Pattern Recognition 156 (2024) 110753

Contents lists available at ScienceDirect

Pattern Recognition

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

Multi-view hypergraph regularized Lp norm least squares twin support vector machines for semi-supervised learning

Junqi Lu [a], Xijiong Xie [a,b,*], Yujie Xiong [c]

[a] School of Information Science and Engineering, Ningbo University, Ningbo 315211, China
[b] Key Laboratory of Mobile Network Application Technology of Zhejiang Province, Ningbo 315211, China
[c] The School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

ARTICLE INFO

Keywords:
Multi-view semi-supervised learning
Twin support vector machines
Lp norm graph regularization
Hypergraph regularized

ABSTRACT

In recent years, multi-view semi-supervised learning has gradually become a popular research direction. The classic binary classification methods in this field are multi-view Laplacian support vector machines (MvLapSVM) and multi-view Laplacian twin support vector machines (MvLapTSVM), which extend semi-supervised support vector machine to multi-view learning. Nevertheless, similar to the majority of SVM-based multi-view methods, the above methods are two-view methods that cannot fully leverage the information from all views and are constructed based on the L_2 norm. Additionally, in semi-supervised graph learning, the quality of the graph often has a significant impact on the results. Therefore, we propose a novel multi-view hypergraph regularized Lp norm least squares twin support vector machines (MvHGLpLSTSVM) that can handle general multi-view data for semi-supervised learning. It extends hypergraph learning to multi-view learning and combines Lp norm to further explore the manifold structure and embedded geometric information of multi-view data. By using equality constraints, we design a simple and effective iterative algorithm. In the classification of six multi-view datasets, we compare the proposed method with some other state-of-the-art methods, and the results show that the proposed method is effective.

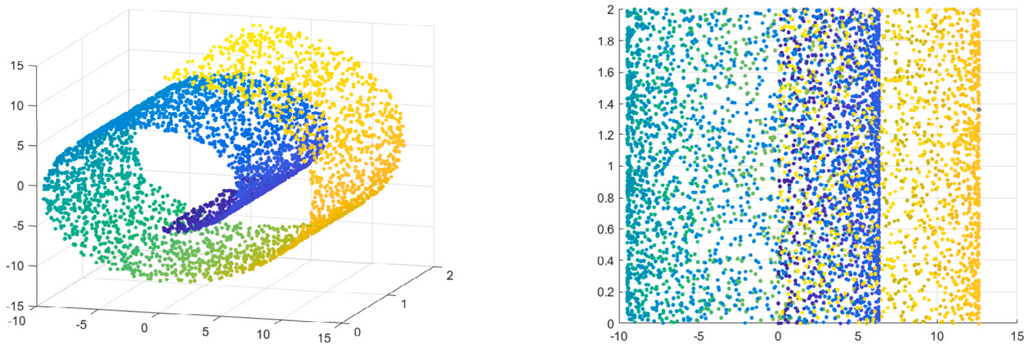
1. Introduction

Support vector machine (SVM) [1] is a machine learning method based on structural risk minimization in statistical theory. It has many advantages, such as global optimization, the sparsity of the solution, which give it good generalization performance. The sparsity of the solution is achieved by the principle of maximal margin, meaning that only the data points near the class boundary become support vectors, while data points far away from the boundary have less impact on the model. However, standard SVM is only suitable for supervised learning and cannot handle unlabeled data. In the real world, most of data are unlabeled. Investing a lot of effort in labeling data is a waste of resources and is not feasible. Insufficient labeled data can seriously degrade the performance of standard SVM. Therefore, semi-supervised learning (SSL), combining labeled and unlabeled data has emerged.

The basic idea of SSL is to use a large amount of unlabeled data to help improve the effectiveness of supervised learning methods [2]. Currently, various types of semi-supervised support vector machines are based on two common assumptions : the clustering assumption and the manifold assumption [3]. The clustering assumption reflects the global feature of the models, suggesting that samples of the same class are more likely to have the same labels. Based on this assumption, Bennett et al. [4] proposed semi-supervised support vector machines (S3VM), while Joachims et al. [5] introduced transductive support vector machines. The manifold assumption, illustrated in Fig. 1, captures the local feature of models, assuming that adjacent samples along the manifold plane share similar properties or labels. Based on this assumptions, numerous SVM-based semi-supervised classifiers have been proposed, including Laplacian support vector machines (LapSVM) [6], Laplacian twin support vector machines (LapTSVM) [7], and Laplacian p-norm proximal support vector machines (Lap-PPSVM) [8]. Since the introduction of S3VM, continuous efforts have been made to develop new SSL methods to enhance SVM. Research indicates that the computational complexity of S3VM sharply increases with the growth of unlabeled data. In response to this challenge, Glenn et al. [9] introduced concave semi-supervised support vector machines (VS3VM) in 2001. Unlike SVM, S3VM is a non-convex optimization problem [10]. Scholars have proposed various optimization solutions to address the non-convex optimization problems arising with the increase in new data. These solutions include semi-definite programming [11], gradient descent [12], DC programming and DCA programming [13]. In 2014, Li et al. [14]

* Corresponding author at: School of Information Science and Engineering, Ningbo University, Ningbo 315211, China.
E-mail addresses: 2211100268@nbu.edu.cn (J. Lu), xjxie11@gmail.com (X. Xie), xiong@sues.edu.com (Y. Xiong).

https://doi.org/10.1016/j.patcog.2024.110753
Received 22 November 2023; Received in revised form 2 July 2024; Accepted 2 July 2024
Available online 6 July 2024



(a) Diagram of manifold distribution

(b) Manifold regularization

Fig. 1. Manifold assumption posits that when two samples are proximate in the manifold (as depicted in (a)), their labels should exhibit similarity or identity, as illustrated in (b). Manifold regularization endeavors to acquire smooth functions over the unrolled manifold (b), enabling the utilization of unlabeled samples in semi-supervised learning.

proposed safe semi-supervised support vector machine (S4VM) to reduce the risk of identifying poor separators using unlabeled data by attempting to utilize multiple candidate low density separators. Subsequently, Zhang et al. [15] integrated S4VM into an ensemble learning framework, proposing the EnsembleS3VM method, which found application in the classification of polarimetric synthetic aperture radar images. On the basis of LapTSVM, Chen et al. [16] proposed Laplacian least squares twin support vector machine (LapLSTSVM), which accelerates the solution speed by solving two linear equations. Following this, Xie et al. [17] proposed Laplacian Lp norm least squares twin support vector machine (Lap-LpLSTSVM) by integrating Lp norm with LapLSTSVM and leveraging the embedded geometric information in the data. In addition, regarding the study of graphs, Sun et al. [18] proposed the hypergraph regularized semi-supervised support vector machine (HGSVM), leveraging hypergraphs to accurately capture the complex multivariate structures of data.

Multi-view learning (MVL) [19], as a popular research topic nowadays, improves generalization performance by combining multiple feature sets [20]. The proposal of MVL is due to the widespread presence of multi-view data in the real world. For instance, web page data can be characterized by employing both the textual content and the information embedded in hyperlinks. Facial expression information can be described by using thermal imaging or images. Multi-view learning algorithms follow two principles: consensus and complementarity [21]. Consensus refers to the consistency of classification or representation learned from different views, meaning that each view contains shared information for all views. Complementarity, on the other hand, underscores that each view contains its unique information, that is, specific information within the view. Fully utilizing the potential connections among different views can often improve generalization ability [22]. There are two strategies for processing multi-view data. The first is the connection strategy, which connects all views to create a single-view with comprehensive information. Representative algorithms include co-training [23], co-testing [24] and co-clustering [25]. The second strategy is a separation strategy, which constructs a learning function on each view and utilizes the consensus alignment features of each view. Representative algorithms include canonical correlation analysis (CCA) [26] and SVM-2K [27].

SVM-2K combines margin maximization problem with co-regularization term to fully utilize multi-view data information and improve the performance of SVM. Subsequently, various multi-view classification methods based on support vector machine have been continuously proposed, including multi-view Laplacian support vector machines (MvLapSVM) [28], multi-view twin support vector machine (MvTSVM) [29] and multi-view Laplacian twin support vector machines (MvLapTSVM) [30]. Tang proposed multi-view learning based on nonparallel support vector machine (MvNPSVM) [31] and multi-view privileged support vector machine (MvPSVM) [32], while Sun [33] proposed multi-view learning with generalized eigenvalue proximal support vector machines (MvGSVM). Building upon this, Xu et al. [34] proposed multi-view learning with the privileged weighted twin support vector machine (MPWTSVM), constructing a multi-view model capable of extracting intra-class information using the K-Nearest Neighbor algorithm and the consensus principle. Xie et al. [35] searched for two sparse subsets of two views through manifold-preserving graph reduction (MPGR). They intersected these two sparse subsets to form a new sparse subset, and proposed MvLapLSTSVM with MPGR. In the same year, they also proposed general multi-view semi-supervised least squares support vector machines with multi-manifold regularization (GMvLapSVM) [36], which combines the principle of consensus and complementarity. Ye et al. [37] proposed multi-view learning with robust double-sided twin SVM (MvRDTSVM) to improve its robustness by introducing bidirectional constraints and L_1 norm distance metrics. They also proposed its fast version, MvFRDTSVM, to accelerate the solution speed. Recently, Xie and Li proposed the deep multi-view multiclass twin support vector machine (DMvTSVM) [38], which integrates deep learning with multi-view learning and twin support vector machines. Multiclass classification are achieved through ove-vs-rest (OVR) strategy, which construct K independent classifiers for K classes. Each classifier treats one class as the positive class and all other classes as negative one.

Similar to standard support vector machine, most of the above methods are constructed based on the L_2 norm. However, L_2 norm is sensitive to outliers and exaggerates the influence of data points with large norm. In the study of LSSVM [39], scholars have pointed out that LSSVM using L_2 norm may ignore boundary points. Additionally, when the sample size is much smaller than the number of features, it can easily lead to be ill-condition or singularity and not suitable for small sample size (SSS) problems. Compared with the L_2 norm, L_0 norm and L_1 norm tend to yield sparser solutions. Therefore, the L_1 norm is often used as a replacement for the L_2 norm to achieve robustness [40], such as L_1-NPSVM [41], L_1-GEPSVM [42]. On the other hand, most graph regularization algorithms, such as LapSVM and LapTSVM, introduce manifold regularization terms to utilize unlabeled data for semi-supervised learning. However, graph construction based on manifold regularization terms often overlooks high-order relationships between data and cannot accurately express the complex relationships between data.

In this paper, inspired by the above mentioned methods, to enhance the model's robustness with different datasets. We achieve this by decomposing the hypergraph through eigenvalue decomposition. This allows it to be combined with the Lp norm, with the adjustability of the p value, and leverages the excellent ability of hypergraph to express complex structural relationships of multivariate data, thereby improving the generalization ability of the model. Based on these, we propose a novel multi-view hypergraph regularized Lp norm least

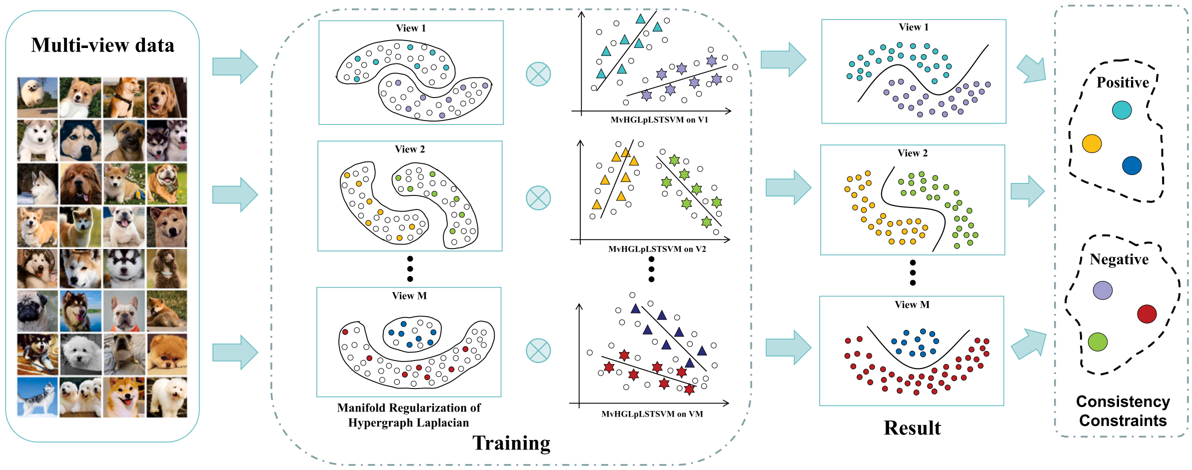


Fig. 2. The framework of MvHGLPLSTSVM. Colored circles represent labeled samples, while white circles represent unlabeled samples. Each view has the same number of labeled and unlabeled samples. By inputting multi-view data into the training set of the model. For the ith view, use hypergraph Laplacian combined with manifold regularization to more accurately represent the complex structure between data, and obtain the classification results of the ith view. Obtain the final result of multi-view classification through consistency constraints.

