Semi-supervised Multi-class Classification Methods Based on Laplacian Vector Projection

Yangtao Xue and Li Zhang(✉)

School of Computer Science and Technology,
Soochow University, Suzhou 215006, China
20184027008@stu.suda.edu.cn, zhangliml@suda.edu.cn

Abstract. Laplacian pair-weight vector projection (LapPVP) is a binary classifier for semi-supervised learning, which seeks a pair of projection vectors only for two-class data. This paper extends LapPVP to semi-supervised multi-class classification tasks and proposes two novel semi-supervised multi-class methods, named one-versus-one LapPVP (OVO-LapPVP) and one-versus-rest LapPVP (OVR-LapPVP). By using the strategy of "one-versus-one", OVO-LapPVP decomposes a semi-supervised multi-class classification task into multiple binary problems that can be directly solved by multiple LapPVPs. Considering the concept of "one-versus-rest", OVR-LapPVP is designed for generating multiple hyperplanes for multiple classes, one for each class. The above proposed semi-supervised multi-class classification methods both consider the discriminative information of labeled data and graph structure of unlabeled data. Experiments are conducted on nine UCI datasets to display the classification performance for multi-class data. Compared with other popular semi-supervised multi-class classification methods based on manifold regularization, the proposed semi-supervised multi-class classification methods hold the advantage of LaPVP and have better performance.

Keywords: Multi-class classification · Semi-supervised learning · Manifold regularization · Vector Projection

1 Introduction

Twin support vector machine (TSVM) has become one of the popular binary classifiers because it can greatly reduce the computational complexity of support vector machine (SVM) [5]. TSVM obtains two nonparallel hyperplanes by two smaller-sized and related SVM-type problems and is a supervised binary classifier that require a large amount of labeled data. However, it is difficult for TSVM to deal with multi-classification problems. In order to inherit the advantages of TSVM, some multi-class versions of TSVM have been proposed for wider applications [9,10].

Inspired by the manifold regularization, Laplacian TSVM (LapTSVM) was proposed for semi-supervised learning and become a useful extension of TSVM [8]. Similar to LapTSVM, Laplacian twin parametric-margin SVM

H. Zhang et al. (Eds.): NCAA 2023, CCIS 1869, pp. 160–174, 2023.
https://doi.org/10.1007/978-981-99-5844-3_12

(LapTPMSVM) [12] and Laplacian least squares TSVM (LapLSTSVM) [1] are the semi-supervised versions of twin parametric-margin SVM [7] and least squares TSVM [6], respectively, which are variants of TSVM. The above three semi-supervised methods all aim to generate two nonparallel hyperplanes so that each hyperplane is closer to its own class as far as possible from the other. Laplacian pair-weight vector projection (LapPVP) was also motivated by the idea of nonparallel hyperplanes [11]. LapPVP is an excellent binary classifier compared with other nonparallel hyperplanes methods. Furthermore, LapPVP integrates the between-class scatter, the within-class scatter, and the Laplacian regularization together for semi-supervised binary classification. Each class data provides the class-specific information by computing the between-class and within-class scatters, which enhances the power of discriminative representation. The Laplacian regularization provides the graph structure of labeled and unlabeled data, which is the intrinsic geometrical structure of data.

Although these semi-supervised binary classifiers are promising, their multi-class versions have been rarely explored. To solve the semi-supervised multi-class classification problems, we extend the formulation of LapPVP to its multi-class versions and propose two novel semi-supervised multi-class methods, named one-versus-one LapPVP (OVO-LapPVP) and one-versus-rest LapPVP (OVR-LapPVP). Researchers have developed the ideas of "one-versus-one" (OVO) and "one-versus-rest" (OVR) to decompose a multi-class problem into multiple binary sub-problems [3,9]. Originally, LapPVP obtains a pair of projection vectors for two-class data. Both OVO-LapPVP and OVR-LapPVP solve a multi-class classification task using multiple LapPVPs, but differ in the number of LapPVPs. In specific, the multi-class versions of LapPVP have the following characteristics:

(1) OVO-LapPVP obtains $C(C-1)/2$ pairs of projection vectors for C-class data, whereas OVR-LapPVP achieves C nonparallel hyperplanes for C-class data, one for each class.
(2) The proposed multi-class classification methods are all solved by eigenvalue decomposition, which avoids finding solutions to complex quadratic programmings.
(3) Experimental results on UCI datasets demonstrate the effectiveness of OVO-LapPVP and OVR-LapPVP.

2 Semi-supervised Multi-class LapPVPs

In this section, we describe the proposed OVO-LapPVP and OVR-LapPVP detailly. Now, we consider a semi-supervised multi-class classification task. Let $\mathbf{X}_l = [\mathbf{x}_1, \cdots, \mathbf{x}_l]^T \in \mathbb{R}^{l \times m}$ and $\mathbf{X}_u = [\mathbf{x}_{l+1}, \cdots, \mathbf{x}_n]^T \in \mathbb{R}^{(n-l) \times m}$ be the labeled and unlabeled sample matrices, respectively, $\mathbf{x}_i \in \mathbb{R}^m$ is an m-dimensional sample, l and n are the numbers of labeled and total samples, respectively. Let y_i be the label of $\mathbf{x}_i$, where $y_i \in [1, 2, ..., C]$, and C is the number of classes. Let $\mathbf{X}_{l,c}$ be the labeled sample matrix belonging to the cth class. The total sample matrix can be denoted as $\mathbf{X} = [\mathbf{X}_{l,1}; \cdots; \mathbf{X}_{l,C}; \mathbf{X}_u] \in \mathbb{R}^{n \times m}$.

In LapPVP, we need to construct an adjacency graph using the K nearest neighbor method. The ith row and jth column element of the adjacency matrix $\mathbf{S}$ induced by the adjacency graph can be represented as

$$S_{ij} = \begin{cases} 1, & if \quad \mathbf{x}_i \in N_K(\mathbf{x}_j) \quad or \quad \mathbf{x}_j \in N_K(\mathbf{x}_i) \\ 0, & otherwise \end{cases}, \tag{1}$$

where $N_K(\mathbf{x}_i)$ is the set of K nearest neighbors of $\mathbf{x}_i$. Then we get the classical Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{S}$, where $\mathbf{D}$ is the diagonal matrix with $D_{ii} = \sum_j S_{ij}$.

