{aligned}
f_+(x) &= K(x^\top, M^\top) u_1 + b_1 = 0,\\
f_-(x) &= K(x^\top, M^\top) u_2 + b_2 = 0,
\end{aligned} \tag{44}$$

where $M \in \mathbb{R}^{(l+u)\times d}$ represents total labeled and unlabeled data, and $K(\cdot,\cdot)$ is a chosen kernel function such as RBF kernel. The optimization problems for nonlinear Lap-LpLSTSVM can be expressed as

$$\min_{u_1,b_1} \mathcal{L}_1 = \frac{1}{2}\|K(A, M^\top)u_1 + e_1 b_1\|_p^p + c_1\|u_1\|_p^p + c_1|b_1|^p + c_2\|K(B, M^\top)u_1 + e_2 b_1 + e_2\|_p^p + c_3\|P\rho_1\|_p^p, \tag{45}$$

$$\min_{u_2,b_2} \mathcal{L}_2 = \frac{1}{2}\|K(B, M^\top)u_2 + e_2 b_2\|_p^p + c_1\|u_2\|_p^p + c_1|b_2|^p + c_2\| -K(A, M^\top)u_2 - e_1 b_2 + e_1\|_p^p + c_3\|P\rho_2\|_p^p, \tag{46}$$

where $\rho_k = [u_k^\top \, b_k]^\top (k = 1, 2)$, $P = V^{\frac{1}{2}} U^\top [K(M, M^\top) \; e]$, u_1 and u_2 are relevant coefficients. We set

$$H_\phi = [K(A, M^\top) \; e_1], G_\phi = [K(B, M^\top) \; e_2]. \tag{47}$$

Similarly to the linear Lap-LpLSTSVM, we can obtain

$$\rho_1 = -c_2 (H_\phi^\top D_{\phi 1} H_\phi + c_1 D_{\phi 2} + c_2 G_\phi^\top D_{\phi 3} G_\phi + c_3 P_\phi^\top D_{\phi 4} P_\phi)^{-1} \cdot G_\phi^\top D_{\phi 3} e_2, \tag{48}$$

where

$$\begin{aligned}
D_{\phi 1} &= \mathrm{diag}\Big(\tfrac{1}{|H_{\phi 1}\rho_1|^{2-p}}, \tfrac{1}{|H_{\phi 2}\rho_1|^{2-p}}, \cdots, \tfrac{1}{|H_{\phi m_1}\rho_1|^{2-p}}\Big),\\
D_{\phi 2} &= \mathrm{diag}\Big(\tfrac{1}{|\rho_{11}|^{2-p}}, \tfrac{1}{|\rho_{12}|^{2-p}}, \tfrac{1}{|\rho_{13}|^{2-p}}, \cdots, \tfrac{1}{|\rho_{1(m+1)}|^{2-p}}\Big),\\
D_{\phi 3} &= \mathrm{diag}\Big(\tfrac{1}{|G_{\phi 1}\rho_1 + 1|^{2-p}}, \tfrac{1}{G_{\phi 2}\rho_1 + 1|^{2-p}}, \cdots, \tfrac{1}{|G_{\phi m_2}\rho_1 + 1|^{2-p}}\Big),\\
D_{\phi 4} &= \mathrm{diag}\Big(\tfrac{1}{|P_{\phi 1}\rho_1|^{2-p}}, \tfrac{1}{|P_{\phi 2}\rho_1|^{2-p}}, \cdots, \tfrac{1}{|P_{\phi m}\rho_1|^{2-p}}\Big),
\end{aligned}$$

$G_{\phi i}$, $H_{\phi i}$ denote the ith row data of matrix G_ϕ and H_ϕ, respectively. $P_{\phi i}$ denotes the ith row data of matrix P. ρ_{1i} denotes the ith element of vector ρ_1. m_1 and m_2 are the number of positive and negative data, respectively. m is the number of total training data. Similarly to ρ_1, we can obtain

$$\rho_2 = c_2(G_\phi^\top E_{\phi 1} G_\phi + c_1 E_{\phi 2} + c_2 H_\phi^\top E_{\phi 3} H_\phi + c_3 P_\phi^\top E_{\phi 4} P_\phi)^{-1} \cdot H_\phi^\top E_{\phi 3} e_1, \tag{49}$$

where

$$\begin{aligned} E_{\phi 1} &= \mathrm{diag}\left(\frac{1}{|G_{\phi 1}\rho_2|^{2-p}}, \frac{1}{|G_{\phi 2}\rho_2|^{2-p}}, \cdots, \frac{1}{|G_{\phi m_2}\rho_2|^{2-p}}\right), \\ E_{\phi 2} &= \mathrm{diag}\left(\frac{1}{|\rho_{21}|^{2-p}}, \frac{1}{|\rho_{22}|^{2-p}}, \cdots, \frac{1}{|\rho_{2(m+1)}|^{2-p}}\right), \\ E_{\phi 3} &= \mathrm{diag}\left(\frac{1}{|H_{\phi 1}\rho_2+1|^{2-p}}, \frac{1}{|H_{\phi 2}\rho_2+1|^{2-p}}, \cdots, \frac{1}{|H_{\phi m_1}\rho_2+1|^{2-p}}\right), \\ E_{\phi 4} &= \mathrm{diag}\left(\frac{1}{|P_{\phi 1}\rho_2|^{2-p}}, \frac{1}{|P_{\phi 2}\rho_2|^{2-p}}, \cdots, \frac{1}{|P_{\phi m}\rho_2|^{2-p}}\right). \end{aligned}$$

ρ_1 and ρ_2 can be updated by Eqs. (48) and(49). A new data $x \in \mathbb{R}^d$ is assigned to the corresponding class depending on which of two hyperplanes it is closer to, i.e.,

$$\text{class } i = \arg\min_{k=1,2} \frac{|K(x^\top, M^\top)u_k + b_k|}{\sqrt{u_k^\top K(M, M^\top)u_k}}. \tag{50}$$

The algorithm for nonlinear Lap-LpLSTSVM is formulated in the Algorithm 2.

Algorithm 2 Nonlinear Laplacian Lp norm least squares twin support vector machine.

Input:the data matrix M, the weight parameter δ_1 for k nearest neighbor,the parameters c_1, c_2, c_3 and the RBF kernel parameter $\delta_2$of nonlinear Lap-LpLSTSVM (chosen by five-fold cross-validation),convergence constant ϵ and maximum iterations $k_{\max}$.
Initial: initialize ρ_1 and ρ_2 with thevectors of ones.
Obtain the adjacency matrix by Eq. (3), and then the graph Laplacian can be obtained by $L = D - W$.
Repeat
    update ρ_1 by the Eq. (48).
    update ρ_2 by the Eq. (49).
    $t = t + 1$.
Until $t == k_{\max}$ or $(\max(\mathrm{abs}(\rho_1^t - \rho_1^{t+1})) \le \epsilon$ and $\max(\mathrm{abs}(\rho_2^t - \rho_2^{t+1})) \le \epsilon)$.
Obtain the label of test data by Eq. (50).