squares twin support vector machines (MvHGLpLSTSVM). We choose LSTSVM as the basic method for extension for the following reasons. Firstly, unlike standard SVM which requires solving a single large Quadratic Programming Problem (QPP), Twin SVM (TSVM) obtains two non-parallel hyperplanes by solving two QPPs of smaller size. Theoretical analysis indicates that the solving speed of TSVM is approximately four times faster than SVM. Secondly, compared to TSVM, LSTSVM modifies the original inequality constraints to equality constraints, and its solution is obtained by directly solving two systems of linear equations, thus avoiding the need to solve QPPs. This enables it to accurately solve without any external optimizer, achieving similar accuracy to TSVM but with faster training time. Its structure is depicted in Fig. 2. Unlike previous methods limited to handling two-view data, we construct a general multi-view learning framework with consistency constraints. This enables our method to address general multi-view semi-supervised classification problems. Compared with other multi-view semi-supervised learning methods, our contributions are as follows:

- Embed hypergraph learning into semi-supervised SVM by utilizing the hypergraph Laplacian matrix to construct manifold regularization terms, exploring the multivariate and complex relationships between data. The hypergraph is decomposed by the technique of eigenvalue decomposition, allowing its combination with the Lp norm to form a unified formula and achieving the optimal performance by selecting the appropriate p value.
- Construct a general multi-view learning framework. Extend the semi-supervised SVM that combines hypergraph regularization and Lp norm to MVL. The consistency information of multi-view data is used to construct a multi-view consensus matrix, which can handle general multi-view semi-supervised problems.
- An effective iterative algorithm is designed to solve complex optimization problems, and the solution process is simple and unified. The hyperparameters are obtained through cross-validation, and compared with other advanced methods on multiple multi-view datasets, the effectiveness of the proposed method is verified.

The remainder of this paper is structured as follows: Section 2 provides a brief introduction to the hypergraph, SVM-2K, and LapLpLSTSVM. In Section 3, we elaborate on the specifics of our proposed method, MvHGLpLSTSVM, in both linear and nonlinear cases. In Section 4 we compare our method with previous methods. Section 5 presents the experimental results, and the paper concludes in Section 6.

2. Related work

2.1. Hypergrah

Since Zhou et al. [43] proposed hypergraph, it has demonstrated its powerful learning abilities in clustering and classification. Unlike the learning method of traditional graph (simple graph) that assumes pairwise relationships between samples, the relationships between similar samples are represented by hyperedges, which can contain n vertices ($n \geq 2$). Hypergraph as a generalization of graph learning, is used for label propagation on hypergraph structures. It preserves complex relationships without losing similar information between samples, which helps improve the classification performance of the model. In Fig. 3, an example is provided to illustrate the difference between a hypergraph and the simple graph.

Given a set of vertices $V=\{v_1, v_2, \dots, v_n\}$ that $\bigcup_{e\in E} = V$ and hyperedges $E=\{e_1, e_2, \dots, e_m\}$, the definition of the incidence matrix $H \in R^{n\times m}$ is as

$$H = h(v,e) = \begin{cases} 1, & v\in e \\ 0, & v\notin e \end{cases} \tag{1}$$

Based on H, the degree of hyperedges $\delta(e)$ can be expressed as $|e| = \sum_{v\in V} h(v,e)$ and the degree of vertex $d(v)$ can be expressed as $\sum_{e\in E} w(e)h(v,e)$. The calculation method of the weight of hyperedges can be used as proposed by Hang et al. [44]:

$$w(e) = \frac{1}{\delta(e)(\delta(e)-1)} \sum_{\{v_i,v_j\}\in e} \exp(-\frac{\left\|x_i - x_j\right\|^2}{\mu}). \tag{2}$$

Hypergraph Laplacian is an important concept in hypergraph theory. Similar to the Laplacian matrix in general graph theory, there are many ways to define its Laplacian operator. In this paper, Zhou's normalized Laplacian [43] is used, which is defined as:

$$L = \boldsymbol{I} - \boldsymbol{D}_v^{-\frac{1}{2}} \boldsymbol{H} \boldsymbol{W} \boldsymbol{D}_e^{-1} \boldsymbol{H}^T \boldsymbol{D}_v^{-\frac{1}{2}}, \tag{3}$$

where $\boldsymbol{D}_v$, $\boldsymbol{D}_e$, $\boldsymbol{W}$ are the diagonal matrices composed of $d(v)$, $\delta(e)$, $w(e)$ respectively. $\boldsymbol{I}$ is an identity matrix with appropriate dimensions.

2.2. SVM-2K

In order to fully utilize the consensus information between views, the differences between different views are embedded into a collaborative regularization. Farquhar et al. [27] proposed the SVM-2K method,

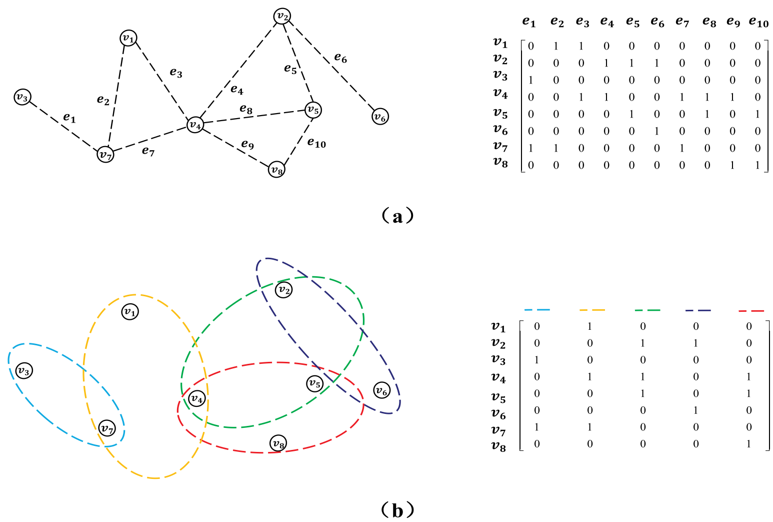


Fig. 3. Simple graph vs. Hypergraph. (a) Simple graph and its vertex-edge incidence matrix. (b) Hypergraph and its vertex-edge incidence matrix.

which jointly achieves the minimization of the loss function by combining the margin maximization problem in the support vector machine and the co-regularization term into a unified framework. SVM from two different feature spaces are integrated using similarity constraints:

$$\left| w_A^{\mathrm{T}} \phi_A(x_i^A) + b_A - w_B^{\mathrm{T}} \phi_B(x_i^B) - b_B \right| \leq \eta_i + \epsilon, \tag{4}$$

where w_A, b_A, w_B, b_A are parameters of binary SVM classifier with different feature spaces, x_i^A, x_i^B are the training samples of two views. $\phi_A(\cdot)$, $\phi_B(\cdot)$ represent the kernel function of the corresponding views. To obtain the optimal solution for the aforementioned parameters, the optimization problem for SVM-2K is expressed as follows:

$$\begin{aligned} \min_{w_A, w_B} \ & \frac{1}{2}(\|w_A\|_2^2 + \|w_B\|_2^2) + c_1 \sum_{i=1}^n \xi_i^A + c_2 \sum_{i=1}^n \xi_i^B + \lambda \sum_{i=1}^n \eta_i \\ s.t. \ & \left| w_A^{\mathrm{T}} \phi_A(x_i^A) + b_A - w_B^{\mathrm{T}} \phi_B(x_i^B) - b_B \right| \leq \eta_i + \epsilon, \\ & y_i (w_A^T \phi(x_i^A) + b_A) \geq 1 - \xi_i^A, \\ & y_i (w_B^T \phi(x_i^B) + b_B) \geq 1 - \xi_i^B, \\ & \xi_i^A \geq 0, \xi_i^B \geq 0, \eta_i \geq 0, \\ & 1 \leq i \leq n, \end{aligned} \tag{5}$$

where ξ_i^1, ξ_i^2, η_i are slack variables and $c_1, c_2, \lambda, \epsilon$ are nonnegative parameters, and set ϵ to be zero. The dual problem of the above optimization problem can be written as:

$$\begin{aligned} \min_{\zeta_i^A, \zeta_i^B, \alpha_i^A, \alpha_i^B} \ & -\frac{1}{2} \sum_{i,j=1}^n (\zeta_i^A \zeta_j^A K_A(x_i, x_j) + \zeta_i^B \zeta_j^B K_B(x_i, x_j)) + \sum_{i=1}^n (\alpha_i^A + \alpha_i^B) \\ s.t. \ & \zeta_i^A = \alpha_i^A y_i - \beta_i^+ + \beta_i^-, \\ & \zeta_i^B = \alpha_i^B y_i + \beta_i^+ - \beta_i^-, \\ & \sum_{i=1}^n \zeta_i^A = \zeta_i^B = 0 , \\ & 0 \leq \beta_i^{+/-}, \beta_i^+ + \beta_i^- \leq \lambda , \\ & 0 \leq \alpha_i^{+/-} \leq c_{1/2} , \end{aligned} \tag{6}$$

where $\alpha_i^{A/B}, \beta_i^{+/-}$ are the vectors of Lagrange multipliers. With the optimal solution of this optimization problem $\hat{w}_1, \hat{b}_1, \hat{w}_2, \hat{b}_2$. The decision function of SVM-2K is

$$f(x) = \frac{1}{2}(\hat{w}_1^{\mathrm{T}} \phi_1(x_i^1) + \hat{b}_1 + \hat{w}_2^{\mathrm{T}} \phi_2(x_i^2) + \hat{b}_2). \tag{7}$$

2.3. Lap-LpLSTSVM

Recently, Xie and Sun et al. [17] proposed a novel Laplacian Lp norm least squares twin support vector machine (Lap-LpLSTSVM), which used Lp norm to replace L_2 norm in Laplacian least squares twin support vector machine (Lap-LSTSVM) [16]. It can be more effectively utilize unlabeled data by introducing an Lp norm graph regularization term utilizing eigenvalue decomposition, harnessing data's embedded geometric information, and enhancing the model's generalization capability. Suppose the total number of training samples is m in the d dimensional real space, with l labeled samples and u unlabeled samples, which $l + u = m$, and the positive sample number of labeled samples is m_1 and the negative sample number is m_2. Lap-LpLSTSVM can be given as follows:

$$\min_{Z_1} F_1 = \frac{1}{2} \|HZ_1\|_p^p + c_1 \|Z_1\|_p^p + c_2 \|GZ_1 + e_2\|_p^p + c_3 \|PZ_1\|_p^p \ , \tag{8}$$

$$\min_{Z_2} F_2 = \frac{1}{2} \|GZ_2\|_p^p + c_1 \|Z_2\|_p^p + c_2 \|-HZ_2 + e_1\|_p^p + c_3 \|PZ_2\|_p^p \ , \tag{9}$$

where $H = \left[A \ e_1\right], G = \left[B \ e_2\right], Z_k = \left[w_k^T \ b_k\right]^T (k = 1, 2)$, and $A \in R^{m_1 \times d}, B \in R^{m_2 \times d}$ the positive and negative data respectively, c_1, c_2, $c_3 > 0$ are the regularization hyperparameters, e_1, e_2 are the vectors of ones with suitable dimensions. The Lp norm can be expressed as $\|x\|_p^p = \left(\sum_{i=1}^n |x_i|^p\right)^{\frac{1}{p}}$. For matrix P, since the Laplacian matrix L is symmetric, allowing it to be decomposed using eigenvalue decomposition as $L = Q\Lambda Q^T$, where Q is a matrix of eigenvectors, and Λ is a diagonal matrix with eigenvalues along the diagonal. The graph regularization term of Lap-LpLSTSVM can be expressed as:

$$\left(JZ_k\right)^T Q\Lambda Q^T \left(JZ_k\right) = \|PZ_k\|^2, \tag{10}$$

where $J = [C \ e], C \in R^{m \times d}$ includes all the labeled and unlabeled data, $e \in R^{m \times 1}$ and $P = \Lambda^{\frac{1}{2}} Q^T J$. In order to solve Eq. (8) and Eq. (9) more conveniently, a diagonal matrix is introduced, the above equations can be written as:

$$\begin{aligned} \min_{Z_1} F_1 = & \frac{1}{2} Z_1^T H^T D_1 H Z_1 + c_1 Z_1^T D_2 Z_1 + c_2 (GZ_1 + e_2)^T D_3 (GZ_1 + e_2) \\ & + c_3 Z_1^T P^T D_4 P Z_1 \ , \end{aligned} \tag{11}$$

$$\begin{aligned} \min_{Z_2} F_2 = & \frac{1}{2} Z_2^T G^T E_1 G Z_2 + c_1 Z_2^T E_2 Z_2 + c_2 (-HZ_2 + e_1)^T E_3 (-HZ_2 + e_1) \\ & + c_3 Z_2^T P^T E_4 P Z_2 \ , \end{aligned} \tag{12}$$

where

$$D_1 = diag(\frac{1}{|H_1Z_1|^{2-p}}, \frac{1}{|H_2Z_1|^{2-p}}, \dots, \frac{1}{|H_{m1}Z_1|^{2-p}}),$$
$$D_2 = diag(\frac{1}{|Z_{11}|^{2-p}}, \frac{1}{|Z_{12}|^{2-p}}, \dots, \frac{1}{|Z_{1(d+1)}|^{2-p}}),$$
$$D_3 = diag(\frac{1}{|G_1Z_1+1|^{2-p}}, \frac{1}{|G_2Z_1+1|^{2-p}}, \dots, \frac{1}{|G_{m2}Z_1+1|^{2-p}}),$$
$$D_4 = diag(\frac{1}{|P_1Z_1|^{2-p}}, \frac{1}{|P_2Z_1|^{2-p}}, \dots, \frac{1}{|P_mZ_1|^{2-p}}),$$
$$E_1 = diag(\frac{1}{|G_1Z_2|^{2-p}}, \frac{1}{|G_2Z_2|^{2-p}}, \dots, \frac{1}{|G_{m2}Z_2|^{2-p}}),$$
$$E_2 = diag(\frac{1}{|Z_{21}|^{2-p}}, \frac{1}{|Z_{22}|^{2-p}}, \dots, \frac{1}{|Z_{2(d+1)}|^{2-p}}),$$
$$E_3 = diag(\frac{1}{|-H_1Z_2+1|^{2-p}}, \frac{1}{|-H_2Z_2+1|^{2-p}}, \dots, \frac{1}{|-H_{m1}Z_2+1|^{2-p}}),$$
$$E_4 = diag(\frac{1}{|P_1Z_2|^{2-p}}, \frac{1}{|P_2Z_2|^{2-p}}, \dots, \frac{1}{|P_mZ_2|^{2-p}}),$$

here 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. $Z_{ki}\,(k=1,2)$ denotes the ith element of Z_k. By taking the derivative of Eq. (11) and Eq. (12) with respect to Z_1 and Z_2 and setting them to zero,we can derive the solution as follows:

$$\nabla_{Z_1}F_1 = H^T D_1 H Z_1 + 2c_1 D_2 Z_1 + 2c_2 G^T D_3 \left(GZ_1 + e_2\right) + 2c_3 P^T D_4 P Z_1 = 0, \tag{13}$$
$$\nabla_{Z_2}F_2 = G^T E_1 G Z_2 + 2c_1 E_2 Z_2 - 2c_2 H^T E_3 \left(-HZ_2 + e_1\right) + 2c_3 P^T E_4 P Z_2 = 0. \tag{14}$$

Then we can obtain:

$$Z_1 = -2c_2\left(H^T D_1 H + 2c_1 D_2 + 2c_2 G^T D_3 G + 2c_3 P^T D_4 P\right)^{-1} G^T D_3 e_2, \tag{15}$$
$$Z_2 = 2c_2\left(G^T E_1 G + 2c_1 E_2 + 2c_2 H^T E_3 H + 2c_3 P^T E_4 P\right)^{-1} H^T E_3 e_1. \tag{16}$$

After we obtain the two nonparallel hyperplanes, the prediction function of new data can be written as

$$f(x) = sign\left(\frac{w_1^T x + b_1}{\|w_1\|_2} - \frac{w_2^T x + b_2}{\|w_2\|_2}\right). \tag{17}$$

3. Our proposed methods

Based on the multi-view semi-supervised learning framework, we broaden the single-view Lap-LpLSTSVM to multi-view learning by incorporating multi-view regularization terms that connect predictions from diverse viewpoints. In order to further explore the manifold structure and embedded geometric information of multi-view data, we introduced hypergraph learning, which employs hypergraph regularized to replace the original simple graph structure information.

Suppose a multi-view dataset with m training samples training is denoted as: $S^i (i = 1, 2, \dots, M)$, M is the number of views. For the ith view, the training dataset can be represented as:

$$S^i = \left\{ \left(x_1^{(i)}, y_1\right), \dots, \left(x_l^{(i)}, y_l\right), x_{l+1}^{(i)}, \dots, x_{l+u}^{(i)} \right\},$$

which included l labeled samples and u unlabeled samples and $l+u = m$. Let $X_l^{(i)} = \{x_j^{(i)}\}_{j=1}^{l} \in R^{l\times d_i}$, $X_u^{(i)} = \{x_j^{(i)}\}_{j=l+1}^{l+u} \in R^{u\times d_i}$ denote the labeled data and unlabeled data respectively, d_i is the feature space dimension of examples in the ith view, $Y_l^{(i)} = \{y_i^{(i)}\}_{j=1}^{l} \in \{-1, +1\}$ denote the labels. For labeled data we set matrix $A_i \in R^{m_1\times d_i}$, $B_i \in R^{m_2\times d_i}$ to denote the "+1" class and "−1" class, where $m_1 + m_2 = l$.

3.1. Linear MvHGLpLSTSVM

The optimization problem for linear MvHGLpLSTSVM is formulated as:

$$\min_{v_i}\ \mathcal{L}_1 = \frac{1}{2}\sum_{i=1}^{M}\|G_i v_i\|_p^p + c_1\sum_{i=1}^{M}\|v_i\|_p^p + c_2\sum_{i=1}^{M}\|H_i v_i + e_2\|_p^p + c_3\sum_{i=1}^{M}\sum_{j=1,j>i}^{M}\left\|G_i v_i - G_j v_j\right\|_p^p + c_4\sum_{i=1}^{M}\|P_i v_i\|_p^p, \tag{18}$$
$$\min_{u_i}\ \mathcal{L}_2 = \frac{1}{2}\sum_{i=1}^{M}\|H_i u_i\|_p^p + c_1\sum_{i=1}^{M}\|u_i\|_p^p + c_2\sum_{i=1}^{M}\|-G_i u_i + e_1\|_p^p + c_3\sum_{i=1}^{M}\sum_{j=1,j>i}^{M}\left\|H_i u_i - H_j u_j\right\|_p^p + c_4\sum_{i=1}^{M}\|P_i u_i\|_p^p, \tag{19}$$

where $G_i = \left[A_i\ e_1\right], H_i = \left[B_i\ e_2\right]$, e_1 and e_2 are vectors of ones with appropriate dimensions. v_i and u_i are hyperplane parameters of positive and negative examples respectively to the ith view. Similar to Eq. (10), for the ith view, eigendecompose the corresponding Laplacian matrix $L_i = Q_i \Lambda_i Q_i^T$, then $P_i = \Lambda_i^{\frac{1}{2}} Q_i^T J_i$, c_1, c_2, c_3, c_4 are non-negative hyperparameters obtained by the grid method. Before optimizing the aforementioned problems, we provide the geometric interpretation. In Eq. (18) for the ith view, we conduct the following analysis:

(1) $\|G_i v_i\|_p^p + c_2\|H_i v_i + e_2\|_p^p$ is empirical risk of positive examples, which is similar to the loss in standard SVM. It considers the loss of labeled and unlabeled samples, used to measure the model's fit to the training data. The loss of unlabeled samples is influenced by the graph Laplacian, which helps to correlate unlabeled samples with their surrounding samples, thereby utilizing the distribution information of unlabeled samples during the learning process. And c_2 is the penalty parameter that determines the trade-off between the loss terms in Eq. (18).

(2) $c_1\|v_i\|_p^p$ is the Reproducing Kernel Hilbert Space (RKHS) regularization terms, c_1 controls the complexity of $\mathcal{L}$ in the associated Hilbert Space. For RKHS regularization terms, it uses kernel functions to calculate the complexity of the model in the feature space, mapping the data in the input space to the high dimensional feature space, thus making linearly indivisible problems linearly separable in the high dimensional space.

(3) $c_3\left\|G_i v_i - G_j v_j\right\|_p^p$ represents a co-regularization term which aims at minimizing inconsistency among views, obtaining the optimal consensus matrix, and enhancing the consistency of distinct views from multiple perspectives. c_3 is the penalty parameters.

(4) $c_4\|P_i v_i\|_p^p$ is a manifold regularization term. c_4 controls the function's complexity within the inherent geometry of the marginal distribution. In the SSL framework, the manifold learning assumes that in a high dimensional data space, the actual data distribution is typically situated on one or more low dimensional manifolds. In short, this is like penalizing the "rapid changes" of the classification function when assessed among neighboring samples on the graph [45].

Then we introduce diagonal matrix, the Eq. (18) and Eq. (19) can be rewritten as:

$$\begin{aligned}\min_{v_i}\ \mathcal{L}_1 = &\frac{1}{2}\sum_{i=1}^{M}\left(G_i v_i\right)^T D_{1i}\left(G_i v_i\right) + c_1\sum_{i=1}^{M} v_i^T D_{2i} v_i \\ &+ c_2\sum_{i=1}^{M}\left(H_i v_i + e_2\right)^T D_{3i}(H_i v_i + e_2) \\ &+ c_3\sum_{i=1}^{M}\sum_{j=1,j>i}^{M}\left(G_i v_i - G_j v_j\right)^T D_{4i}\left(G_i v_i - G_j v_j\right) \\ &+ c_4\sum_{i=1}^{M}\left(P_i v_i\right)^T D_{5i}\left(P_i v_i\right),\end{aligned} \tag{20}$$

$$I_1 = \begin{pmatrix} G_1^T\left(D_{4_{12}} + \cdots + D_{4_{1M}}\right)G_1 & -G_1^T D_{4_{12}} G_2 & \cdots & -G_1^T D_{4_{1M}} G_M \\ -G_2^T D_{4_{12}} G_2 & G_2^T\left(D_{4_{12}} + \cdots + D_{4_{2M}}\right)G_2 & \cdots & -G_2^T D_{4_{2M}} G_M \\ \cdots & \cdots & \ddots & \cdots \\ -G_M^T D_{4_{1M}} G_1 & -G_M^T D_{4_{2M}} G_2 & \cdots & G_M^T\left(D_{4_{1M}} + \cdots + D_{4_{(M-1)M}}\right)G_M \end{pmatrix},$$

$$I_2 = \begin{pmatrix} H_1^T\left(E_{4_{12}} + \cdots + E_{4_{1M}}\right)H_1 & -H_1^T E_{4_{12}} H_2 & \cdots & -H_1^T E_{4_{1M}} H_M \\ -H_2^T E_{4_{12}} H_2 & H_2^T\left(E_{4_{12}} + \cdots + E_{4_{2M}}\right)H_2 & \cdots & -H_2^T E_{4_{2M}} H_M \\ \cdots & \cdots & \ddots & \cdots \\ -H_M^T E_{4_{1M}} H_1 & -H_M^T E_{4_{2M}} H_2 & \cdots & H_M^T\left(E_{4_{1M}} + \cdots + E_{4_{(M-1)M}}\right)H_M \end{pmatrix},$$

Box I.