2.1 OVO-LapPVP

The one-versus-one strategy is a popular technique that can easily extend binary classifiers to multi-class ones. By using the one-versus-one strategy, OVO-LapPVP is presented, which can generate $C(C-1)/2$ binary classifiers for C-class data. Given a C-class dataset, we obtain the set of class pairs $Z = \{z = (c_1, c_2) | c_1, c_2 = 1, 2, \cdots, C, c_1 \neq c_2\}$. To construct a binary classifier with class pair z, suppose class c_1 is the positive class, and class c_2 as the negative one. The corresponding LapPVP classifier is trained with labeled samples in both classes c_1 and c_2 and all unlabeled samples.

Let $\mathbf{X}_z = [\mathbf{X}_{l,c_1}; \mathbf{X}_{l,c_2}; \mathbf{X}_u] \in \mathbb{R}^{n_z \times m}$ with $z = (c_1, c_2)$, where n_z is the total number of samples in both classes c_1 and c_2 and without labels. The objective functions of OVO-LapPVP with respect to z can be expressed as follows:

$$\begin{aligned} &\max_{\mathbf{v}_{c_1}} \mathbf{v}_{c_1}^T \mathbf{H}_{c_1} \mathbf{v}_{c_1} - \rho_{c_1} \mathbf{v}_{c_1}^T \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z \mathbf{v}_{c_1} \\ &s.t. \quad \mathbf{v}_{c_1}^T \mathbf{v}_{c_1} = 1 \end{aligned} \tag{2}$$

and

$$\begin{aligned} &\max_{\mathbf{v}_{c_2}} \mathbf{v}_{c_2}^T \mathbf{H}_{c_2} \mathbf{v}_{c_2} - \rho_{c_2} \mathbf{v}_{c_2}^T \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z \mathbf{v}_{c_2} \\ &s.t. \quad \mathbf{v}_{c_2}^T \mathbf{v}_{c_2} = 1 \end{aligned}, \tag{3}$$

where $\mathbf{v}_{c_1} \in \mathbb{R}^m$ and $\mathbf{v}_{c_2} \in \mathbb{R}^m$ are projection vectors, ρ_{c_1} and ρ_{c_2} are regularization parameters that are greater than 0, the Laplacian matrix $\mathbf{L}_z \in \mathbb{R}^{n_z \times n_z}$ is computed based on $\mathbf{X}_z$, and the discriminative matrices $\mathbf{H}_{c_1} \in \mathbb{R}^{m \times m}$ and $\mathbf{H}_{c_2} \in \mathbb{R}^{m \times m}$ are defined as

$$\begin{aligned} \mathbf{H}_{c_1} = &\alpha_{c_1} \left(\mathbf{X}_z - \mathbf{e}_z \mathbf{u}_{c_1}^T\right)^T \left(\mathbf{X}_z - \mathbf{e}_z \mathbf{u}_{c_1}^T\right) \\ &- (1 - \alpha_{c_1})(\mathbf{X}_{c_1} - \mathbf{e}_{c_1} \mathbf{u}_{c_1}^T)^T (\mathbf{X}_{c_1} - \mathbf{e}_{c_1} \mathbf{u}_{c_1}^T) \end{aligned} \tag{4}$$

and

$$\begin{aligned} \mathbf{H}_{c_2} = &\alpha_{c_2} \left(\mathbf{X}_z - \mathbf{e}_z \mathbf{u}_{c_2}^T\right)^T \left(\mathbf{X}_z - \mathbf{e}_z \mathbf{u}_{c_2}^T\right) \\ &- (1 - \alpha_{c_2})(\mathbf{X}_{c_2} - \mathbf{e}_{c_2} \mathbf{u}_{c_2}^T)^T (\mathbf{X}_{c_2} - \mathbf{e}_{c_2} \mathbf{u}_{c_2}^T), \end{aligned} \tag{5}$$

where $\alpha_{c_1} \in [0, 1]$ and $\alpha_{c_2} \in [0, 1]$ are parameters to balance the between-class scatter matrix and the within-class one, class centers $\mathbf{u}_{c_j} = \frac{1}{|X_{l,c_j}|} \sum_{\mathbf{x}_i \in X_{l,c_j}} \mathbf{x}_i$

$(j = 1, 2)$, X_{l,c_j} is the set of labeled samples in the c_jth class, $\mathbf{e}_z \in \mathbb{R}^{n_z}$ and $\mathbf{e}_{c_j} \in \mathbb{R}^{|X_{l,c_j}|}$ $(j = 1, 2)$ are vectors of all ones.

The following theorems describe the solutions to the optimization problems (2) and (3).

Theorem 1. *The optimal solution* $\mathbf{v}_{c_1}$ *of the optimization problem (2) is the eigenvector corresponding to the maximum eigenvalue of the matrix* $(\mathbf{H}_{c_1} - \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z)$.

Proof. To find the solution to (2), we first generate the corresponding Lagrangian function with positive multipliers λ_{c_1}. That is

$$L(\mathbf{v}_{c_1}, \lambda_{c_1}) = \mathbf{v}_{c_1}^T \mathbf{H}_{c_1} \mathbf{v}_{c_1} - \rho_{c_1} \mathbf{v}_{c_1}^T \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z \mathbf{v}_{c_1} - \lambda_{c_1} (\mathbf{v}_{c_1}^T \mathbf{v}_{c_1} - 1). \tag{6}$$

Next, we derive the partial derivative of $L(\mathbf{v}_{c_1}, \lambda_{c_1})$ with respect to the primal variable $\mathbf{v}_{c_1}$, and then make it equal zero, which results in

$$\begin{aligned} &\frac{\partial L(\mathbf{v}_{c_1}, \lambda_{c_1})}{\partial \mathbf{v}_{c_1}} = \left(\mathbf{H}_{c_1} - \rho_{c_1} \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z\right) \mathbf{v}_{c_1} - \lambda_{c_1} \mathbf{v}_{c_1} = 0 \\ &\Rightarrow \left(\mathbf{H}_{c_1} - \rho_{c_1} \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z\right) \mathbf{v}_{c_1} = \lambda_{c_1} \mathbf{v}_{c_1}. \end{aligned} \tag{7}$$