3.3. Time complexity

In this section, we analyze the time complexity of our proposed Lap-LpLSTSVM. N and d are set as the number of total training data and the dimension of each data, respectively. T is the maximal number of iterations. The time complexity of constructing adjacency matrix and eigenvalue decomposition are both $O(N^2)$. Meanwhile, in the linear case, we need to find the inverse matrix of a $(d+1) \times (d+1)$ matrix whose time complexity is $O((d+1)^3)$. As a whole, the time complexity of our proposed Lap-LpLSTSVM in the linear case is $O(2T(d+1)^3)$. As for its kernel version, the inverse matrix of a $N \times N$ matrix whose time complexity is $O(N^3)$ needs to be obtained. Therefore, for the nonlinear Lap-LpLSTSVM, the time complexity is $O(2TN^3)$.

4. Experiment

In this section, to demonstrate the effectiveness of our proposed algorithm, we evaluate our algorithm on the task of binary classification in synthetic and real-world datasets. The synthetic datasets we use in the experiment are two moons and two lines. UCI and Handwritten Numeral are real-world datasets. Our experiments are run in MATLAB 2016b on an Intel Core i7-9700K with 32GB RAM. We compare our proposed method with Lap-SVM, Lap-TSVM, Lap-LSTSVM, and manifold proximal support vector machine (MPSVM) [33]. As known, least squares twin support vector machine [14] is a classical machine learning algorithm. To show that our proposed method can make full use of unlabeled data to exploit the geometric information embedded in the data, we adopt LSTSVM which uses the same labeled data to train the classifier for comparison.

- Lap-SVM: Laplacian SVM. It uses the hinge loss with the manifold assumption to construct a hyperplane classifier.
- Lap-TSVM: Laplacian twin SVM. It adopts manifold assumptions and exploits the geometric information embedded in data to construct two nonparallel hyperplanes to classify the data.
- Lap-LSTSVM: Laplacian least squares twin SVM. It introduces a meaningful regularization hyperparameter to balance the RKHS term and manifold regularization term. With the CG algorithm, two systems of linear equations can be solved.
- MPSVM: Manifold proximal SVM. It includes discriminant and underlying geometric information to construct a more reasonable classifier. The optimal nonparallel proximal hyperplanes can be obtained by solving a pair of standard eigenvalue decomposition problems.
- LSTSVM: Least squares twin SVM. It is a classical machine learning algorithm, which requires the solution of two systems of linear equations.

4.1. Experiment setting

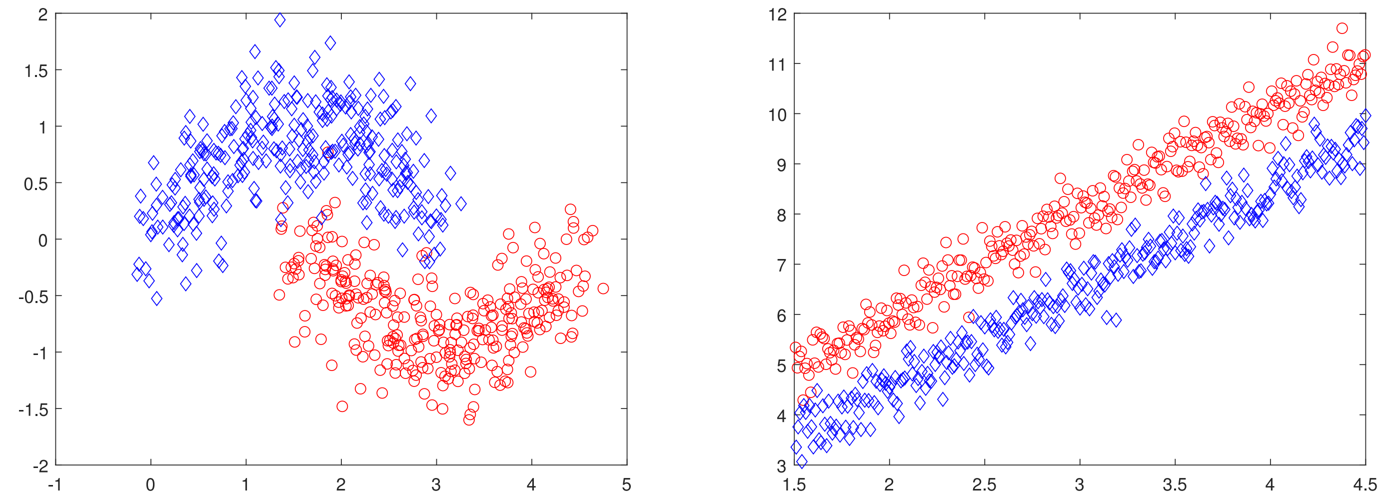
In this section, we demonstrate the detailed setting of our experiments. To partition datasets, the percentages of labeled and total training data are set to 8% and 60%, respectively. The percentage of test data is set to 40%. The maximum number of iterations is set to 20 and the threshold ϵ is set to 10^{-4}. The p value of Lp norm is selected from the set $\{1, 1.5, 2, 3, 5\}$. All the data are located in [0,1] before the training, and the best performance is highlighted.

As known, the performance of our proposed algorithm and other compared algorithms depends on the choice of hyperparameters [34] and the cross-validation method is employed to find the optimal hyperparameters. In our experiments, the five-fold cross-validation is used to select the optimal hyperparameters. In the procedure of selecting hyperparameters, both labeled and unlabeled data are divided into five groups, and four groups of labeled and unlabeled data are used to train the model. After the model is trained, the rest one group of the labeled data is used to be the validation set to obtain the accuracy. The process is repeated for five times. In the linear case, for the sake of brevity, c_1, c_2 are chosen from the set $\{2^i | i = -5, -4, \dots, 4, 5\}$ and c_3 is equal to c_2. In the nonlinear case, we set $c_1 = c_2 = c_3$, c_1, c_2, c_3 and RBF kernel parameter δ_2 are also selected from the set $\{2^i | i = -5, -4, \dots, 4, 5\}$. After the optimal hyperparameters are obtained by the strategy of cross-validation, they can be used to train the model. This process is repeated for five times, and the average accuracies and time are reported.