$$\begin{aligned} \min_{u_i} \mathcal{L}_2 = & \frac{1}{2}\sum_{i=1}^{M}\left(H_i u_i\right)^T E_{1i}\left(H_i u_i\right) + c_1 \sum_{i=1}^{M} u_i^T E_{2i} u_i \\ & + c_2 \sum_{i=1}^{M}\left(-G_i u_i + e_1\right)^T E_{3i}(-G_i u_i + e_1) \\ & + c_3 \sum_{i=1}^{M}\sum_{j=1,j>i}^{M}\left(H_i u_i - H_j u_j\right)^T E_{4i}\left(H_i u_i - H_j u_j\right) \\ & + c_4 \sum_{i=1}^{M}\left(P_i u_i\right)^T E_{5i}\left(P_i u_i\right), \end{aligned} \tag{21}$$

where

$$D_{1i} = diag(\frac{1}{\left|G_{i1}v_i\right|^{2-p}}, \frac{1}{\left|G_{i2}v_i\right|^{2-p}}, \ldots, \frac{1}{\left|G_{im_1}v_i\right|^{2-p}}),$$
$$D_{2i} = diag(\frac{1}{\left|v_{i1}\right|^{2-p}}, \frac{1}{\left|v_{i2}\right|^{2-p}}, \ldots, \frac{1}{\left|v_{i(d_i+1)}\right|^{2-p}}),$$
$$D_{3i} = diag(\frac{1}{\left|H_{i1}v_i + 1\right|^{2-p}}, \frac{1}{\left|H_{i2}v_i + 1\right|^{2-p}}, \ldots, \frac{1}{\left|H_{im_2}v_i + 1\right|^{2-p}}),$$
$$D_{4ij} = diag(\frac{1}{\left|G_{i1}v_i - G_{j1}v_j\right|^{2-p}}, \frac{1}{\left|G_{i2}v_i - G_{j2}v_j\right|^{2-p}}, \ldots, \frac{1}{\left|G_{im_1}v_i - G_{jm_1}v_j\right|^{2-p}}),$$
$$D_{5i} = diag(\frac{1}{\left|P_{i1}v_i\right|^{2-p}}, \frac{1}{\left|P_{i2}v_i\right|^{2-p}}, \ldots, \frac{1}{\left|P_{im}v_i\right|^{2-p}}),$$
$$E_{1i} = diag(\frac{1}{\left|H_{i1}u_i\right|^{2-p}}, \frac{1}{\left|H_{i2}u_i\right|^{2-p}}, \ldots, \frac{1}{\left|H_{im_2}u_i\right|^{2-p}}),$$
$$E_{2i} = diag(\frac{1}{\left|u_{i1}\right|^{2-p}}, \frac{1}{\left|u_{i2}\right|^{2-p}}, \ldots, \frac{1}{\left|u_{i(d_i+1)}\right|^{2-p}}),$$
$$E_{3i} = diag(\frac{1}{\left|-G_{i1}u_i + 1\right|^{2-p}}, \frac{1}{\left|-G_{i2}u_i + 1\right|^{2-p}}, \ldots, \frac{1}{\left|-G_{im_1}u_i + 1\right|^{2-p}}),$$
$$E_{4ij} = diag(\frac{1}{\left|H_{i1}u_i - H_{j1}u_j\right|^{2-p}}, \frac{1}{\left|H_{i2}u_i - H_{j2}u_j\right|^{2-p}}, \ldots, \frac{1}{\left|H_{im_2}u_i - H_{jm_2}u_j\right|^{2-p}}),$$
$$E_{5i} = diag(\frac{1}{\left|P_{i1}u_i\right|^{2-p}}, \frac{1}{\left|P_{i2}u_i\right|^{2-p}}, \ldots, \frac{1}{\left|P_{im}u_i\right|^{2-p}}).$$

Algorithm 1 . Linear MvHGLpLSTSVM

1: **Input:** Multi-view training data $S^i (i = 1, 2, ..., M)$, hyperparameters $c_1,\ c_2,\ c_3,\ c_4$, convergence constant ϵ and max iterations k_{max} .
2: Calculate the hyperedge weight $w(e)$ by Eq (2) and construct the hypergraph L_i of the i-th view by Eq (3).
3: Initialize v and u with the vectors of ones.
4: **for** $i \le k_{max}$ or $abs(v^t - v^{t+1}) \le \epsilon$ and $abs(u^t - u^{t+1}) \le \epsilon$ **do**
5: Update v by Eq (25).
6: Update u by Eq (26).
7: $i = i + 1$.
8: **end for**
9: **Output:** Given a multi-view test data $x = \left(x_1 \cdots x_M\right)$, if the result of Eq (27) > 0, it is classified to class +1, otherwise class -1.

Then, we define the following matrices, I_1 and I_2 consistency matrices of the positive and negative classes (see Box I).

$$\begin{aligned} & G = \mathrm{diag}(G_1, G_2, \ldots, G_M),\ H = \mathrm{diag}(H_1, H_2, \ldots, H_M), \\ & P = \mathrm{diag}(P_1, P_2, \ldots, P_M),\ D_i = \mathrm{diag}(D_{i1}, D_{i2}, \ldots, D_{iM}), \\ & E_i = \mathrm{diag}(E_{i1}, E_{i2}, \ldots, E_{iM}),\ v = \left(v_1^T \cdots v_M^T\right)^T,\quad u = \left(u_1^T \cdots u_M^T\right)^T. \end{aligned} \tag{22}$$

The original optimization problems for Eq. (20) and Eq. (21) can be rewritten as the following forms:

$$\begin{aligned} \min_{v} \mathcal{L}_1 = & \frac{1}{2}v^T G^T D_1 G v + c_1 v^T D_2 v + c_2 (Hv + e)^T D_3 (Hv + e) \\ & + c_3 v^T I_1 v + c_4 v^T P^T D_5 P v, \end{aligned} \tag{23}$$

$$\begin{aligned} \min_{u} \mathcal{L}_2 = & \frac{1}{2}u^T H^T E_1 H u + c_1 u^T E_2 u + c_2 (-Gu + e)^T E_3 (-Gu + e) \\ & + c_3 u^T I_2 u + c_4 u^T P^T E_5 P u. \end{aligned} \tag{24}$$

By taking the derivative of v and u in Eq. (23) and Eq. (24) respectively, then set them to zero, we can obtain:

$$\begin{aligned} v = & -\left(G^T D_1 G + 2c_1 D_2 + 2c_2 H^T D_3 H + 2c_3 I_1 + 2c_4 P^T D_5 P\right)^{-1} \\ & \cdot 2c_2 H^T D_3 e, \end{aligned} \tag{25}$$

$$u = \left(H^T E_1 H + 2c_1 E_2 + 2c_2 G^T E_3 G + 2c_3 I_2 + 2c_4 P^T E_5 P\right)^{-1} \cdot 2c_2 G^T E_3 e. \tag{26}$$

Upon obtaining optimal parameters, the prediction function of linear MvHGLpLSTSVM is as follows:

$$f = sign(\sum_{i=1}^{M}\left|[x_i, 1]^T u_i\right| - \sum_{i=1}^{M}\left|[x_i, 1]^T v_i\right|). \tag{27}$$

The algorithm of linear MvHGLpLSTSVM is shown in Algorithm 1.

3.2. Nonlinear MvHGLpLSTSVM

For nonlinear classification, we extend MvHGLpLSTSVM using appropriate kernel functions. The kernel-generated hyperplanes of the ith view can be written as:

$$K\left(x_i^T, C_i^T\right) w_i^+ + b_i^+ = 0, \quad K\left(x_i^T, C_i^T\right) w_i^- + b_i^- = 0, \tag{28}$$

where $K(\cdot,\cdot)$ is the kernel function we can choose. $C_i \in R^{(l+u)\times d_i}$ represents training data of the ith view. Then we define:

$$\begin{aligned} &G_{\phi i} = (K(A_i, C_i^T), e_1), H_{\phi i} = (K(B_i, C_i^T), e_2),\\ &J_{\phi i} = \left(K(C_i, C_i^T), e\right), P_{\phi i} = \left(\Lambda_{\phi i}^{\frac{1}{2}} Q_{\phi i}{}^T J_{\phi i}\right),\\ &v = \begin{pmatrix} w_i^+ \\ b_i^+ \end{pmatrix}, u = \begin{pmatrix} w_i^- \\ b_i^- \end{pmatrix}. \end{aligned} \tag{29}$$

The optimization problems for nonlinear MvHGLpLSTSVM can be expressed as:

$$\begin{aligned} \min_{v} \mathcal{L}_1 = &\frac{1}{2}\sum_{i=1}^{M}\left\|G_{\phi i} v_i\right\|_p^p + c_1 \sum_{i=1}^{M}\|v_i\|_p^p + c_2\sum_{i=1}^{M}\left\|H_{\phi i} v_i + e_2\right\|_p^p \\ &+ c_3\sum_{i=1}^{M}\sum_{j=1,j>i}^{M}\left\|G_{\phi i} v_i - G_{\phi j} v_j\right\|_p^p + c_4\sum_{i=1}^{M}\left\|P_{\phi i} v_i\right\|_p^p, \end{aligned} \tag{30}$$

$$\begin{aligned} \min_{u} \mathcal{L}_2 = &\frac{1}{2}\sum_{i=1}^{M}\left\|H_{\phi i} u_i\right\|_p^p + c_1 \sum_{i=1}^{M}\|u_i\|_p^p + c_2\sum_{i=1}^{M}\left\|-G_{\phi i} u_i + e_1\right\|_p^p \\ &+ c_3\sum_{i=1}^{M}\sum_{j=1,j>i}^{M}\left\|H_{\phi i} u_i - H_{\phi j} u_j\right\|_p^p + c_4\sum_{i=1}^{M}\left\|P_{\phi i} u_i\right\|_p^p. \end{aligned} \tag{31}$$

Similarly to the linear MvHGLpLSTSVM, we can obtain the optimal solution for hyperplane parameters v and u after using kernel functions:

$$\begin{aligned} v = &-\left(G_\phi^T D_{\phi 1} G_\phi + 2c_1 D_{\phi 2} + 2c_2 H_\phi^T D_{\phi 3} H_\phi + 2c_3 I_{\phi 1} + 2c_4 P_\phi^T D_{\phi 5} P_\phi\right)^{-1} \\ &\cdot 2c_2 H_\phi^T D_{\phi 3} e, \end{aligned} \tag{32}$$

$$\begin{aligned} u = &\left(H_\phi^T E_{\phi 1} H_\phi + 2c_1 E_{\phi 2} + 2c_2 G_\phi^T E_{\phi 3} G_\phi + 2c_3 I_{\phi 2} + 2c_4 P_\phi^T E_{\phi 5} P_\phi\right)^{-1} \\ &\cdot 2c_2 G_\phi^T E_{\phi 3} e, \end{aligned} \tag{33}$$

where G_ϕ, H_ϕ, P_ϕ, $D_{\phi i}$, $E_{\phi i}$ $(i = 1,2,\dots,5)$, $I_{\phi k}$ $(k = 1,2)$ are the matrices corresponding to Eq. (22) that are mapped from a low dimensional space to higher dimensional space by using a kernel function. After obtaining the updated values of v and u using the above equations, for a new data point x, we use the following formula to predict its class:

$$f = sign\left(\sum_{i=1}^{M}\left|[K(x_i^T, C_i^T), 1]^T u_i\right| - \sum_{i=1}^{M}\left|[K(x_i^T, C_i^T), 1]^T v_i\right|\right). \tag{34}$$

In summary, the nonlinear MvHGLpLSTSVM algorithm is given in Algorithm 2.

3.3. Time complexity

In this section, we analyze the time complexity of MvHGLpLSTSVM. For the training process, N represents the total number of training samples, d_i represents the feature dimension corresponding to the ith view, M represents the number of views and T represents the maximum number of iterations. Based on LSTSVM, the optimal solution is obtained by solving a linear equation system. Therefore, for linear MvHGLpLSTSVM, we need to solve an inverse matrix of size $\sum_{i=1}^{M}(d_i+1)\times\sum_{i=1}^{M}(d_i+1)$, set $\rho = \sum_{i=1}^{M}(d_i+1)$, so its time complexity is $O(2T\rho^3)$. For the nonlinear MvHGLpLSTSVM, set $\Omega = (N+1)\times M$, need to solve an inverse matrix of size $\Omega\times\Omega$, so the required time complexity is $O(2T\Omega^3)$.

Algorithm 2 . Nonlinear MvHGLpLSTSVM

1: **Input:** Multi-view training data $S^i (i = 1,2,\dots,M)$, hyperparameters c_1, c_2, c_3, c_4, appropriate kernel function $K(\cdot,\cdot)$ and its parameters σ, convergence constant ϵ and max iterations k_{max} .
2: Calculate the matrices $G_{\phi i}$, $H_{\phi i}$, $J_{\phi i}$, $P_{\phi i}$ mapped using kernel function in the i-th view.
3: Calculate the hyperedge weight $w_\phi(e)$ by Eq (2) and construct the hypergraph $L_{\phi i}$ of the i-th view by Eq (3).
4: Initialize v and u with the vectors of ones.
5: **for** $i \le k_{max}$ or $abs(v^t - v^{t+1}) \le \epsilon$ and $abs(u^t - u^{t+1}) \le \epsilon$ **do**
6: Update v by Eq (32).
7: Update u by Eq (33).
8: $i = i + 1$.
9: **end for**
10: **Output:** Given a multi-view test data $x = \left(x_1 \cdots x_M\right)$, if the result of Eq (34) > 0, it is classified to class +1, otherwise class -1.