By introducing the Tikhonov regularization term in Eq. (7), the solution can be obtained by solving a classical eigenvalue problem. Finally, the solution $\mathbf{v}_{c_1}$ can computed as the eigenvector corresponding to the maximum eigenvalue of $\left(\mathbf{H}_{c_1} - \rho_{c_1} \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z\right)$. □

Theorem 2. *The optimal solution* $\mathbf{v}_{c_2}$ *of the optimization problem (3) is the eigenvector corresponding to the maximum eigenvalue of the matrix* $(\mathbf{H}_{c_2} - \mathbf{X}_z^T \mathbf{L}_z \mathbf{X}_z)$.

The proof of Theorem 2 is similar to that of Theorem 1; thus, we omit the proof of Theorem 2. Theorems 1 and 2 provide the optimal projection vectors $\mathbf{v}_{c_1}$ and $\mathbf{v}_{c_2}$ by performing eigen-decomposition, respectively. After obtaining $\mathbf{v}_{c_1}$ and $\mathbf{v}_{c_2}$ for class pair $z = (c_1, c_2)$, we can respectively compute the distances between an unknown test point $\mathbf{x}$ and class centers $\mathbf{u}_{c_j}$ $(j = 1, 2)$ by

$$d(\mathbf{x}, \mathbf{u}_{c_1}) = \|\mathbf{v}_{c_1}^T \mathbf{x} - \mathbf{v}_{c_1}^T \mathbf{u}_{c_1}\| \tag{8}$$

and

$$d(\mathbf{x}, \mathbf{u}_{c_2}) = \|\mathbf{v}_{c_2}^T \mathbf{x} - \mathbf{v}_{c_2}^T \mathbf{u}_{c_2}\| \quad , \tag{9}$$

where $\|\cdot\|$ is the 2-norm of a vector. If $d(\mathbf{x}, \mathbf{u}_{c_1}) \leq d(\mathbf{x}, \mathbf{u}_{c_2})$ for class pair z, then $\mathbf{x}$ is more like a sample in class c_1.

Algorithm 1. OVO-LapPVP

Training Phase

Input: Sample matrices $\mathbf{X}_{l,c}$, $c = 1, \cdots, C$, and $\mathbf{X}_u$;

Obtain the set of class pairs $Z = \{z = (c_1, c_2) | c_1, c_2 = 1, 2, \cdots, C, c_1 < c_2, c_1 \neq c_2\}$.
for $\forall z \in Z$ **do**
    Construct an LapPVP classifier for class pair $z = (c_1, c_2)$;
    Obtain the training sample matrix $\mathbf{X}_z = [\mathbf{X}_{l,c_1}; \mathbf{X}_{l,c_2}; \mathbf{X}_u] \in \mathbb{R}^{n_z \times m}$ for the LapPVP with class pair (c_1, c_2), where samples in class c_1 are positive, and ones in class c_2 negative;
    Compute the matrices $\mathbf{H}_{c_1}$ and $\mathbf{H}_{c_2}$ by Eqs. (4) and (5), respectively;
    Generate the adjacency matrix $\mathbf{S}_z$ by Eq. (1) based on the training sample matrix $\mathbf{X}_z$;
    Compute the Laplacian matrix $\mathbf{L}_z = \mathbf{D}_z - \mathbf{S}_z$, where $\mathbf{D}_z$ is the diagonal matrix with $D_{z_{ii}} = \sum_j S_{z_{ij}}$;
    Obtain the projection vectors $\mathbf{v}_{c_1}$ and $\mathbf{v}_{c_2}$ according to Theorems 1 and 2, respectively;
end for

Output: Projection vector pairs $(\mathbf{v}_{c_1}, \mathbf{v}_{c_2})$, $(c_1, c_2) \in Z$.

Testing Phase

Input: Unknown test sample $\mathbf{x}$, sample matrices $\mathbf{X}_{l,c}$, $c = 1, \cdots, C$, $\mathbf{X}_u$, and projection vector pairs $(\mathbf{v}_{c_1}, \mathbf{v}_{c_2})$, $(c_1, c_2) \in Z$;

Initialize $\hat{\mathbf{y}} = [0, \cdots, 0]^T \in \mathbb{R}^C$;
for $\forall z \in Z$ **do**
    Compute the distances between $\mathbf{x}$ and class centers by Eq. (9) with optimal vectors $\mathbf{v}_{c_1}$ and $\mathbf{v}_{c_2}$;
    Update the c_1th element $\hat{y}_{c_1}$ or the c_2th element $\hat{y}_{c_2}$ in $\hat{\mathbf{y}}$ by Eq. (10);
end for
Assign the class label for $\mathbf{x}$ with Eq. (11);

Output: Estimated label for $\mathbf{x}$.

To assign a class label to $\mathbf{x}$, we construct a vote vector $\hat{\mathbf{y}} = [\hat{y}_1, \cdots, \hat{y}_C]^T$, where $\hat{y}_c$ is the vote for the cth class. Generally, the initial value of $\hat{y}_c$ is 0 for all $c = 1, \cdots, C$. For each class pair $z = (c_1, c_2)$, we update the vote for only class c_1 or class c_2. That is

$$\begin{cases} \hat{y}_{c_1} \leftarrow \hat{y}_{c_1} + 1, if \quad d(\mathbf{x}, \mathbf{u}_{c_1}) \leq d(\mathbf{x}, \mathbf{u}_{c_2}) \\ \hat{y}_{c_2} \leftarrow \hat{y}_{c_2} + 1, otherwise \end{cases}. \tag{10}$$

The update procedure is performed on the whole set Z. Thus, the strategy of classification rule for $\mathbf{x}$ is defined as

$$c^* = \arg \max_{c=1,\cdots,C} \hat{y}_c, \tag{11}$$

where c^* is the estimated class label for $\mathbf{x}$. We summarize the specific procedure of OVO-LapPVP in Algorithm 1.