4.2. Experiments on synthetic datasets

In this section, we use two synthetic datasets containing two moons and two lines to make experiments. These two synthetic datasets have two features and can be divided into two categories. In order to further validate the effectiveness of the proposed method, we also add Gaussian noise with the mean value of zero and standard deviation of 0.1, 0.2, and 0.3 to the original data. The distributions of two moons and two lines which are added to the noise of mean value of zero and standard deviation of 0.2 are shown in Fig. 2. We compare our algorithm with the other state-of-the-art algorithms in linear and nonlinear cases. The results of linear and nonlinear cases for synthetic datasets are shown in Tables 1 and 2, respectively.

In the linear case, we can see that our proposed method achieves the optimal performance in five datasets except for the



(a) Two balls (b) Two Lines

Fig. 2. Distributions of two moons and two lines datasets under the noise of mean value of zero and standard deviation of 0.2.

Table 1
The testing accuracy and training time on synthetic datasets(linear case).

Datasets	Lap-SVM Acc±Std(%) Time(s)	Lap-TSVM Acc±Std(%) Time(s)	Lap-LSTSVM Acc±Std(%) Time(s)	MPSVM Acc±Std(%) Time(s)	LSTSVM Acc±Std(%) Time(s)	Lap-LpLSTSVM Acc±Std(%) Time(s)
Two moons(0.1)	85.83±10.79 0.0371	91.42±4.55 0.0205	94.25±1.85 0.0098	94.00±4.31 0.0217	93.50±4.40 0.0004	**95.00±1.06** 0.0708
Two moons(0.2)	76.25±18.45 0.0611	91.50±3.32 0.0232	93.75±0.98 0.0153	93.08±1.55 0.0144	93.67±4.15 0.0004	**93.83±1.08** 0.0339
Tow moons(0.3)	71.83±19.61 0.0311	88.67±5.45 0.0201	91.33±2.52 0.0103	89.17±4.04 0.0228	91.92±3.18 0.0004	**92.17±2.27** 0.0762
Two lines(0.1)	53.75±7.47 0.0325	97.42±1.26 0.0215	97.58±1.46 0.0097	**99.00±0.63** 0.0124	96.83±1.83 0.0004	**99.00±0.86** 0.0793
Two lines(0.2)	53.75±7.47 0.0415	98.50±0.48 0.0214	**99.33±0.37** 0.0096	99.25±0.1 0.0135	96.17±1.16 0.0004	99.08±0.46 0.0475
Two lines(0.3)	83.08±9.38 0.0579	97.50±0.78 0.0157	97.67±0.63 0.0107	98.00±0.80 0.0116	94.41±3.26 0.0004	**98.08±0.81** 0.0691

dataset two lines(0.2). In the nonlinear case, our method also obtains the optimal performance in four datasets except two moons(0.1) and two moons(0.3) datasets. The results show the effectiveness of our proposed Lap-LpLSTSVM.

4.3. Experiments on UCI datasets

In this section, we perform our proposed method and other compared methods on the real-world datasets from the UCI machine learning repository. The details of the datasets are summarized in Table 3. Tables 4 and 7 list the results of all methods in the linear and nonlinear cases for UCI datasets. In the linear case, our proposed method obtains 1.43%, 6.39%, 7.86%, and 2.76% improvement on datasets Seeds, Vehicle, Ionosphere, and Liver respectively than the second baseline. In the nonlinear case, our proposed method obtains 1.38%, 0.57%, 3.9% improvement in datasets Glass, Ionosphere, and Wilt respectively than the second baseline. These results show the effectiveness of our proposed method.

4.4. Experiments on handwritten numeral datasets

Handwritten Numeral datasets contain the digits from 0 to 9, and each digit contains 200 data. Since Handwritten Numeral are multi-view datasets, we use one of its views to make the experiments. The Fourier coefficients of characters shapes (FOU) are used to evaluate the performance of our proposed Lap-LpLSTSVM and compared methods. Two digits which contain 400 data are randomly selected from the datasets, and we select six numeral pairs for the experiments. Tables 6 and 8 show the results of the linear and nonlinear cases for our and compared methods on Handwritten Numeral datasets.

From Tables 6 and 8, we can see that our method achieves the optimal performance for all of the datasets in the linear case and obtains the optimal performance in five datasets in the nonlinear case. During the process of cross-validation, the accuracy versus c_1 and c_2 for Handwritten Numeral datasets are shown in Figs. 3, 4 and 5. They show the effect of changing values of c_1 and c_2 on accuracy. In the dataset Handwritten Numeral (0 9), we can see that the accuracy is not sensitive to the choice of c_1 and c_2. In dataset Handwritten Numeral (1 4), the optimal accuracy can be obtained when $c_1 \leq 0.0625$. In dataset Handwritten Numeral (1 6), optimal accuracy is obtained when $c_1 \leq 0.125$. In dataset Handwritten Numeral (2 6), the performance is optimal when $c_1 \leq 0.025$ and $c_2 \geq 8$. In Handwritten Numeral (2 9) and Handwritten Numeral (6 9) datasets, the performance is optimal when $c_1 \leq 0.0625$. According to the analysis above and Figs. 3–5, we can obtain the follow-

Table 2
The testing accuracy and training time on synthetic datasets(nonlinear case).

Datasets	Lap-SVM Acc±Std(%) Time(s)	Lap-TSVM Acc±Std(%) Time(s)	Lap-LSTSVM Acc±Std(%) Time(s)	MPSVM Acc±Std(%) Time(s)	LSTSVM Acc±Std(%) Time(s)	Lap-LpLSTSVM Acc±Std(%) Time(s)
Two moons(0.1)	86.75±2.13 0.1809	96.17±7.18 0.1530	92.75±0.96 0.0509	91.08±3.11 1.1421	**99.75±0.23** 0.0012	96.75±1.54 0.1957
Two moons(0.2)	72.08±17.27 0.2571	96.67±2.95 0.1656	71.25±17.85 0.0810	84.50±9.35 0.4541	96.08±3.52 0.0011	**96.75±1.16** 0.3581
Two moons(0.3)	49.75±3.32 0.1183	93.63±2.31 0.1372	58.00±16.07 0.8270	94.13±1.80 1.0076	**94.67±4.54** 0.0012	90.63±1.59 1.0100
Two lines(0.1)	51.00±4.15 0.0616	94.17±3.82 0.0366	52.25±6.11 0.0609	89.00±3.65 0.1690	95.00±3.08 0.0012	**95.17±4.77** 0.0730
Two lines(0.2)	50.92±4.02 0.0792	97.83±1.65 0.0256	52.75±5.92 0.0304	85.67±11.61 0.8450	92.42±2.91 0.0011	**98.33±0.66** 0.3033
Two lines(0.3)	50.75±3.98 0.0408	95.83±4.70 0.0412	52.33±5.26 0.0280	75.75±12.71 0.1683	92.42±4.08 0.0011	**96.00±2.10** 0.0991

Table 3
Description of the UCI datasets.