4. Connections with previous methods

4.1. MvHGLpLSTSVM vs. Lap-LpLSTSVM

Lap-LpLSTSVM can effectively solve the single-view semi-supervised problem by combining Lp norm with LapLSTSVM. Based on this, we extend it to multi-view learning and our method is able to handle multi-view data. The difference between the two is as follows.

1. Lap-LpLSTSVM is a single-view method. MvHGLpLSTSVM can not only process single-view data, but also deal with general multi-view data, using the potential relationships between views to maintain the accuracy of prediction results through consistency constraints.
2. Unlike most methods, the manifold regularization term of our method considers not only pairwise relationships between data points. MvHGLpLSTSVM utilizes the multivariate manifold structure of the hypergraph to better explore the high-order correlation among data, thus obtaining better performance [18, 46].

4.2. MvHGLpLSTSVM vs. MVNPSVM

MvHGLpLSTSVM and MVNPSVM are both proposed for multi-view classification, effectively utilize the consensus principle of MVL. However, MVNPSVM cannot utilize unlabeled information and is not suitable for semi-supervised learning. There are significant differences between the two.

1. MvHGLpLSTSVM, based on the manifold assumption and utilizing manifold structure by hypergraph, can effectively leverage the unlabeled information in samples to assist in training. Even with a limited number of labeled samples, it can still perform remarkably well. In contrast, MVNPSVM, as a supervised learning method, exhibits a significant decrease in classification performance under such circumstances.
2. MvHGLpLSTSVM is a general multi-view learning method. Unlike MVNPSVM, which can only handle two-view problems, MvHGLpLSTSVM can utilize information from all views, fully exploiting the latent connections between views, and achieving better performance in multi-view classification tasks.
3. Unlike MVNPSVM, which is constructed based on the L_2 norm, MvHGLpLSTSVM adopts Lp norm and the adjustability of the p value, resulting in less affected by outliers and noise. As a result, it exhibits the excellent performance across different datasets.

Table 1
Description of the real-world datasets.

Dataset	Samples	Attributes	Views	Classes
Handwritten	2000	649	6	10
NUS	2400	1134	6	12
BBCSport	544	6386	2	5
Caltech-101	1474	3766	6	7
Cora	2708	9553	2	7
WebKB	203	2163	3	4

Table 2
The specific number of each class in the imbalanced dataset.

Dataset	1	2	3	4	5	6	7
WebKB	21	66	107	9			
BBCSport	62	104	193	124	61		
Cora	818	180	217	426	351	418	298
Caltech-101	435	798	52	34	35	64	56

4.3. MvHGLpLSTSVM vs. GMvLapSVM vs. GMvLapTSVM

GMvLapSVM and GMvLapTSVM are natural extensions of MvLapSVM and MvLapTSVM to general multi-view methods. Similar to MvHGLpLSTSVM, they can address the general multi-view classification problem by jointly learning multiple distinct views in a non-paired way. However, they differ in specific studies.

1. MvHGLpLSTSVM adopts manifold regularization by hypergraph on each view, while GMvLapSVM and GMvLapTSVM utilize manifold regularization terms based on simple graph. This allows our method to accurately represent the complex structure of the data.
2. Similarly, MvHGLpLSTSVM is based on LSTSVM, modifying inequality constraints into equality constraints. The system of linear equations is directly solved to obtain its solution without any external optimizers. By incorporating the Lp norm, it enhances the model's insensitivity to noise and better utilizes the geometric information embedded in the data.

5. Experiments

In this section, we validate the binary classification performance of MvHGLpLSTSVM on *Handwritten*, *Course3*, *BBCSport*, *Caltech101-7*, *Cora* and *WebKB* multi-view datasets. Table 1 provides detailed descriptions of the real-world datasets. The number of each class in the Handwritten and NUS is 200, while the number of each class varies across the remaining four multi-view datasets, as shown in Table 2. Our experiment was conducted on Matlab R2021a with Intel (R) Core (TM) i7-9700K CPU @ 3.60 GHz and 32 GB RAM.

For the partitioning of the dataset, in order to avoid the imbalance in sample size caused by random selection, we maintain an equal proportion of positive and negative classes. 8% of the samples are labeled as training samples, 52% are unlabeled as training samples, and the remaining 40% are test samples. Regarding the convergence conditions, we set the convergence threshold ϵ to 0.01 and the maximum number of iterations k_{max} to 20. The p-value of Lp norm is selected from the set $\{1,\ 1.5,\ 2,\ 2.5,\ 3\}$.

To better improve generalization performance and model reliability, we perform 5-fold cross-validation to obtain the optimal hyperparameters $c_1,\ c_2,\ c_3,\ c_4$. This means that during each training round, we divide the labeled and unlabeled samples into 5 groups, using 4 groups as training samples each time and the remaining set of labeled samples as test samples, this process is repeated five times. To simplify the process of training hyperparameters, we set $c_3 = c_1$, $c_4 = c_2$, $c_1,\ c_2$ are chosen from the set $\left\{2^i | i = -2,\ -1,\ \ldots,\ 1,\ 2\right\}$.

For the case of nonlinear case, we use two kernel functions, linear kernel function $K(x_i, x_j) = x_i^T x_j$ and RBF kernel $K(x_i, x_j) = \exp(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2})$, where σ are chosen from the set $\left\{2^i | i = -2, -1, \ldots, 1, 2\right\}$. After hyperparameter training is concluded, it is applied to test samples to determine the final accuracy. This procedure is iterated five times for each p-value. Subsequently, accuracy and standard deviation are computed. The optimal performance is marked in bold. If the accuracy is the same, the one with the smaller standard deviation is better.

In comparative experiments, for SVM classifiers, we use different suffixes to represent different kernel functions, where "1" represents the linear kernel and "2" represents the RBF kernel. For a single-view model, we use View 1, for a two-view model, we use View 1 and 2, and for others, all views are used by default.

5.1. Benchmark methods

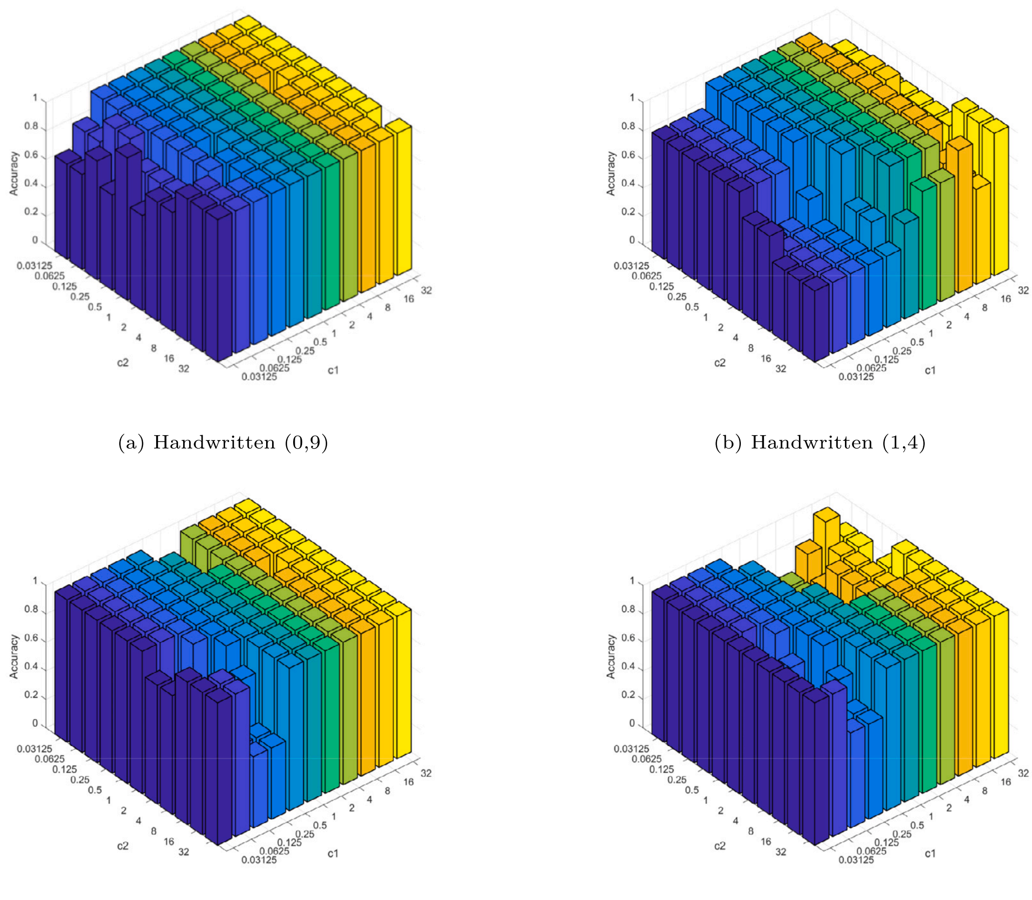
We compare MvHGLpLSTSVM with following methods:

- **MvLapLpLSTSVM** is the original version of our proposed method. It is a multi-view semi-supervised learning method that expresses manifold regularization by using Laplacian graphs of simple graph.
- **Lap-LpLSTSVM** [17] is a novel single-view semi-supervised classification method that leverages adjustability of the parameter p, incorporates Lp norm graph regularization to extract geometric information.
- **MvLapSVM** [28] is a method for multi-view semi-supervised learning that integrates manifold regularization and multi-view regularization into the standard SVM formulation, providing a natural extension from multi-view supervised learning to multi-view semi-supervised learning.
- **MVNPSVM** [31] is a novel multi-view supervised classification method based on the nonparallel support vector machine (NPSVM) that combines the principles of large margins and consensus, extending the advantages of NPSVM to the multi-view learning domain.
- **GMvLapSVM** [36] is a general multi-view semi-supervised learning method that takes into account the principles of consistency and complementarity. It is suitable for handling general multi-view semi-supervised classification problems and improves efficiency by reducing time complexity and introducing multi-manifold regularization. It is a general multi-view least squares version of MvLapSVM.
- **GMvLapTSVM** [36] is a novel multi-view semi-supervised classification method that utilizes multi-manifold regularization and combination weights to leverage information from multiple distinct views. It is a general multi-view least squares version of MvLapTSVM.
- **MVAR** [47] is a novel multi-view semi-supervised classification method, leveraging adaptive regression with $L_{2,1}$ matrix norm loss functions and linear weighted combination to efficiently handle large-scale datasets, offering robustness to low-quality views.
- **ERL-MVSC** [48] is a novel method for multi-view semi-supervised classification. It innovatively addresses the challenge of multi-view data with limited labels through a diverse, sparse, and consensus-driven embedding regularizer learning scheme.

Note: Except for MvLapSVM and MvNPSVM, which are two-view learning models, all other multi-view methods are general multi-view learning models, meaning that they can use data from all views.

5.2. Handwritten

Handwritten data comprises features extracted from Dutch Utilities utility maps, featuring handwritten digits ('0'–'9'), with 200 samples per class, totally 2000 samples. These samples have been digitized into binary images and represented by six feature (views) sets : (1) FOU:



(a) Handwritten (0,9) (b) Handwritten (1,4)

(c) Handwritten (2,6) (d) Handwritten (5,8)

Fig. 4. Sensitivity of c_1 and c_2 in different pairs of Handwritten.

Table 3
Classification accuracies and standard deviations (%) on handwritten dataset.