2.2 OVR-LapPVP

Originally, LapPVP obtains a projection vector for each class that keeps data points in the same class as close to one another, meanwhile as far from points in the other class. In LapPVP, the within-class scatter requires only one-class training data, and the Laplacian regularization requires all data to construct the adjacent graph. However, the between-scatter captures the information of different classes. Hence, it is easy to generate multiple projection vectors for multi-class data by using the one-versus-rest strategy. This section proposes OVR-LapPVP by constructing C projection vectors for C-class training data. In other words, each class has its own projection vector.

To construct a binary classifier for the cth class, we suppose samples in class c are positive, and samples in other rest $(C-1)$ classes are negative. Let $\mathbf{X}_{l,c}$ be the labeled sample matrix for the cth class, and l_c is the number of samples in the cth class. Then the total labeled sample matrix is $\mathbf{X}_l = [\mathbf{X}_{l,1}; \cdots; \mathbf{X}_{l,C}]$, and the training sample matrix is $\mathbf{X} = [\mathbf{X}_l; \mathbf{X}_u]$, where $\mathbf{X}_u$ is the unlabeled sample matrix.

Thus, the cth optimization formulation of OVR-LapPVP is given by

$$\begin{aligned} \max_{\mathbf{v}_c} \quad & \mathbf{v}_c^T \mathbf{H}_c \mathbf{v}_c - \rho_c \mathbf{v}_c^T \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{v}_c, \\ s.t. \quad & \mathbf{v}_c^T \mathbf{v}_c = 1, \end{aligned} \tag{12}$$

where the discriminative matrix $\mathbf{H}_c$ is defined as

$$\mathbf{H}_c = \alpha_c \mathbf{B}_c - (1-\alpha_c)\mathbf{W}_c, \tag{13}$$

where $\alpha_c \in [0,1]$ is the parameter to balance the between-class scatter matrix $\mathbf{B}_c$ and within-class scatter matrix $\mathbf{W}_c$ of the cth class, $\mathbf{B}_c$ and $\mathbf{W}_c$ respectively have the forms

$$\mathbf{B}_c = \left(\mathbf{X} - \mathbf{e}\mathbf{u}_c^T\right)^T \left(\mathbf{X} - \mathbf{e}\mathbf{u}_c^T\right) \tag{14}$$

and

$$\mathbf{W}_c = (\mathbf{X}_c - \mathbf{e}_c\mathbf{u}_c^T)^T (\mathbf{X}_c - \mathbf{e}_c\mathbf{u}_c^T), \tag{15}$$

where $\mathbf{e} \in \mathbf{R}^n$ and $\mathbf{e}_c \in \mathbb{R}^{l_c}$ are vectors of ones. To find the solution of the optimization problem (12), we have the following theorem.

Theorem 3. *The optimal solution $\mathbf{v}_c$ of the optimization problem (12) is the eigenvector corresponding to the maximum eigenvalue of the matrix $(\mathbf{H}_c - \mathbf{X}^T\mathbf{L}\mathbf{X})$.*

Proof. The Lagrangian function of Eq. (12) can be written as

$$L(\mathbf{v}_c, \lambda_c) = \mathbf{v}_c^T \mathbf{H}_c \mathbf{v}_c - \rho_c \mathbf{v}_c^T \mathbf{X}^T \mathbf{L} \mathbf{X} \mathbf{v}_c - \lambda_c(\mathbf{v}_c^T \mathbf{v}_c - 1). \tag{16}$$

Then, we derive the partial derivative of $L(\mathbf{v}_c, \lambda_c)$ with respect to $\mathbf{v}_c$ and make it equal zero and have

$$\begin{aligned} & \frac{\partial L(\mathbf{v}_c, \lambda_c)}{\partial \mathbf{v}_c} = \left(\mathbf{H}_c - \rho_c \mathbf{X}^T \mathbf{L} \mathbf{X}\right) \mathbf{v}_c - \lambda_c \mathbf{v}_c = 0 \\ & \Rightarrow \left(\mathbf{H}_c - \rho_c \mathbf{X}^T \mathbf{L} \mathbf{X}\right) \mathbf{v}_c = \lambda_c \mathbf{v}_c. \end{aligned} \tag{17}$$

Finally, the cth optimization problem of OVR-LapPVP is also transferred to an eigenvalue decomposition problem. The solution $\mathbf{v}_c$ is obtained as the eigenvectors corresponding to the maximum eigenvalue of the matrix $\left(\mathbf{H}_c - \rho_c \mathbf{X}^T \mathbf{L} \mathbf{X}\right)$. $\square$

After obtaining C projection vectors, one for each class, the unknown test data is predicted by the projected distances between it and all class centers. The decision function is defined as:

$$c^* = \arg \min_{c=1,2,\ldots,C} \|\mathbf{v}_c^T \mathbf{x} - \mathbf{v}_c^T \mathbf{u}_c\| \tag{18}$$

where c^* is the estimated class label of $\mathbf{x}$. We summarize the specific procedure of OVR-LapPVP in Algorithm 2.

Algorithm 2. OVR-LapPVP

Training Phase

Input: Sample matrices $\mathbf{X}_{l,c}$, $c = 1, \cdots, C$, and $\mathbf{X}_u$;

```
Let X = [X_{l,1}; ··· ; X_{l,C}; X_u] ∈ ℝ^{n×m} be the training samples;
Obtain the adjacency matrix S by Eq. (1) based on X;
Compute the Laplacian matrix L = D − S, where D is the diagonal matrix with
  D_ii = ∑_j S_ij;
for c = 1 to C do
    Compute the discriminative matrix H_c by Eq. (13) for the cth class data points;
    Obtain the vectors v_c by solving eigenvalue decomposition problem Eq. (17);
end for
```

Output: Projection vectors $\mathbf{v}_c$, $c = 1, \cdots, C$.