Datasets	Selected classes	Instances	Attributes
Glass	(1,2)	146	9
Seeds	(1,2)	140	7
Vehicle	(1,2)	429	18
Australian	-	690	14
Housevotes	-	435	16
Haberman	-	306	3
Ionosphere	-	351	34
Liver	-	345	6
Pima	-	768	8
Wilt	-	500	5

ing conclusions: (1) The optimal value of c_1 is selected from the set {0.03125, 0.0625, 0.125}, so we recommend that c_1 takes a smaller value. (2) In order to achieve better performance, c_2 tends to take a larger value.

4.5. Discussion

Tables 1 –2 shows the results of our proposed Lap-LpLSTSVM in the linear and nonlinear cases on synthetic datasets. We can see that our proposed method achieves the optimal performance in five of the six datasets for the linear case and in four of the six datasets for the nonlinear case, which indicates that our method can also deal with noisy datasets and obtain good performance. Tables 4 and 7 show the results of UCI datasets. Our proposed method outperforms other compared methods all except dataset Wilt in the linear case. In the nonlinear case, our method does not achieve the optimal performance just for the datasets Seeds and Australian. Tables 6 and 8 show the results of six selected numeral pairs on Handwritten Numeral datasets. In the linear case, our proposed method outperforms all compared methods in all datasets, which shows the advantage of our proposed algorithm. In the nonlinear case, we can see that our method obtains the optimal performance in five datasets. Meanwhile, our method outperforms LSTSVM which only uses the same labeled data to train on almost all of the datasets. It indicates that our method can take advantage of unlabeled data to exploit the geometric information embedded in the data. In a word, we can see that our proposed method not only can handle noisy synthetic and real-world datasets but also can make full use of unlabeled data.

Analysis of Variance (ANOVA) is a statistical hypothesis test, which can be used to analyze the accuracy results and test the significance differences between several groups of results. In order to evaluate whether the performance improvement observed

Table 4
The testing accuracy and training time on UCI datasets(linear case).

Datasets	Lap-SVM Acc±Std(%) Time(s)	Lap-TSVM Acc±Std(%) Time(s)	Lap-LSTSVM Acc±Std(%) Time(s)	MPSVM Acc±Std(%) Time(s)	LSTSVM Acc±Std(%) Time(s)	Lap-LpLSTSVM Acc±Std(%) Time(s)
Glass	51.72±5.45 0.0087	56.21±7.07 0.0190	58.62±7.31 0.0012	49.29±3.91 0.0013	**61.03±6.96** 0.0003	**61.03±7.57** 0.0090
Seeds	67.86±20.04 0.0067	94.64±3.09 0.0184	94.64±3.99 0.0011	73.57±20.92 0.0013	93.21±7.29 0.0002	**96.07 ± 4.07** 0.0026
Vehicle	49.19±4.08 0.0203	51.51±1.77 0.0154	50.70±5.43 0.0050	52.62±3.08 0.0078	54.77±4.22 0.0005	**61.16±5.72** 0.0362
Australian	85.65±2.27 0.0586	85.72±2.47 0.0275	85.80±2.13 0.0181	63.28±6.54 0.0218	85.86±1.93 0.0007	**85.87±2.23** 0.1095
Housevotes	87.01±5.15 0.0204	**91.26±3.53** 0.0160	**91.26±3.02** 0.0058	91.18±2.33 0.0058	55.17±4.92 0.0004	**91.26±4.01** 0.0469
Haberman	72.79±2.68 0.0140	70.82±5.46 0.0199	72.62±5.84 0.0048	72.13±2.24 0.0037	66.23±13.38 0.0003	**73.61±1.58** 0.0555
Ionosphere	69.57±7.01 0.0371	72.29±9.25 0.0115	77.86±6.59 0.0049	72.14±4.26 0.0066	80.00±4.68 0.0006	**87.86±2.42** 0.0.0940
Liver	54.32±9.36 0.0207	53.48±8.28 0.0115	56.81±6.39 0.0049	56.81±10.61 0.0066	55.51±10.67 0.0003	**59.57±5.53** 0.0278
Pima	67.62±2.75 0.0760	75.18±2.90 0.0338	74.85±3.53 0.0228	57.22±3.90 0.0185	63.97±2.41 0.0006	**76.03±2.72** 0.0517
Wilt	61.80±2.84 0.0479	58.60±5.33 0.0172	56.10±11.21 0.0064	**87.10±1.71** 0.0090	60.7±4.55 0.0004	82.80±3.21 0.0492

Table 5
The *p*-value for the results of UCI datasets (linear case) for ANOVA.

Ours V.S.	Lap-SVM	Lap-TSVM	Lap-LSTSVM	MPSVM	LSTSVM
p-value	0.0048	0.0278	0.0630	0.0102	0.0590

Table 6
The testing accuracy and training time on Handwritten Numeral datasets(linear case).

Selected numeral pairs	Lap-SVM Acc±Std(%) Time(s)	Lap-TSVM Acc±Std(%) Time(s)	Lap-LSTSVM Acc±Std(%) Time(s)	MPSVM Acc±Std(%) Time(s)	LSTSVM Acc±Std(%) Time(s)	Lap-LpLSTSVM Acc±Std(%) Time(s)
0 9	70.25±24.99	99.63±0.56	99.63±0.56	99.75±0.34	99.50±0.28	**99.88±0.28**
	0.1145	0.0699	0.0200	0.0114	0.0025	0.3129
1 4	51.00±3.66	75.75±8.61	77.63±6.75	78.88±14.57	84.88±9.22	**86.00±5.97**
	0.0301	0.0422	0.0193	0.0093	0.0040	0.0484
1 6	52.87±6.40	83.38±4.93	84.13±3.47	87.00±3.26	85.50±5.40	**89.13±4.30**
	0.0379	0.0544	0.0281	0.0088	0.0034	0.6306
2 6	55.50±9.64	91.75±7.41	92.12±5.51	96.12±2.91	91.88±4.10	**97.63±0.81**
	0.0298	0.0443	0.0167	0.0093	0.0471	0.2641
2 9	58.13±12.36	92.75±5.39	89.25±7.17	95.75±2.31	93.38±4.89	**97.37±2.00**
	0.0178	0.0207	0.0178	0.0081	0.0505	0.0902
6 9	49.75±3.32	78.62±2.70	50.13±3.26	51.13±2.44	51.13±4.56	**55.00±3.278**
	0.0186	0.0567	0.0162	0.0082	0.0025	0.045

Table 7
The testing accuracy and training time on UCI datasets(nonlinear case).