Method	(0,9)	(1,4)	(2,6)	(5,8)	(3,7)
MvHGLpLSTSVM1	98.75(0.88)	98.13(1.93)	97.50(0.13)	93.75(4.28)	94.75(3.55)
MvHGLpLSTSVM2	**99.75(0.34)**	**99.38(0.62)**	**99.75(0.34)**	**99.75(0.34)**	**99.25(0.52)**
MvLapLpLSTSVM1	97.00(2.36)	92.00(1.12)	96.37(2.77)	95.50(2.23)	93.50(5.28)
MvLapLpLSTSVM2	99.63(0.56)	94.88(3.86)	99.13(0.71)	99.13(0.56)	97.25(1.80)
Lap-LpLSTSVM1	82.50(8.89)	78.75(3.52)	93.50(4.38)	95.45(1.34)	74.50(3.49)
Lap-LpLSTSVM2	96.63(3.21)	83.00(3.17)	97.00(1.35)	95.38(2.24)	92.62(6.18)
MvLapSVM1	99.50(0.28)	89.00(4.97)	98.13(1.17)	84.25(5.44)	96.37(0.93)
MvLapSVM2	99.13(1.30)	90.00(2.90)	96.37(1.73)	98.87(0.68)	97.79(1.00)
MVNPSVM1	93.26(1.46)	73.47(6.54)	94.13(2.41)	87.45(8.34)	89.74(6.32)
MVNPSVM2	80.38(4.23)	51.50(1.80)	91.63(4.52)	64.50(2.91)	71.63(3.80)
GMvLapSVM1	99.39(0.74)	95.67(0.58)	99.11(0.83)	99.14(0.69)	96.75(2.98)
GMvLapSVM2	99.62(0.34)	96.36(0.53)	99.56(0.26)	98.19(0.47)	98.11(1.29)
GMvLapTSVM1	99.65(0.33)	97.19(1.88)	99.47(0.49)	98.00(1.20)	97.88(1.14)
GMvLapTSVM2	99.63(0.56)	96.88(1.47)	99.00(0.95)	98.75(0.88)	99.25(0.81)
MVAR	99.33(0.26)	98.03(0.80)	99.17(0.24)	99.13(0.63)	96.70(0.85)
ERL-MVSC	99.62(0.24)	96.36(1.24)	99.40(0.35)	98.80(0.60)	98.75(0.24)

76 Fourier coefficients of the character shapes; (2) FAC: 216 profile correlations; (3) KAR: 64 Karhunen–Love coefficients; (4) PIX: 240 pixel averages in 2 x 3 windows; (5) ZER: 47 Zernike moments; (6) MOR: 6 morphological features.

As shown in Table 3, in experiments on randomly selecting 5 pairs from the handwritten dataset for binary classification which include of all '0–9', our method obtained the best performance by using the RBF kernel function. Compared with the single-view semi-supervised learning method Lap-LpLSTSVM, our method has significant improvements in both linear kernel function and RBF kernel function, reflecting the excellent performance of multi-view learning. Compared with the multi-view supervised method MvNPSVM, it can be seen that our method can better utilize unlabeled samples to assist classification tasks when there are fewer labeled samples, further demonstrating the effectiveness of semi-supervised learning. Compared with other multi-view semi-supervised learning methods, the experimental results show that our method can utilize multi-view data more effectively.

The accuracy versus c_1 and c_2 for handwritten datasets is shown in Fig. 4. We can see that for Handwritten (0, 9) and Handwritten (2, 6), accuracy is not sensitive to the selection of c_1 and c_2. For Handwritten (1, 4), it often obtains better results when $1 \leq c_1 \leq 16$. For Handwritten (5, 8), when $c_2 \geq 0.5$, better results are often obtained. From the above

Table 4
Classification accuracies and standard deviations (%) on NUS dataset.

Method	(4,10)	(9,12)	(8,10)	(6,12)	(5,7)
MvHGLpLSTSVM1	78.50(2.82)	75.38(1.14)	84.75(5.22)	74.38(2.61)	76.63(3.08)
MvHGLpLSTSVM2	76.88(5.47)	**78.38(3.02)**	**88.25(2.52)**	**77.18(2.35)**	74.63(4.11)
MvLapLpLSTSVM1	80.63(5.21)	69.00(5.96)	82.75(3.89)	72.88(3.32)	68.63(6.99)
MvLapLpLSTSVM2	**82.50(4.17)**	75.75(2.04)	87.87(2.24)	76.63(3.08)	76.25(3.34)
Lap-LpLSTSVM1	76.13(8.09)	66.75(6.08)	78.25(3.07)	64.87(4.56)	67.50(3.06)
Lap-LpLSTSVM2	78.38(6.71)	63.13(7.94)	81.00(5.30)	60.50(3.99)	66.38(4.60)
MvLapSVM1	76.50(1.69)	66.50(4.16)	85.88(3.02)	66.63(3.41)	73.00(2.91)
MvLapSVM2	80.50(2.31)	69.75(4.69)	85.50(1.95)	71.63(2.57)	72.00(2.81)
MVNPSVM1	44.25(28.39)	52.50(23.95)	61.63(23.99)	55.31(18.52)	62.50(21.41)
MVNPSVM2	76.13(6.77)	62.63(2.48)	83.13(0.77)	60.12(6.03)	69.13(4.45)
GMvLapSVM1	67.38(6.01)	73.48(3.44)	86.59(0.40)	68.19(2.13)	71.88(1.25)
GMvLapSVM2	73.38(2.71)	73.45(2.43)	82.64(2.03)	72.53(2.62)	68.22(1.92)
GMvLapTSVM1	77.93(5.26)	62.26(3.63)	84.25(0.94)	72.56(1.70)	73.13(2.17)
GMvLapTSVM2	77.30(2.44)	71.10(3.25)	82.26(6.04)	66.09(1.04)	69.29(1.20)
MVAR	76.88(4.08)	71.68(4.17)	82.74(4.88)	70.28(3.37)	70.87(5.88)
ERL-MVSC	80.60(2.60)	77.77(1.83)	87.77(1.22)	75.27(2.98)	**77.83(2.12)**

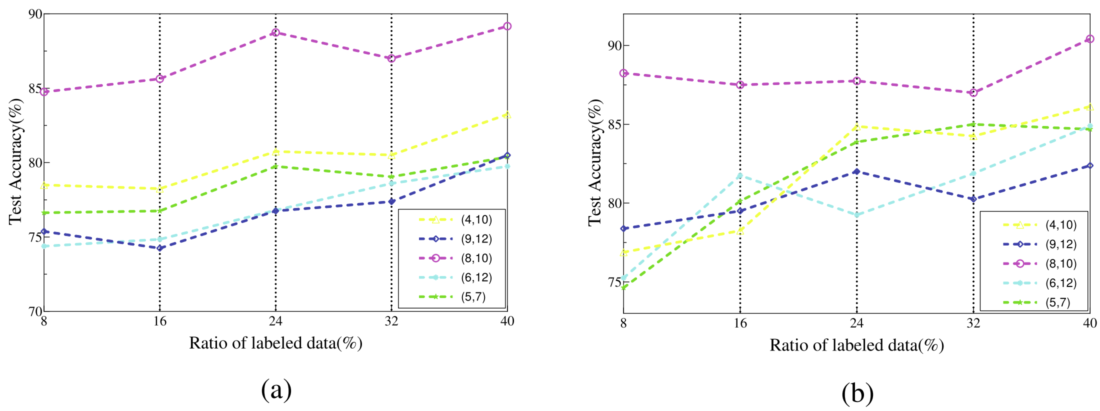


Fig. 5. The effect of different ratios labeled data of NUS on MvHGLpLSTSVM with (a) Linear kernel and (b) RBF kernel.

analysis and Fig. 4, it can be concluded that on the handwritten dataset, the recommended values for c_1 and c_2 on are between 0.25 and 4.

5.3. NUS

NUS is a network image dataset created by the Media Search Laboratory of the National University of Singapore. Six low-level features (viewed as views) were extracted from these images: (1) 64-D color histogram; (2) 144-D color correlogram; (3) 73-D edge direction histogram; (4) 128-D wavelet texture; (5) 225-D block-wise color moments; (6) 500-D bag of words based on SIFT descriptions. In this experiment, we used 6 views, 2400 samples, and 12 classes. Randomly select several pairs for binary classification experiments.

As shown in Table 4, our method still obtained good performance on the image dataset NUS. Compared with the single-view method, our method achieved 15.23%, 7.25%, and 16.68% improvements on NUS (9,12), NUS (8,10), and NUS (6,12). Compared with the benchmark method MvLapSVM, it achieved improvements of 8.63%, 2.75%, and 5.55% in the above three experiments, respectively.

To verify the effectiveness of semi-supervised learning. On the NUS dataset, the linear kernel and RBF kernel use labeled training sets with different ratios (8%–40%), and testing sets with a ratio of 40%. The hyperparameters are trained using a 5-fold cross-validation method, and the final performance is shown in Fig. 5. The results indicate that this method effectively utilizes unlabeled sample information and has strong robustness, without causing overfitting or underfitting.

5.4. BBCSport

BBCSport is a synthetic multi-view text dataset that divides news articles into relevant text fragments. It comprises 24 documents from the BBC Sport website, containing sports news articles in five categories from 2004 to 2005. The categories include athletics, cricket, football, rugby, and tennis. The dataset used in this experiment consists of 2 views, and the dimensions are: (1) 3183; (2) 3203.

As shown in Table 5, our method obtained excellent results on the BBCSport dataset with a sample size (544) much smaller than the feature number (6386). As a dataset with strong sparsity, multi-view learning method such as MvNPSVM perform poorly. On the one hand, it is related to the sparsity of the dataset, and on the other hand, as an SVM-based classifier, when there are few labeled samples and the selection of support vector points is not ideal, it often achieves poor classification results. In addition, we can see from Table 5 that the effect of using the linear kernel function is generally better than the RBF kernel function. It shows that the choice of kernel function is affected by the data set to a certain extent.

5.5. Caltech-101

Caltech-101 is an image database containing 101 different types of items, mainly used for research and experiments in the fields of pattern recognition and computer vision. The image collection and annotation of this database were completed by Li Fei-Fei, Rob Ferguson, Pietro Perona, and others from the California Institute of Technology. The Caltech-101-7 dataset contains 7 classes of image samples from 101 different object categories, with a total of 1474 samples, each with a different number of samples. All images are represented by 6 views, and the dimensions of each view are: (1) 48; (2) 40; (3) 254; (4) 1984; (5) 512; (6) 928. In order to simplify the calculation, we apply principal component analysis (PCA) [49] to reduce the dimensionality of views 4–6, and the processed dimensions are (4) 388; (5) 123; (6) 207.

Table 5
Classification accuracies and standard deviations (%) on BBCSport dataset.

Method	(3,5)	(2,4)	(1,2)	(4,5)
MvHGLpLSTSVM1	**96.24(0.83)**	**97.17(0.97)**	**98.81(1.95)**	**99.46(0.74)**
MvHGLpLSTSVM2	95.25(4.39)	94.57(4.07)	97.61(1.33)	97.84(2.26)
MvLapLpLSTSVM1	96.24(1.29)	96.52(1.61)	96.12(4.30)	98.38(2.22)
MvLapLpLSTSVM2	94.46(3.18)	95.65(2.03)	93.13(2.50)	98.92(1.48)
Lap-LpLSTSVM1	87.65(8.01)	95.38(5.57)	84.55(12.65)	91.08(9.96)
Lap-LpLSTSVM2	88.04(8.70)	92.97(2.97)	85.45(10.03)	87.84(7.99)
MvLapSVM1	85.32(6.16)	94.57(0.76)	94.63(6.47)	94.59(3.70)
MvLapSVM2	88.37(8.22)	93.26(6.22)	94.19(7.54)	88.89(3.83)
MVNPSVM1	78.61(7.51)	63.48(19.25)	73.88(18.02)	83.78(17.09)
MVNPSVM2	76.24(0.00)	54.35(0.00)	73.73(17.79)	67.57(0.00)
GMvLapSVM1	83.96(4.92)	97.39(1.65)	98.51(0.82)	87.30(4.23)
GMvLapSVM2	76.24(0.00)	93.48(4.28)	81.49(14.06)	69.73(4.83)
GMvLapTSVM1	87.13(4.08)	90.00(3.21)	92.84(5.11)	86.22(6.78)
GMvLapTSVM2	95.45(2.85)	96.96(1.79)	95.22(3.72)	91.89(6.89)
MVAR	80.45(5.35)	82.22(17.24)	81.63(13.52)	79.09(10.73)
ERL-MVSC	93.25(3.12)	95.90(1.53)	98.56(0.24)	94.74(3.20)

Table 6
Classification accuracies and standard deviations (%) on Caltech-101 dataset with Class 1 and Class 2.