Testing Phase

Input: Unknown test sample $\mathbf{x}$, sample matrices $\mathbf{X}_{l,c}$, $c = 1, \cdots, C$, $\mathbf{X}_u$, and projection vectors $\mathbf{v}_c$, $c = 1, \cdots, C$;

```
for c = 1 to C do
    Calculate the distance from x to the cth class;
end for
Assign the class label to x by Eq. (18).
```

Output: Estimated label for $\mathbf{x}$.

2.3 Comparison of Multi-class LapPVP

We make a comparison for the proposed two multi-class methods here. OVO-LapPVP constructs $C(C-1)/2$ binary classifiers by solving $C(C-1)/2$ pairs of eigenvalue decomposition problems. The computational complexity of OVO-LapPVP is approximately $O(C^2 m^3)$. OVR-LapPVP generates C projection vectors by solving C eigenvalue decomposition problems. The computational complexity of OVR-LapPVP is approximately $O(Cm^3)$.

Obviously, OVO-LapPVP requires a large amount of training time as compared to OVR-LapPVP. Moreover, the strategy of classification in OVO-LapPVP is more complex than that of OVR-LapPVP because OVO-LapPVP handles more binary classifiers. Of course, OVR-LapPVP has its own disadvantage of having a greater space complexity because its optimization problems are performed on all training data.

3 Experiments

3.1 Experimental Setup

To confirm the feasibility and effectiveness of the proposed multi-class methods, we need to compare them with other multi-class ones. However, few paper discusses the semi-supervised multi-class tasks. Thus, algorithms compared here are extended by binary semi-supervised classifiers (LapTSVM, LapTPMSVM, and LapLSTSVM) using strategies of both OVO and OVR, where LapTSVM, LapTPSVM and LapLSTSVM are recently popular nonparallel classifiers for semi-supervised learning as well as LapPVP. Finally, we get six multi-class algorithms: OVO-LapTSVM, OVR-LapTSVM, OVO-LapTPMSVM, OVR-LapTPMSVM, OVO-LapLSTSVM and OVR-LapLSTSVM.

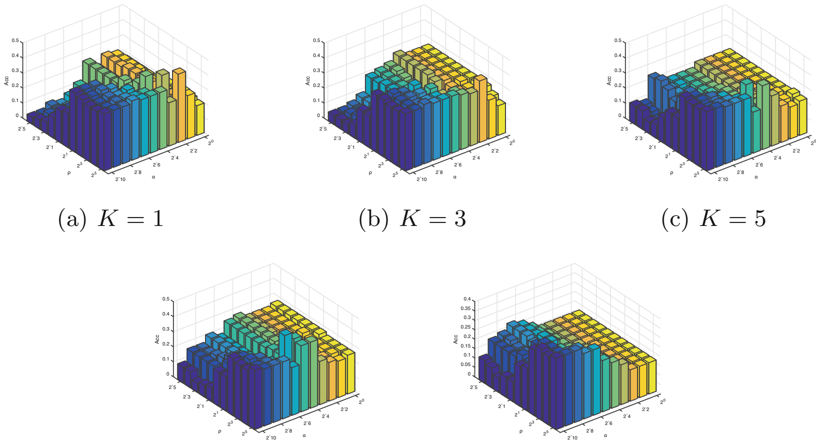
Experiments are conducted on benchmark datasets that are from the University of California at Irvine (UCI) Machine Learning Repository [2]. The details of datasets are presented in Table 1. For each dataset, we randomly select 70% of samples from each class for training, and the remaining 30% for test. The features of data are all normalized to the interval $[0, 1]$. To reduce the running time of parameter selection, hyper-parameters are set to the same value for all binary classifiers in one multi-class method. In addition, the grid search method [4] is used for selecting the optimal hyper-parameters. In OVO-LapPVP and OVR-LapPVP, α takes its value from the set $\{2^{-10}, 2^{-9}, \ldots, 2^{0}\}$, and ρ from $\{2^{-5}, 2^{-4}, \ldots, 2^{5}\}$. Moreover, the number of nearest neighbors varies in the set $\{1, 3, 5, 7, 9\}$.

Table 1. Information of benchmark datasets

Dataset	#Sample	#Feature	#Class
Balance	625	4	3
Dnatest	1186	180	3
Glass	214	9	6
Iris	150	4	3
Lungcancer	32	56	3
Waveform	5000	21	3
Wine	178	13	3
X8D5K	1000	8	5
Zoo	101	16	7

All Matlab scripts of the involved algorithms are written by ourselves. For a fair comparison, we exploit the Matlab toolbox of quadratic programming to solve quadratic programmings in the relevant methods. All methods are implemented in MATLAB R2015b on a personal computer, whose system configuration is Intel Core i5 (3.6 GHz) and 8 GB random access memory.

Parameters Analysis. It is well known, the parameters of a classifier may have a great influence on its classification performance. Here, we conduct the grid search method to analyze the impact of parameters on proposed methods through the Glass dataset. As mentioned before, the pair of parameters take the same values. In OVO-LapPVP and OVR-LapPVP, α takes its value from the set $\{2^{-10}, 2^{-9}, \ldots, 2^{0}\}$, and ρ from $\{2^{-5}, 2^{-4}, \ldots, 2^{5}\}$. Here, K is selected from the set $\{1, 3, 5, 7, 9\}$.

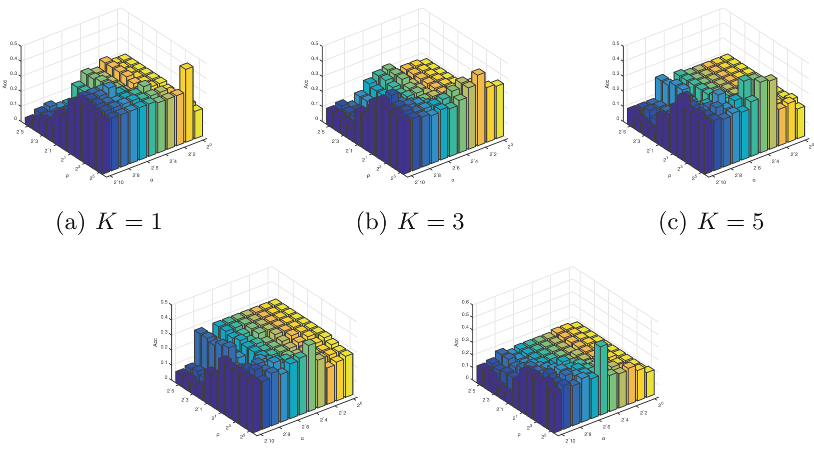


(a) $K = 1$ (b) $K = 3$ (c) $K = 5$

(d) $K = 7$ (e) $K = 9$

Fig. 1. Accuracy vs. parameters (α, ρ) of OVO-LapPVP on Glass dataset under different K