Datasets	Lap-SVM Acc±Std(%) Time(s)	Lap-TSVM Acc±Std(%) Time(s)	Lap-LSTSVM Acc±Std(%) Time(s)	MPSVM Acc±Std(%) Time(s)	LSTSVM Acc±Std(%) Time(s)	Lap-LpLSTSVM Acc±Std(%) Time(s)
Glass	51.72±5.45	63.45±5.64	52.07±7.75	51.03±10.74	60.35±6.90	**64.83±11.48**
	0.0073	0.0331	0.0137	0.0070	0.0003	0.0380
Seeds	85.00±2.68	83.41±1.99	85.65±2.27	**94.64±2.82**	93.93±3.25	85.00± 3.35
	0.0943	0.0353	0.0289	0.0056	0.0003	0.1502
Vehicle	72.79±2.68	71.31±5.71	67.87±5.66	55.57±21.43	51.63±4.22	**72.95±5.82**
	0.0190	0.0431	0.0219	0.0881	0.0008	0.3495
Australian	85.00±2.68	83.41±1.99	**85.65±2.27**	70.94±12.52	85.00±2.78	85.00±3.35
	0.0943	0.0353	0.0289	0.2920	0.0014	0.1502
Housevotes	87.70±2.13	88.85±4.35	87.13±7.68	54.14±4.64	58.28±1.70	**89.20±5.38**
	0.0223	0.0676	0.0196	0.0558	0.0011	0.0820
Haberman	72.79±2.68	71.31±5.71	67.87±5.66	55.57±21.43	69.67±4.26	**72.95±5.82**
	0.0190	0.0431	0.0219	0.0881	0.0005	0.3495
Ionosphere	68.57±5.27	84.86±7.99	56.14±23.34	79.14±6.80	85.29±3.37	**85.43±7.03**
	0.2004	0.0469	0.0265	0.2444	0.0145	0.1774
Liver	53.04±8.79	57.97±4.26	53.62±9.15	59.13±5.99	57.68±2.02	**59.28±3.92**
	0.0188	0.0477	0.0283	0.0290	0.0065	0.4338
Pima	67.30±2.23	74.33±3.52	70.49±4.17	68.08±6.55	64.95±1.29	**74.92±3.96**
	0.0532	0.0406	0.0363	0.1947	0.0017	0.8700
Wilt	61.80±2.84	60.00±2.42	61.80±2.84	45.70±11.19	58.10±8.48	**65.70±4.74**
	0.0200	0.0171	0.0104	0.1062	0.0008	0.5857

Table 8
The testing accuracy and training time on Handwritten Numeral datasets(nonlinear case).

Selected numeral pairs	Lap-SVM Acc±Std(%) Time(s)	Lap-TSVM Acc±Std(%) Time(s)	Lap-LSTSVM Acc±Std(%) Time(s)	MPSVM Acc±Std(%) Time(s)	LSTSVM Acc±Std(%) Time(s)	Lap-LpLSTSVM Acc±Std(%) Time(s)
0 9	77.62±17.10	98.13±4.19	49.75±3.32	**99.63±0.56**	99.13±0.84	98.75±0.99
	0.0082	0.1338	0.0209	1.1174	0.0007	0.0289
1 4	66.50±3.74	82.13±7.48	49.75±3.32	83.75±1.93	86.00±5.62	**88.00±2.18**
	0.0083	0.0182	0.0146	0.1551	0.0008	0.0892
1 6	82.38±14.03	87.50±3.03	49.75±3.32	72.75±19.47	88.75±4.49	**89.63±4.52**
	0.0103	0.0348	0.0288	0.1128	0.0008	0.0892
2 6	88.50±12.08	96.37±1.43	49.75±3.32	95.63±2.58	94.88±3.35	**96.62±0.95**
	0.0105	0.1437	0.0458	0.0725	0.0007	0.3092
2 9	70.75±15.61	94.75±2.67	49.75±3.32	95.38±2.28	93.63±2.355	**95.63±2.50**
	0.0081	0.0570	0.0741	0.0872	0.0007	0.2575
6 9	49.75±4.77	51.00±3.87	50.25±3.32	47.87±2.19	50.25±3.24	**52.38±3.17**
	0.0081	0.1665	0.0465	0.4755	0.0007	1.3161

from the proposed Lap-LpLSTSVM is statistically significant. The proposed Lap-LpLSTSVM and compared methods are used for pairwise comparison on UCI datasets. The null hypothesis is that there is no significant difference in accuracy between the two methods on these datasets. The accuracy results on UCI datasets are shown in Table 4, and we use the MATLAB toolbox to obtain the *p*-value whose results are described in Table 5. The *p*-value for the accuracy of Table 7 is shown in Table 9. Since all the *p*-value in Tables 5 and 9 are less than 0.1, the significant differences are at the 0.1 significant level.

Figures 6–8 indicates the change in the objective function value with respect to each iteration on Handwritten Numeral datasets.

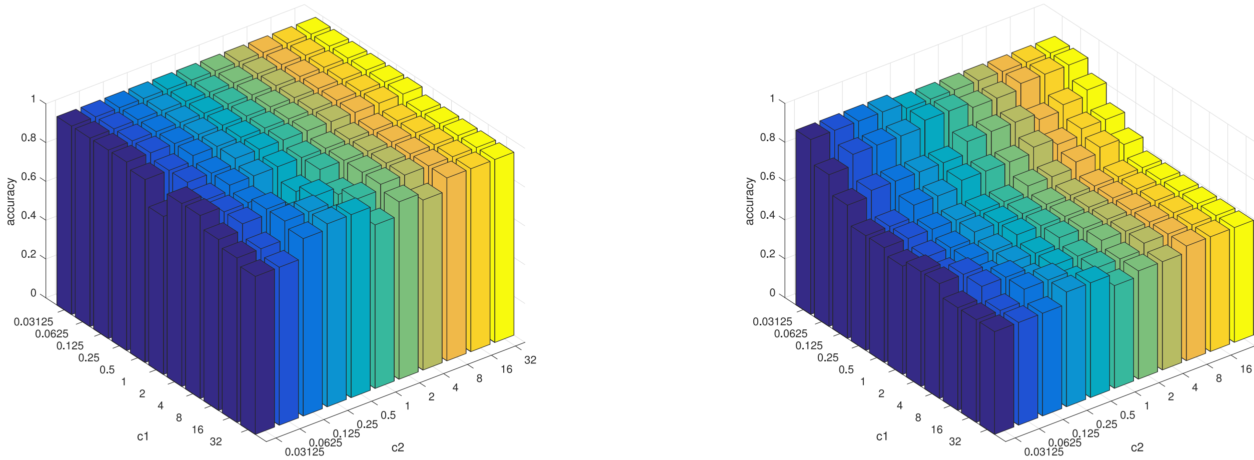


Fig. 3. Accuracy versus c_1 and c_2 on datasets Handwritten Numeral (0 9) and Handwritten Numeral (1 4).