Method	MvHGLpLSTSVM1	MvHGLpLSTSVM2	MvLapLpLSTSVM1	MvLapLpLSTSVM2
Accuracy	96.55(1.06)	**99.03(0.53)**	95.33(1.11)	98.90(0.77)
Method	Lap-LpLSTSVM1	Lap-LpLSTSVM2	MvLapSVM2	MvLapSVM2
Accuracy	95.86(1.37)	94.52(0.82)	90.23(2.28)	93.59(1.12)
Method	MvNPSVM1	MvNPSVM2	MVAR	ERL-MVAR
Accuracy	98.66(0.34)	92.27(1.34)	98.64(0.72)	98.57(0.35)
Method	GMvLapSVM1	GMvLapSVM2	GMvLapTSVM1	GMvLapTSVM2
Accuracy	96.38(0.97)	95.08(0.74)	96.75(1.21)	96.14(1.21)

From Table 2, it can be seen that class 1 and class 2 account for the majority of the dataset. Therefore, we selected these two classes for comparison, and the results are shown in Table 6. Firstly, most methods obtain good classification results due to the sufficient sample size, which makes it easier for most methods to find support vector. Therefore, both single-view and multi-view models have good performance, and the best classification performance of 99.03% is obtained by MvHGLpLSTSVM using RBF kernel function. Compared to MvLapSVM, which utilizes only View 1 and View 2, our method achieved a 5.44% improvement. Other multi-view methods also show improvements compared to the two-view methods. Therefore, it can be concluded that effectively utilizing the information of other views can better assist binary classification.

5.6. Cora and WebKB

Cora dataset comprises 2708 scientific publications categorized into 7 classes. The dataset is represented by four views, each with the following dimensions: (1) 2708; (2) 1433; (3) 2706; (4) 2706. Each publication in the dataset is characterized by a binary word vector, denoted by 0/1 values, indicating the absence or presence of the corresponding word in the dictionary. WebKB dataset consists of 203 web pages categorized into 4 classes. Each web page is characterized by its page content, the anchor text of hyperlinks, and the text in its title. The dataset is represented by three views, with the following dimensions: (1) 1703; (2) 230; (3) 230. Same as the Cora dataset, WebKB consists of 0/1 values.

On these two datasets, we conducted experiments with three pairs selected for each, and the results are presented in Table 7. It is evident that whether a linear kernel or an RBF kernel is employed, methods utilizing hypergraph generally outperform those without, underscoring the effectiveness of hypergraph learning. Notably, for the multi-view supervised learning method MVNPSVM, semi-supervised learning leveraging unlabeled samples proves to be more advantageous in meeting the classification task. In a series of comparative experiments on the Cora dataset, we observe that the single-view Lap-LpLSTSVM achieved the most favorable results. This suggests that in certain experiments, multi-view data may introduce some noise and redundancy that impact the final results. However, on the whole, our method consistently achieves superior performance.

5.7. Discussion

In the binary classification experiments conducted on the above six datasets, our proposed method, MvHGLpLSTSVM, outperformed others and achieved outstanding results. By incorporating a manifold regularization term composed of Lp norm and hypergraph Laplacian matrix for each view, and the classification performance is improved by integrating information from different view under consistency constraints. To validate the effectiveness of our proposed method, we conducted several analysis experiments. The results of the paired t-test are presented in Table 8. Figs. 6 and 7 record the convergence and the relative running time of the model . Based on these tables and figures, we can draw the following conclusions.

1. MvHGLpLSTSVM dominates in the majority of $h = 1$ cases (13/18) in NUS, BBCSport, and Caltech-101, indicating the most significant performance improvement on these three datasets. Although the improvement on the common dataset Handwritten and the small dataset WebKB is modest, overall, our proposed method proves to be effective.
2. Fig. 6 shows the convergence of MvHGLpLSTSVM across the six datasets. We observe a rapid decrease in iteration difference, indicating that our proposed algorithm is efficient, converging within several iterations.
3. In Fig. 7, MvHGLpLSTSVM takes approximately twice as much time as MvLapLpLSTSSVM due to the need to construct hypergraphs. The complexity of hypergraph calculation is related to the sample size and the number of nearest neighbors considered. On the smaller datasets BBCSport and WebKB, the running time of MvHGLpLSTSVM is comparable to other comparison methods. Particularly, the time required for ERL-MVSC on BBCSport exceeds others significantly. Usually, as a high-order graph, it is acceptable to significantly improve performance by spending extra time to construct high-quality graph.

Table 7
Classification accuracies and standard deviations (%) on Cora and WebKB.

Method	Cora			WebKB		
	(1,3)	(4,6)	(1,6)	(1,2)	(1,3)	(2,3)
MvHGLpLSTSVM1	83.29(4.90)	72.46(4.82)	74.21(3.95)	74.94(8.68)	81.57(6.29)	**78.26(5.32)**
MvHGLpLSTSVM2	**86.22(3.38)**	**88.96(2.63)**	80.20(2.38)	**77.06(1.32)**	84.71(0.88)	68.70(3.92)
MvLapLpLSTSVM1	83.29(5.32)	69.67(5.61)	75.47(6.87)	76.47(3.60)	83.53(1.07)	65.80(7.36)
MvLapLpLSTSVM2	84.64(2.33)	86.17(2.96)	80.53(3.60)	75.29(1.61)	82.75(1.64)	70.43(9.75)
LapLpLSTSVM1	76.43(1.61)	74.38(1.99)	79.76(2.77)	62.29(7.11)	60.78(11.00)	55.07(4.23)
LapLpLSTSVM2	82.85(5.64)	82.07(2.34)	**84.01(2.53)**	76.00(3.11)	82.35(5.00)	71.30(12.78)
MvLapSVM1	79.37(0.63)	70.89(9.18)	73.51(10.79)	75.43(9.39)	82.40(4.98)	66.96(7.70)
MvLapSVM2	79.95(2.05)	77.51(9.33)	78.95(8.82)	70.86(13.16)	82.80(2.68)	70.72(8.41)
MvNPSVM1	84.35(0.86)	68.78(3.48)	71.86(5.14)	73.53(2.94)	80.78(3.77)	67.83(5.16)
MvNPSVM2	78.99(0.00)	76.91(9.46)	66.84(0.41)	68.82(9.44)	75.49(11.48)	61.16(13.99)
GMvLapSVM1	75.07(4.77)	78.28(5.65)	77.09(3.67)	67.06(4.83)	73.33(6.14)	73.04(5.85)
GMvLapSVM2	81.35(2.07)	75.61(4.08)	80.08(1.20)	77.06(2.46)	**85.10(1.07)**	64.06(1.59)
GMvLapTSVM1	80.77(3.19)	75.31(3.14)	76.92(5.28)	74.12(11.65)	77.65(11.89)	70.47(5.85)
GMvLapSVM2	80.68(0.59)	59.70(5.13)	71.66(2.31)	76.47(3.60)	83.14(2.63)	66.09(3.49)
MVAR	81.30(1.22)	86.67(10.62)	81.28(6.17)	74.57(7.60)	82.73(4.47)	75.78(12.57)
ERL-MVAR	81.53(2.52)	71.53(7.50)	77.86(9.30)	75.80(2.56)	83.05(1.20)	76.88(8.81)

Table 8
Paired t-test results of the accuracy and comparison algorithms of MvHGLpLSTSVM on all datasets.

Method	Handwritten		NUS		BBCSport		Caltech-101		Cora		WebKB	
	h	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*	*h*	*p*
MvLapLpLSTSVM	0	6.81E−01	0	8.10E−01	0	4.86E−01	0	7.78E−01	0	1.54E−01	0	1.54E−01
Lap-LpLSTSVM	0	6.20E−02	1	2.45E−02	1	1.19E−02	1	6.77E−06	1	2.35E−03	0	2.94E−01
MvLapSVM	0	3.28E−01	0	8.93E−02	1	2.04E−02	1	9.49E−06	1	3.74E−02	1	2.70E−02
MvNPSVM	1	1.81E−17	1	4.74E−04	1	8.84E−08	1	3.81E−20	0	0.00E−00	1	7.55E−03
GMvLapSVM	0	5.20E−01	1	3.18E−13	1	1.28E−10	1	1.27E−18	1	2.75E−04	0	1.79E−01
GMvLapTSVM	0	6.81E−01	1	4.88E−13	1	2.57E−10	1	1.74E−15	1	7.30E−05	0	5.79E−02
MVAR	0	5.05E−01	1	2.90E−02	1	2.80E−03	1	4.67E−01	0	4.34E−01	0	8.74E−01
ERL-MVSC	0	5.07E−01	0	7.12E−01	1	1.23E−02	1	1.46E−01	1	1.20E−02	1	9.63E−01

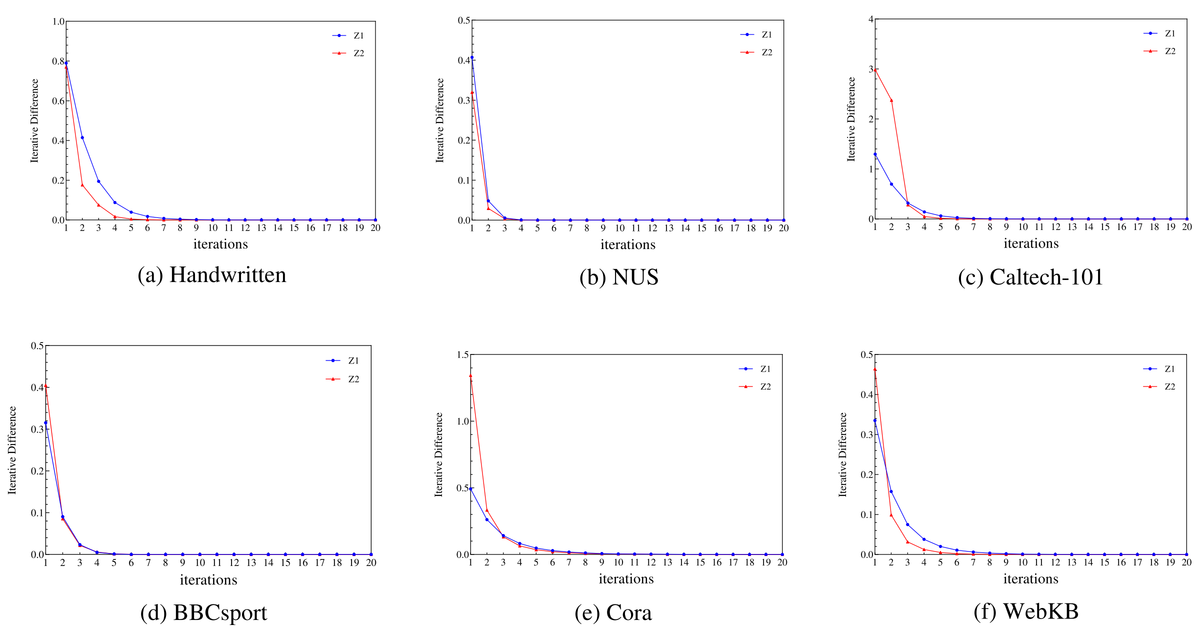


Fig. 6. The convergence curves of Z_1 and Z_2 on all datasets.

Our method is superior to other state-of-the-art methods for several reasons: (1) Hypergraph, as high-order graph, can extend graph to high dimensional and more complete nonlinear spaces, and better explore high-order correlation among data, thus obtaining better performance in practice. (2) The p value in the Lp norm is adjustable. The Lp norm graph regularization term constructed through eigenvalue decomposition can utilize the geometric information embedded in the data and effectively reduce the impact of outliers and noise. For different datasets, appropriate p value can be selected to achieve the desired performance and improve generalization capabilities. (3) A general multi-view framework is proposed, enabling the utilization of information from all views. Experiments demonstrate that incorporating more view information can enhance the model's performance.

6. Conclusion

In this paper, we propose a novel multi-view semi-supervised learning method, multi-view hypergraph regularized Lp norm least squares twin support vector machines. Extending the semi-supervised model constructed with Lp norm and hypergraph Laplacian to the MVL, we

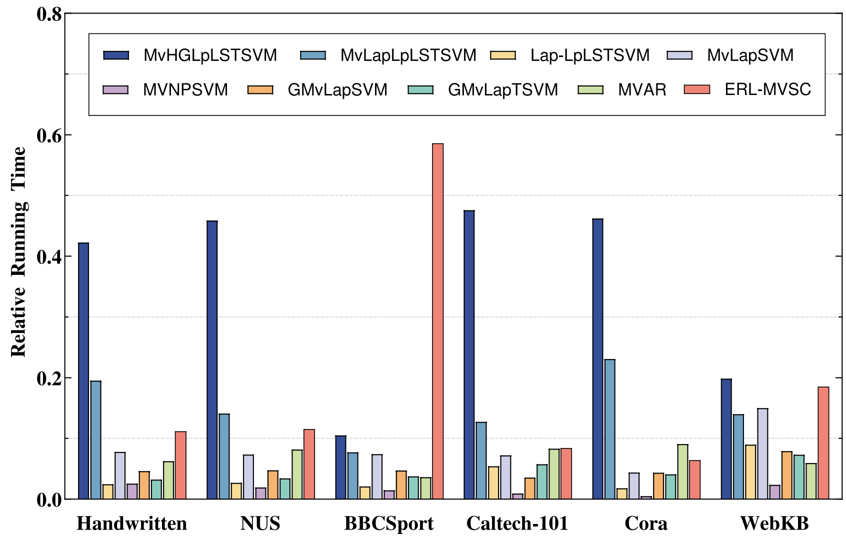


Fig. 7. Relative running time of all multi-view methods.

propose a general framework for multi-view semi-supervised learning. We compare it with the single-view semi-supervised method Lap-LpLSTSVM and the multi-view supervised method MvNPSVM. Furthermore, we conduct comparisons with other state-of-the-art multi-view semi-supervised methods, and experimental results demonstrate the effectiveness of the proposed method. However, challenges arise when dealing with large-scale datasets, such as the complexity of constructing hypergraphs and the time-consuming matrix inversion process. Furthermore, performance may degrade in the presence of redundancy and significant noise in the data from other views.