Figures 1 and 2 separately show the influence of parameters of OVO-LapPVP and OVR-LapPVP with 20% of labeled data and 50% of unlabeled data on the Glass dataset. It is obviously observed that the pair of parameters (α, ρ) can also greatly affect the performance of OVO-LapPVP and OVR-LapPVP. Additionally, when varying the number of nearest neighbors, the optimal pair of parameters (α, ρ) is totally different. Thus, the pair of optimal parameters cannot be fixed for all datasets. Therefore, parameter selection may be an issue for our method. However, the grid search method can help us to find appropriate parameters for OVO-LapPVP and OVR-LapPVP in experiments.



(a) $K = 1$ (b) $K = 3$ (c) $K = 5$

(d) $K = 7$ (e) $K = 9$

Fig. 2. Accuracy vs. parameters (α, ρ) of OVR-LapPVP on Glass dataset under different K

3.2 Results and Discussion

Experiments are conducted on nine UCI datasets to compare the classification performance of semi-supervised multi-class methods. We run 10 trials on each dataset and report the average accuracy.

Table 2 lists the results of eight semi-supervised multi-class classification methods on nine datasets with 10% of labeled data and 50% of unlabeled data. It is clear that the proposed methods including OVO-LapPVP and OVR-LapPVP can perform better. On Dnatest, Iris, Lungcancer, Wine, X85DK and Zoo datasets, OVO-LapPVP has the highest accuracy, followed by OVR-LapPVP. On the Glass dataset, OVR-LapPVP has the highest accuracy, followed by OVO-LapPVP.

Tables 3 and 4 separately summary the results of eight multi-class methods obtained with 20% and 30% of labeled data, respectively. Both OVO-LapPVP and OVR-LapPVP outperform other methods on all datasets except Balance and Waveform. The classification performance on the X8D5K dataset shows that the proposed semi-supervised classification methods are effective for multi-class classification tasks.

Observation on Tables 2, 3 and 4 indicates that OVO-LapPVP achieves the highest performance on 15 cases and the second highest on 6 cases. Note that there are 27 cases totally. At the same time, we can see that OVR-LapPVP is the best on 9 cases and the second best on 12 cases. To be simply, our methods are superior to other methods in 24 out of 27 cases. Findings suggests that OVO-LapPVP has the best performance for multi-class classification tasks among compared methods, followed by OVR-LapPVP.

Table 2. Average accuracy and standard deviation (%) obtained by eight multi-class methods with 10% of labeled data

Dataset	OVO-LapTSVM	OVO-LapTPMSVM	OVO-LapLSTSVM	OVO-LapPPV
Balance	84.13 ± 2.41	73.17 ± 4.67	**85.93 ± 2.70**	84.13 ± 1.28
Dnatest	79.52 ± 3.13	69.86 ± 2.50	52.67 ± 0.79	**83.17 ± 1.94**
Glass	44.55 ± 8.69	41.82 ± 5.98	45.00 ± 6.70	47.12 ± 7.01
Iris	91.33 ± 4.12	92.89 ± 3.60	94.89 ± 2.78	**96.44 ± 1.55**
Lungcancer	45.00 ± 10.80	46.00 ± 10.75	37.00 ± 12.52	**64.00 ± 8.43**
Waveform	84.38 ± 0.77	81.81 ± 0.55	**85.37 ± 0.93**	79.95 ± 1.46
Wine	95.27 ± 2.74	94.36 ± 2.90	93.64 ± 3.76	**95.45 ± 2.14**
X8D5K	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
Zoo	88.79 ± 4.56	89.70 ± 4.09	90.00 ± 5.16	**91.70 ± 3.79**
Dataset	OVR-LapTSVM	OVR-LapTPMSVM	OVR-LapLSTSVM	OVR-LapPPV
Balance	81.06 ± 5.24	68.41 ± 5.04	85.56 ± 2.60	83.02 ± 2.83
Dnatest	66.80 ± 4.10	73.46 ± 2.16	60.98 ± 2.17	82.92 ± 1.87
Glass	42.73 ± 6.05	41.97 ± 4.95	46.21 ± 4.24	**48.79 ± 5.19**
Iris	84.22 ± 8.41	86.44 ± 6.58	82.44 ± 4.85	95.56 ± 3.47
Lungcancer	39.00 ± 11.97	43.00 ± 12.52	37.00 ± 13.37	60.00 ± 12.47
Waveform	85.29 ± 0.87	78.75 ± 1.87	83.42 ± 1.31	78.30 ± 2.54
Wine	92.55 ± 5.10	93.27 ± 2.28	92.91 ± 5.17	95.64 ± 5.09
X8D5K	99.97 ± 0.11	99.77 ± 0.39	99.97 ± 0.11	**100.00 ± 0.00**
Zoo	89.09 ± 4.56	86.67 ± 3.83	90.61 ± 4.15	88.79 ± 3.21

The running time of eight methods on nine datasets with 20% of labeled and 50% of unlabeled training data is recorded in Fig. 3. The first, third, fifth, and seventh columns in every sub-figure represent the running time obtained by methods using the OVO strategy, and the rest columns represent those using OVR. Obviously, the OVR-based methods spend less time than OVO-based ones because the OVR-based methods need to solve less optimization problems. Four methods, OVO-LapLSTSVM, OVR-LapLSTSVM, OVO-LapPVP and OVR-LapPVP, that avoid the complex quadratic programmings can save much time. Therefore, OVO-LapPVP and OVR-LapPVP are promising when applied to multi-class classification tasks.