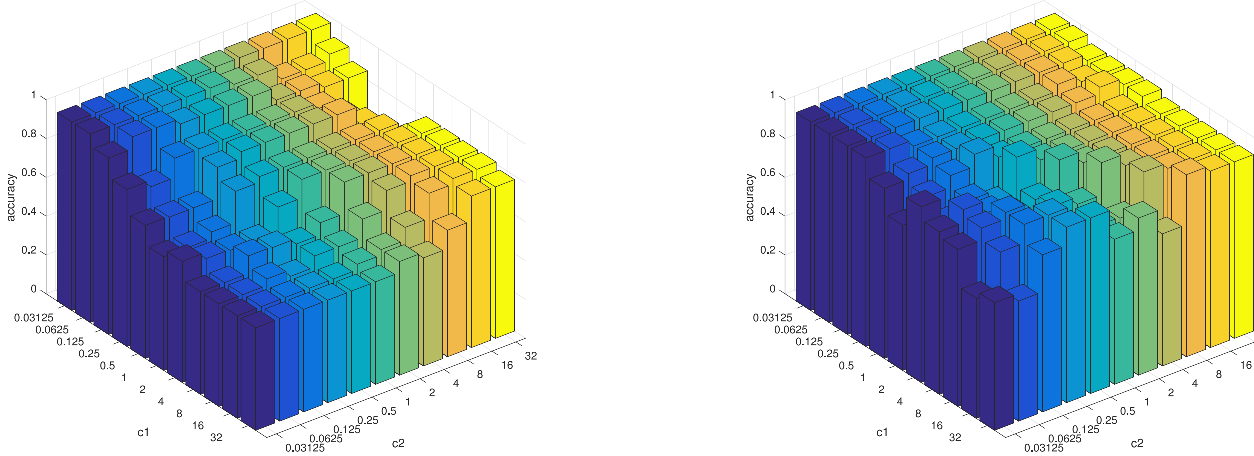


Fig. 4. Accuracy versus c_1 and c_2 on datasets Handwritten Numeral (1 6) and Handwritten Numeral (2 6).

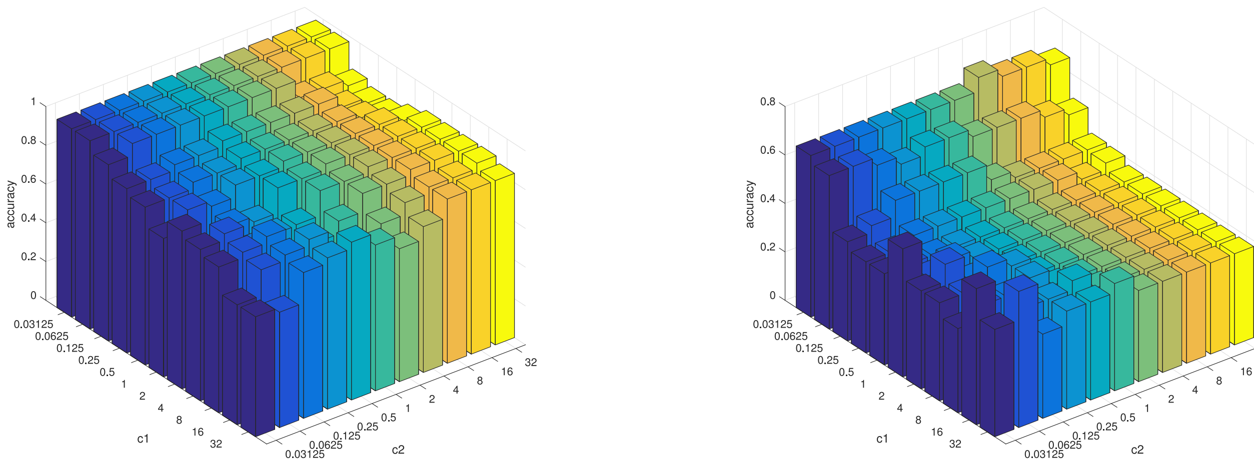


Fig. 5. Accuracy versus c_1 and c_2 on datasets Handwritten Numeral (2 9) and Handwritten Numeral (6 9).

Table 9
The p-value for the results of UCI datasets (nonlinear case) for ANOVA.

Ours V.S.	Lap-SVM	Lap-TSVM	Lap-LSTSVM	MPSVM	LSTSVM
p-value	0.0284	0.0076	0.0393	0.0116	0.0844

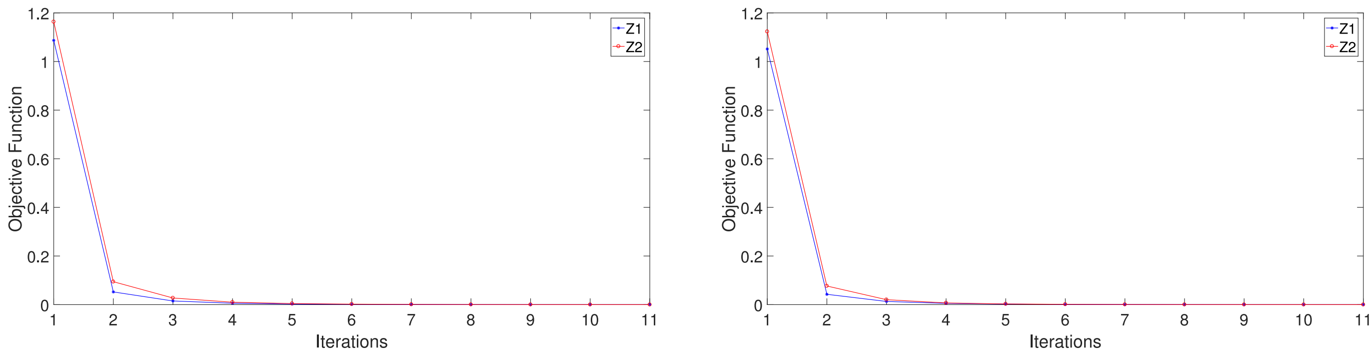


Fig. 6. Convergence curves for Z_1 and Z_2 on datasets Handwritten Numeral (0 9) and Handwritten Numeral (1 4).