In future work, we will analyze its bound by using the statistical analysis theory, investigate the impact of high-order graph on multi-view SVM, and combine dimensionality reduction techniques with our method. Additionally, we have noted some recent advancements in matrix inversion methods [50,51] and improvements and acceleration algorithms related to LapSVM [52–54]. In subsequent work, we will consider incorporating these new technologies to accelerate model's solving, enabling better handling of large-scale data. In the field of MVL, we will utilize the complementary information between views, combine with weight information, and adaptively allocate weights between views to alleviate the effects of view redundancy and noise. In addition, our method is suitable for binary classification. We will adopt either the one-vs-one or one-vs-rest strategy to extend it to multiclass classification.

CRediT authorship contribution statement

Junqi Lu: Writing – original draft, Visualization, Validation, Software, Resources, Data curation, Conceptualization. **Xijiong Xie:** Writing – review & editing, Supervision, Methodology. **Yujie Xiong:** Writing – review & editing.

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 National Natural Science Foundation of China (No. 61906101). It is also supported by the Ningbo Municipal Natural Science Foundation of China (No. 2023J115).

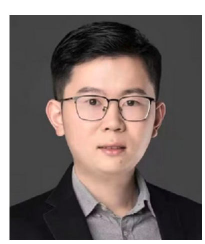
References

[1] C. Cortes, V. Vapnik, Support vector machine, Mach. Learn. 20 (3) (1995) 273–297.
[2] T. Zhang, F. Oles, The value of unlabeled data for classification problems, in: P. Langley (Ed.), Proceedings of the Seventeenth International Conference on Machine Learning, Vol. 20, Citeseer, 2000.
[3] O. Chapelle, B. Scholkopf, A. Zien, Semi-Supervised Learning. 2006, Vol. 2, The MIT Press, Cambridge, Massachusettes, 2006, View Article.
[4] K. Bennett, A. Demiriz, Semi-supervised support vector machines, Adv. Neural Inf. Process. Syst. 11 (1998).
[5] T. Joachims, et al., Transductive inference for text classification using support vector machines, in: Icml, Vol. 99, 1999, pp. 200–209.
[6] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (11) (2006).
[7] Z. Qi, Y. Tian, Y. Shi, Laplacian twin support vector machine for semi-supervised classification, Neural Netw. 35 (2012) 46–53.
[8] J. Tan, L. Zhen, N. Deng, Z. Zhang, Laplacian p-norm proximal support vector machine for semi-supervised classification, Neurocomputing 144 (2014) 151–158.
[9] G. Fung, O.L. Mangasarian, Semi-superyised support vector machines for unlabeled data classification, Optim. Methods Softw. 15 (1) (2001) 29–44.
[10] O. Chapelle, V. Sindhwani, S.S. Keerthi, Optimization techniques for semi-supervised support vector machines, J. Mach. Learn. Res. 9 (2) (2008).
[11] T. De Bie, N. Cristianini, Semi-supervised learning using semi-definite programming, 2006.
[12] F. Gieseke, A. Airola, T. Pahikkala, O. Kramer, Fast and simple gradient-based optimization for semi-supervised support vector machines, Neurocomputing 123 (2014) 23–32.
[13] H.M. Le, H.A. Le Thi, M.C. Nguyen, Sparse semi-supervised support vector machines by DC programming and DCA, Neurocomputing 153 (2015) 62–76.
[14] Y.-F. Li, Z.-H. Zhou, Towards making unlabeled data never hurt, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2014) 175–188.
[15] D. Zhang, L. Jiao, X. Bai, S. Wang, B. Hou, A robust semi-supervised SVM via ensemble learning, Appl. Soft Comput. 65 (2018) 632–643.
[16] W.-J. Chen, Y.-H. Shao, N.-Y. Deng, Z.-L. Feng, Laplacian least squares twin support vector machine for semi-supervised classification, Neurocomputing 145 (2014) 465–476.
[17] X. Xie, F. Sun, J. Qian, L. Guo, R. Zhang, X. Ye, Z. Wang, Laplacian Lp norm least squares twin support vector machine, Pattern Recognit. 136 (2023) 109192.
[18] Y. Sun, S. Ding, L. Guo, Z. Zhang, Hypergraph regularized semi-supervised support vector machine, Inform. Sci. 591 (2022) 400–421.
[19] S. Sun, W. Dong, Q. Liu, Multi-view representation learning with deep gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2020) 4453–4468.

[20] J. Yin, S. Sun, Multiview uncorrelated locality preserving projection, IEEE Trans. Neural Netw. Learn. Syst. 31 (9) (2019) 3442–3455.
[21] X. Xie, S. Sun, Multi-view support vector machines with the consensus and complementarity information, IEEE Trans. Knowl. Data Eng. 32 (12) (2019) 2401–2413.
[22] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, L. Wang, Learning a joint affinity graph for multiview subspace clustering, IEEE Trans. Multimed. 21 (7) (2018) 1724–1736.
[23] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, 1998, pp. 92–100.
[24] I. Muslea, S. Minton, C.A. Knoblock, Active learning with multiple views, J. Artificial Intelligence Res. 27 (2006) 203–233.
[25] P. Yang, W. Gao, Information-theoretic multi-view domain adaptation: A theoretical and empirical study, J. Artificial Intelligence Res. 49 (2014) 501–525.
[26] D.R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Comput. 16 (12) (2004) 2639–2664.
[27] J. Farquhar, D. Hardoon, H. Meng, J. Shawe-Taylor, S. Szedmak, Two view learning: SVM-2K, theory and practice, Adv. Neural Inf. Process. Syst. 18 (2005).
[28] S. Sun, Multi-view Laplacian support vector machines, in: Advanced Data Mining and Applications: 7th International Conference, ADMA 2011, Beijing, China, December 17-19, 2011, Proceedings, Part II 7, Springer, 2011, pp. 209–222.
[29] X. Xie, S. Sun, Multi-view twin support vector machines, Intell. Data Anal. 19 (4) (2015) 701–712.
[30] X. Xie, S. Sun, Multi-view Laplacian twin support vector machines, Appl. Intell. 41 (2014) 1059–1068.
[31] J. Tang, D. Li, Y. Tian, D. Liu, Multi-view learning based on nonparallel support vector machine, Knowl.-Based Syst. 158 (2018) 94–108.
[32] J. Tang, Y. Tian, P. Zhang, X. Liu, Multiview privileged support vector machines, IEEE Trans. Neural Netw. Learn. Syst. 29 (8) (2017) 3463–3477.
[33] S. Sun, X. Xie, C. Dong, Multiview learning with generalized eigenvalue proximal support vector machines, IEEE Trans. Cybern. 49 (2) (2018) 688–697.
[34] R. Xu, H. Wang, Multi-view learning with privileged weighted twin support vector machine, Expert Syst. Appl. 206 (2022) 117787.
[35] X. Xie, Multi-view semi-supervised least squares twin support vector machines with manifold-preserving graph reduction, Int. J. Mach. Learn. Cybern. 11 (2020) 2489–2499.
[36] X. Xie, S. Sun, General multi-view semi-supervised least squares support vector machines with multi-manifold regularization, Inf. Fusion 62 (2020) 63–72.
[37] Q. Ye, P. Huang, Z. Zhang, Y. Zheng, L. Fu, W. Yang, Multiview learning with robust double-sided twin SVM, IEEE Trans. Cybern. 52 (12) (2021) 12745–12758.
[38] X. Xie, Y. Li, S. Sun, Deep multi-view multiclass twin support vector machines, Inf. Fusion 91 (2023) 80–92.
[39] J.A. Suykens, J. Vandewalle, Least squares support vector machine classifiers, Neural Process. Lett. 9 (1999) 293–300.
[40] N. Kwak, Principal component analysis based on L1-norm maximization, IEEE Trans. Pattern Anal. Mach. Intell. 30 (9) (2008) 1672–1680.
[41] C.-N. Li, Y.-H. Shao, N.-Y. Deng, Robust L1-norm non-parallel proximal support vector machine, Optimization 65 (1) (2016) 169–183.
[42] H. Yan, Q. Ye, T. Zhang, D.-J. Yu, Y. Xu, L1-norm GEPSVM classifier based on an effective iterative algorithm for classification, Neural Process. Lett. 48 (2018) 273–298.
[43] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: Clustering, classification, and embedding, Adv. Neural Inf. Process. Syst. 19 (2006).
[44] S. Huang, M. Elhoseiny, A. Elgammal, D. Yang, Learning hypergraph-regularized attribute predictors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 409–417.
[45] L. Gómez-Chova, G. Camps-Valls, J. Munoz-Mari, J. Calpe, Semisupervised image classification with Laplacian support vector machines, IEEE Geosci. Remote Sens. Lett. 5 (3) (2008) 336–340.
[46] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, C. Zou, Hypergraph learning: Methods and practices, IEEE Trans. Pattern Anal. Mach. Intell. 44 (5) (2020) 2548–2566.
[47] H. Tao, C. Hou, F. Nie, J. Zhu, D. Yi, Scalable multi-view semi-supervised classification via adaptive regression, IEEE Trans. Image Process. 26 (9) (2017) 4283–4296.
[48] A. Huang, Z. Wang, Y. Zheng, T. Zhao, C.-W. Lin, Embedding regularizer learning for multi-view semi-supervised classification, IEEE Trans. Image Process. 30 (2021) 6997–7011.
[49] R. Bro, A.K. Smilde, Principal component analysis, Anal. Methods 6 (9) (2014) 2812–2831.
[50] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F.J. R Ruiz, J. Schrittwieser, G. Swirszcz, et al., Discovering faster matrix multiplication algorithms with reinforcement learning, Nature 610 (7930) (2022) 47–53.
[51] J. Liu, M. Liu, J.-P. Liu, Z. Ye, Y. Wang, Y. Alexeev, J. Eisert, L. Jiang, Towards provably efficient quantum algorithms for large-scale machine-learning models, Nature Commun. 15 (1) (2024) 434.
[52] Z. Yang, Y. Xu, Laplacian twin parametric-margin support vector machine for semi-supervised classification, Neurocomputing 171 (2016) 325–334.
[53] Z. Yang, Y. Xu, A safe screening rule for Laplacian support vector machine, Eng. Appl. Artif. Intell. 67 (2018) 309–316.
[54] Z. Yang, Y. Xu, A safe sample screening rule for Laplacian twin parametric-margin support vector machine, Pattern Recognit. 84 (2018) 1–12.

Junqi Lu is a master candidate with the School of Information Science and Engineering, Ningbo University. His research interests include support vector machines, multi-view learning, and semi-supervised learning.

Xijiong Xie received the Ph.D. degree from the Pattern Recognition and Machine Learning Research Group, Department of Computer Science and Technology, East China Normal University, in 2016. He is currently an Associate Professor with the Faculty of Electrical Engineering and Computer Science, Ningbo University China. He has over 30 publications at peer-reviewed journals and conferences in his research areas, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS), IEEE TRANSACTIONS ON CYBERNETICS, Expert systems with application, Pattern Recognition, Neurocomputing, and Information Fusion. His research interests include kernel methods, support vector machines, multi-view learning, and deep learning.



Yujie Xiong received his Ph.D. in Computer Science from the East China Normal University in June 2018. He is currently an Associate Professor in the School of Electronic and Electrical Engineering at Shanghai University of Engineering Science, China. His research interests include biometric, document image analysis, and knowledge graph based application, where he has published more than 30 academic publications.