Additionally, we give the rank of individual methods in Table 5. A method that has the highest accuracy is ranked the first, that has the second highest accuracy is ranked the second, and so on. Digits in the row of "10%" mean the average rank obtained from Table 2, "20%" and "30%" are from Tables 3 and 4, respectively. The row of "Average rank" provides the average rank over 27 cases, and that of "Friedman test" shows the rank difference between methods

Table 3. Average accuracy and standard deviation (%) obtained by eight multi-class methods with 20% of labeled data

Dataset	OVO-LapTSVM	OVO-LapTPMSVM	OVO-LapLSTSVM	OVO-LapPPV
Balance	86.72 ± 2.37	73.81 ± 2.58	**87.30 ± 1.75**	80.32 ± 2.61
Dnatest	82.81 ± 1.52	77.50 ± 1.88	56.97 ± 0.92	86.07 ± 1.61
Glass	48.03 ± 5.05	43.03 ± 6.11	47.88 ± 5.59	**54.85 ± 7.03**
Iris	95.11 ± 2.73	94.89 ± 1.83	94.67 ± 2.81	96.00 ± 4.03
Lungcancer	40.00 ± 14.91	41.00 ± 15.95	40.00 ± 16.33	**68.00 ± 9.19**
Waveform	84.95 ± 1.02	81.72 ± 1.06	**86.59 ± 0.94**	81.09 ± 1.09
Wine	95.64 ± 1.76	95.09 ± 2.28	94.55 ± 4.62	**96.00 ± 1.43**
X8D5K	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
Zoo	92.42 ± 2.58	90.91 ± 3.50	90.30 ± 5.68	**93.33 ± 2.39**
Dataset	OVR-LapTSVM	OVR-LapTPMSVM	OVR-LapLSTSVM	OVR-LapPPV
Balance	81.85 ± 4.11	68.84 ± 3.33	86.83 ± 1.68	83.86 ± 4.54
Dnatest	75.51 ± 2.20	78.51 ± 1.97	70.34 ± 2.72	**86.26 ± 1.70**
Glass	47.88 ± 5.63	39.55 ± 7.71	48.03 ± 5.67	49.73 ± 9.37
Iris	87.33 ± 6.63	88.22 ± 8.38	83.56 ± 4.34	**96.44 ± 3.51**
Lungcancer	40.00 ± 17.64	40.00 ± 9.43	39.00 ± 17.92	62.00 ± 9.19
Waveform	85.69 ± 0.83	77.95 ± 1.62	85.77 ± 0.93	79.13 ± 0.99
Wine	94.00 ± 3.84	93.64 ± 4.39	94.00 ± 3.54	96.73 ± 4.91
X8D5K	**100.00 ± 0.00**	99.73 ± 0.34	99.97 ± 0.11	**100.00 ± 0.00**
Zoo	90.61 ± 3.90	90.61 ± 3.33	89.39 ± 3.57	92.73 ± 2.93

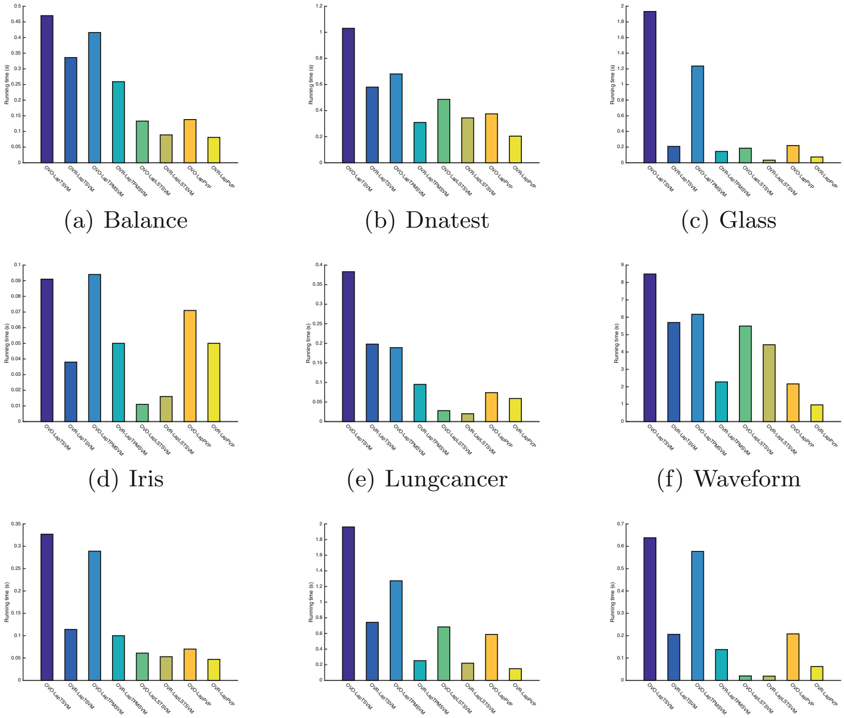
and reference method, where OVR-LapPVP is taken as the reference method. The smaller the value of average rank and Friedman test is, the better performance the corresponding method has. It is obvious that OVO-LapPVP is slightly better than OVR-LapPVP since the value of OVO-LapPVP is less than 0 in the Friedman test. According to Table 5, we can conclude OVO-LapPVP has a greater superiority, and OVR-LapPVP ranks the second.