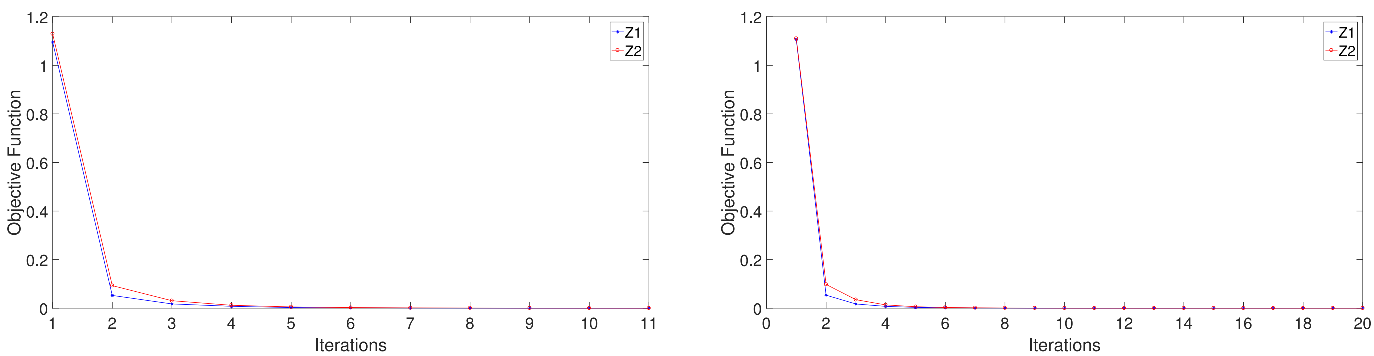


Fig. 7. Convergence curves for Z_1 and Z_2 on datasets Handwritten numerals (1 6) and Handwritten Numeral (2 6).

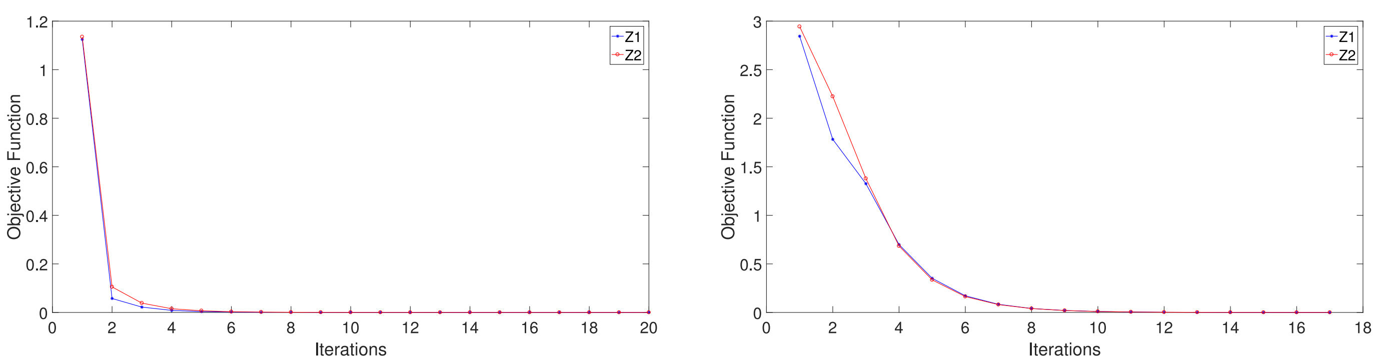


Fig. 8. Convergence curves for Z_1 and Z_2 on datasets Handwritten numerals (2 9) and Handwritten Numeral (6 9).

We can observe that the loss curve falls fast and converges within several iterations. The results empirically show the convergence of the proposed method.

There are several reasons why our method outperforms other compared state-of-the-art methods. (1) The p value of the Lp norm is adjustable, and desired performance can be achieved by choosing the appropriate value of p. (2) The proposed novel Lp graph regularization term can effectively exploit the geometric information embedded in the data, which is beneficial to find the optimal classifier. (3) The iterative strategy can effectively deal with the optimization problem, which will obtain a better performance.

5. Conclusion

In this paper, we propose a novel Laplacian Lp norm least squares twin support vector machine for semi-supervised classification. The strength of the proposed Lap-LpLSTSVM is that it can efficiently exploit the geometric information embedded in the data with the introduced Lp norm graph regularization term and the value of p is flexible. Besides, an efficient iterative method is introduced to update the parameters of two nonparallel hyperplanes. At the same time, our proposed method converges in several iterations, which shows the high efficiency of our approach.

In the experiment, the results on synthetic and real-world datasets demonstrate that the proposed method can achieve better performance than other state-of-the-art methods in linear and nonlinear cases. In order to show that our proposed semi-supervised approaches outperform their supervised counterparts, LSTSVM which uses the same labeled data as the training data is used to compare with the proposed Lap-LpLSTSVM. The results indicate the advantage of our Lap-LpLSTSVM. The disadvantage is that our method has several hyperparameters which need to be tuned. In the future, we can extend our method to unsupervised K plane clustering. Multi-view learning is also an extensible direction, and we can exploit the complementary and consensus information among views based on our method. Besides, we can also extend our model to transfer learning.

Data Availability

Laplacian Lp norm least squares twin support vector machine

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

Data will be made available on request.

Acknowledgments

This work is supported by Ningbo University talent project 421703670 as well as programs sponsored by K.C. Wong Magna Fund in Ningbo University. It is also supported by NSFC 61906101, 62006131 and 61472194.