Table 4. Average accuracy and standard deviation (%) obtained by eight multi-class methods with 30% of labeled data

Dataset	OVO-LapTSVM	OVO-LapTPMSVM	OVO-LapLSTSVM	OVO-LapPPV
Balance	**88.89 ± 1.60**	74.50 ± 3.47	87.46 ± 1.22	84.13 ± 2.51
Dnatest	84.38 ± 1.82	81.97 ± 2.13	60.65 ± 1.75	86.60 ± 0.96
Glass	49.70 ± 5.50	43.48 ± 8.89	51.36 ± 5.73	**51.52 ± 3.71**
Iris	94.89 ± 2.78	93.11 ± 3.70	94.89 ± 2.58	95.78 ± 3.22
Lungcancer	58.00 ± 12.29	43.00 ± 14.94	53.00 ± 16.36	76.00 ± 9.66
Waveform	85.19 ± 1.01	81.80 ± 0.79	**86.86 ± 1.00**	81.46 ± 0.84
Wine	96.55 ± 1.81	96.36 ± 1.21	95.09 ± 3.64	**97.45 ± 1.53**
X8D5K	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**	**100.00 ± 0.00**
Zoo	92.42 ± 1.28	91.82 ± 2.49	90.30 ± 1.92	**93.33 ± 2.58**
Dataset	OVR-LapTSVM	OVR-LapTPMSVM	OVR-LapLSTSVM	OVR-LapPPV
Balance	82.28 ± 3.04	67.99 ± 6.32	87.72 ± 1.43	84.42 ± 5.07
Dnatest	80.93 ± 2.52	81.66 ± 2.29	76.83 ± 2.91	**87.25 ± 0.65**
Glass	49.39 ± 5.06	40.61 ± 6.69	47.27 ± 5.57	50.61 ± 6.45
Iris	90.00 ± 8.13	88.67 ± 6.83	81.33 ± 3.66	**96.00 ± 3.44**
Lungcancer	45.00 ± 15.81	46.00 ± 16.47	45.00 ± 8.50	**77.00 ± 6.75**
Waveform	85.99 ± 0.87	78.02 ± 1.36	86.42 ± 0.79	78.22 ± 0.76
Wine	95.45 ± 3.46	96.00 ± 2.24	96.55 ± 2.18	97.64 ± 1.50
X8D5K	**100.00 ± 0.00**	99.80 ± 0.28	**100.00 ± 0.00**	**100.00 ± 0.00**
Zoo	91.21 ± 3.01	91.21 ± 2.65	90.00 ± 4.75	91.82 ± 3.21

Table 5. Rank of eight methods on UCI datasets

Classifier	10%	20%	30%	Average rank	Friedman test
OVO-LapTSVM	3.67	3.00	2.67	3.11	0.52
OVR-LapTSVM	5.22	4.56	4.67	4.81	2.22
OVO-LapTPMSVM	4.44	4.22	4.67	4.44	1.85
OVR-LapTPMSVM	5.78	5.67	5.67	5.70	3.11
OVO-LapLSTSVM	3.67	3.89	3.78	3.78	1.19
OVR-LapLSTSVM	4.67	4.67	4.44	4.59	2.00
OVO-LapPPV	2.00	2.44	2.44	2.30	−0.30
OVR-LapPPV	3.00	2.33	2.44	2.59	0.00



(a) Balance (b) Dnatest (c) Glass

(d) Iris (e) Lungcancer (f) Waveform

(g) Wine (h) X8D5K (i) Zoo

Fig. 3. Running time of eight methods on UCI datasets

4 Conclusion

In this paper, we propose OVO-LapPVP and OVR-LapPVP for multi-class classification tasks. As the extension of LapPVP, the proposed methods both can take advantage of the discriminative information of the labeled data and the graph structure of unlabeled data to obtain the optimal projection vectors. OVO-LapPVP adopts the OVO strategy, and OVR-LapPVP adopts the OVR strategy. That is to say, OVO-LapPVP obtains $C(C-1)/2$ pairs of projection vectors, and OVR-LapPVP obtains C ones for the C-class data. Since OVO-LapPVP needs more optimization problems to solve than OVR-LapPVP, which results that OVO-LapPVP costs more time than OVR-LapPVP on the same dataset.

Experiments conducted on the UCI datasets have demonstrated that the proposed methods have better classification performance compared with other popular semi-supervised methods based on manifold regularization and twin support vector machine. OVO-LapPVP outperforms OVR-LapPVP in classification performance, but is worse than OVR-LapPVP in running efficiency. Therefore, OVO-LapPVP and OVR-LapPVP can be applied to different scenes.

OVO-LapPVP and OVR-LapPVP are both excellent on solving multi-class classification problems for linear cases, in the following work, we can consider the methods with kernel ticks for nonlinear cases.

Acknowledgment. This work was supported in part by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China under Grant Nos. 19KJA550002 and 19KJA610002, by the Priority Academic Program Development of Jiangsu Higher Education Institutions, and by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Chen, W., Shao, Y., Deng, N., Feng, Z.: Laplacian least squares twin support vector machine for semi-supervised classification. Neurocomputing **145**, 465–476 (2014)
2. Dua, D., Graff, C.: UCI machine learning repository (2017). https://archive.ics.uci.edu/ml
3. Goyal, N., Gupta, K., Kumar, N.: Multiclass twin support vector machine for plant species identification. Multimedia Tools Appl. **78**(19), 27785–27808 (2019)
4. Iwan, S., Adam, P., Gary, W.: SVM parameter optimization using grid search and genetic algorithm to improve classification performance. Telecommun. Comput. Electron. Control **12**, 1502–1509 (2016)
5. Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. IEEE Trans. Pattern Anal. Mach. Intell. **29**(5), 905–910 (2007)
6. Kumar, M.A., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Syst. Appl. **36**(4), 7535–7543 (2009)
7. Peng, X.: TPMSVM: a novel twin parametric-margin support vector machine for pattern recognition. Pattern Recogn. **44**(10–11), 2678–2692 (2011)
8. Qi, Z., Tian, Y., Shi, Y.: Laplacian twin support vector machine for semi-supervised classification. Neural Netw. **35**, 46–53 (2012)
9. Tomar, D., Agarwal, S.: A comparison on multi-class classification methods based on least squares twin support vector machine. Knowl. Based Syst. **81**, 131–147 (2015)
10. Xie, J., Hone, K.S., Xie, W., Gao, X., Shi, Y., Liu, X.: Extending twin support vector machine classifier for multi-category classification problems. Intell. Data Anal. **17**(4), 649–664 (2013)
11. Xue, Y., Zhang, L.: Laplacian pair-weight vector projection for semi-supervised learning. Inf. Sci. **573**, 1–19 (2021)
12. Yang, Z., Xu, Y.: Laplacian twin parametric-margin support vector machine for semi-supervised classification. Neurocomputing **171**, 325–334 (2016)