References

[1] G. Tur, D. Hakkani-Tr, R.E. Schapire, Combining active and semi-supervised learning for spoken language understanding, Speech Commun. 45 (2) (2005) 171–186.
[2] T.S. Guzella, W.M. Caminhas, A review of machine learning approaches to spam filtering, Expert Syst. Appl. 36 (7) (2009) 10206–10222.
[3] C. Li, Z. Lin, H. Zhang, J. Guo, Learning semi-supervised representation towards a unified optimization framework for semi-supervised learning, in: IEEE International Conference on Computer Vision, 2016, pp. 2767–2775.
[4] V.N. Vapnik, Statistical learning theory, DBLP, 1998.
[5] I. El-Naqa, Y. Yang, M. Wernik, N.P. Galatsanos, R.M. Nishikawa, A support vector machine approach for detection of microclassification, in: International Conference on Image Processing, 2002, pp. 1552–1563.
[6] S. Maldonado, R. Weber, F. Famili, Feature selection for high-dimensional class-imbalanced data sets using support vector machines, Inf. Sci. 286 (2014) 228–246.
[7] F. Marchetti, E. Perracchione, Local-to-global support vector machines (LGSVMs), Pattern Recognit. 132 (2022) 108920.
[8] O.L. Mangasarian, E.W. Wild, Multisurface proximal support vector machine classification via generalized eigenvalues, IEEE Trans. Pattern Anal. Mach. Intell. 28 (1) (2006) 69–74.
[9] C. Li, Y. Shao, N. Deng, Robust L1-norm non-parallel proximal support vector machine, Optimization 65 (1) (2016) 169–183.
[10] S. Sun, X. Xie, C. Dong, Multiview learning with generalized eigenvalue proximal support vector machines, IEEE Trans. Cybern. 49 (2) (2019) 688–697.
[11] Y. Cheng, H. Yin, Q. Ye, Improved multi-view GEPSVM via inter-view difference maximization and intra-view agreement minimization, Neural Netw. 125 (2020) 313–329.
[12] R. Jayadeva, R. Khemchandani, S. Chandra, Twin support vector machines for pattern classification, IEEE Trans. Pattern Anal. Mach. Intell. 29 (5) (2007) 905–910.
[13] M.A. Kumar, M. Gopal, Application of smoothing technique on twin support vector machines, Pattern Recognit. Lett. 29 (13) (2008) 1842–1848.
[14] M.A. Kumar, M. Gopal, Least squares twin support vector machines for pattern classification, Expert Syst. Appl. 36 (4) (2009) 7535–7543.
[15] X. Pan, Z. Yang, Y. Xu, L. Wang, Safe screening rules for accelerating twin support vector machine classification, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2018) 1876–1887.
[16] X. Xie, S. Sun, Multi-view twin support vector machines, Intell. Data Anal. 19 (2015) 701–712.
[17] X. Xie, S. Sun, Multi-view Laplacian twin support vector machines, Appl. Intell. 41 (2014) 1059–1068.
[18] X. Xie, S. Sun, Multi-view support vector machines with the consensus and complementarity information, IEEE Trans. Knowl. Data Eng. 32 (12) (2020) 2401–2413.
[19] W. Chen, Y. Shao, C. Li, N. Deng, MLTSVM: a novel twin support vector machine to multi-label learning, Pattern Recognit. 52 (2016) 61–74.
[20] Z. Yang, Nonparallel hyperplanes proximal classifiers based on manifold regularization for labeled and unlabeled examples, Int. J. Pattern Recognit. Artif. Intell. 27 (5) (2013) 1–19.
[21] S. Sun, J. Shawe-Taylor, Sparse semi-supervised learning using conjugate functions, J. Mach. Learn. Res. 11 (5) (2010) 2423–2455.
[22] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (1) (2006) 2399–2434.
[23] Z. Qi, Y. Tian, Y. Shi, Laplacian twin support vector machine for semi-supervised classification, Neural Netw. 35 (2012) 46–53.
[24] W. Chen, Y. Shao, N. Deng, Laplacian least squares twin support vector machine for semi-supervised classification, Neurocomputing 145 (5) (2014) 465–476.
[25] L. Liu, R. Gong, M. Chu, Y. Peng, Nonparallel support vector machine with large margin distribution for pattern classification, Pattern Recognit. 106 (3) (2020) 107374.
[26] Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, L. Tian, Structured graph learning for clustering and semi-supervised classification, Pattern Recognit. 110 (2021) 107627.
[27] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: 31st AAAI Conference on Artificial Intelligence, 2017, pp. 2408–2414.
[28] J.H. Garrett, P. Rizzo, F. Cerda, S. Chen, J. Bielak, J. Kovacevic, Semi-supervised multiresolution classification using adaptive graph filtering with application to indirect bridge structural health monitoring, IEEE Trans. Signal Process. 62 (11) (2014) 2879–2893.
[29] L. Bull, K. Worden, N. Dervilis, Towards semi-supervised and probabilistic classification in structural health monitoring, Mech. Syst. Signal Process. 140 (2020) 106653.
[30] A.K. Agrawal, G. Chakraborty, Semi-supervised implementation of SVM-based error-correcting output code for damage-type identification in structures, Struct. Control Health Monit. 29 (8) (2022) e2967.
[31] N. Kwak, Principal component analysis based on l1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1672–1680.
[32] S. Melacci, M. Belkin, Laplacian support vector machines trained in the primal, J. Mach. Learn. Res. 12 (5) (2009) 1149–1184.
[33] W. Chen, Y. Shao, D. Xu, Y. Fu, Manifold proximal support vector machine for semi-supervised classification, Appl. Intell. 40 (4) (2014) 623–638.
[34] R. Khemchandani, C. S. Jayadeva, Optimal kernel selection in twin support vector machines, Optim. Lett. 3 (1) (2009) 77–88.

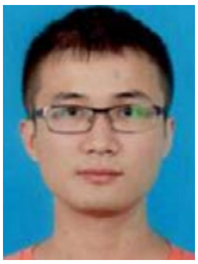
Xijiong Xiereceived his PhD degree in the Pattern Recognition and Machine Learning Research Group from the Department of Computer Science and Technology, East China Normal University in 2016. He is currently an Associate Professor in the Faculty of Electrical Engineering and Computer Science at Ningbo University, China. His research interests include kernel methods, support vector machines, multi-view learning, deep learning etc.

Feixiang Sun received his BE degree from Anhui University of Science and Technology, Huainan, China, in 2020. He is currently pursuing the master's degree in Ningbo University, Ningbo, China. His research interests include support vector machines, multi-view learning, deep learning, clustering etc.

Jiangbo Qian received his PhD degree in Computer Science from Southeast University (China) in 2006. He is currently a Professor in the Faculty of Electrical Engineering and Computer Science at Ningbo University, China. He was a visiting scholar in the Department of Computer and Information Science at The University of Michigan, USA. His research interests include big data management, deep learning, information retrieval, cloud computing, multidimensional indexing, Bloom filter, and hardware/software co-design.

Lijun Guo received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2011. He is currently a Full Professor with Ningbo University, Ningbo, China. His current research interests include computer vision, pattern recognition, and intelligence science.

Rong Zhang received the PhD degree in communication and information system at Ningbo University, Ningbo, China, in 2015. She is currently Associate Professor with College of Information Science and Engineering, Ningbo University, Ningbo, China. Her main interests are in digital image forensics, computer vision.



Xulun Yereceived the MSc and PhD degrees from Ningbo University, China, in 2016 and 2019, respectively. He is currently a Lecturer with Ningbo University. His research interests include Bayesian learning, nonparametric clustering, and convex analysis.

Zhijin Wang received the PhD degree from the Department of Computer Science and Technology, East China Normal University, Shanghai China, in 2016. He is currently With the Computer Engineering College, Jimei University, Xiamen, China. His current Research interests include recommendation system, data mining, and artificial intelligence in healthcare.