



# Robust and sparse label propagation for graph-based semi-supervised classification

Zhiwen Hua<sup>1</sup> · Youlong Yang<sup>1</sup>

Accepted: 15 March 2021 / Published online: 5 July 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

## Abstract

Traditional graph-based semi-supervised classification algorithms are usually composed of two independent parts: graph construction and label propagation. However, the predefined graph may not be optimal for label propagation, and these methods usually use the raw data containing noise directly, which may reduce the accuracy of the algorithm. In this paper, we propose a robust label prediction model called the robust and sparse label propagation (RSLP) algorithm. First, our RSLP algorithm decomposes the raw data into a low-rank clean part and a sparse noise part, and performs graph construction and label propagation in the clean data space. Second, RSLP seamlessly combines the processes of graph construction and label propagation. By jointly minimizing the sample reconstruction error and the label reconstruction error, the resulting graph structure is globally optimal. Third, the proposed RSLP performs  $l_{2,1}$ -norm regularization on the predicted label matrix, thereby enhancing the sparsity and discrimination of soft labels. We also analyze the connection between RSLP and other related algorithms, including label propagation algorithms, the robust graph construction method, and principal component analysis. A series of experiments on several benchmark datasets show that our RSLP algorithm achieves comparable and even higher accuracy than other state-of-the-art algorithms.

**Keywords** Semi-supervised learning · Graph construction · Label propagation · Denoising ·  $l_{2,1}$ -norm regularization

## 1 Introduction

As it takes time and effort to label samples, we often encounter a situation where the number of labeled samples is far less than that of unlabeled samples in practical applications. Therefore, how to make full and efficient use of a small number of labeled samples has become a hot issue [6, 24]. The purpose of semi-supervised learning (SSL) is to use a small number of labeled samples to predict the labels of unlabeled samples. SSL can be roughly divided into four categories according to the principles employed by its algorithms: self-training [34, 36], co-training [1,

48], generation model [14, 18], and graph-based semi-supervised learning (GSSL) [11, 50]. The current deep learning methods have difficulty solving the problem of scarce labeled data, and thus researchers have proposed deep semi-supervised learning [4, 37] and few-shot learning [7, 38]. Here we focus on GSSL, which has been widely used in data mining and pattern recognition [19, 21] due to its high efficiency and effectiveness.

GSSL is comprised of two steps: graph construction [10, 12, 51] and label propagation [16, 22, 31, 44, 45, 47, 49]. Graph construction methods treat all labeled and unlabeled samples as nodes in the graph, construct edges between pairs of nodes and assign corresponding weights to represent the similarity between the samples. The label propagation algorithm propagates the label information from the labeled samples to the unlabeled samples according to the structure of the graph, thereby obtaining the labels of all samples. GSSL has two basic assumptions: the clustering assumption and the manifold assumption. The former assumes that samples in the same cluster are more likely to have the same label; the latter

---

✉ Youlong Yang  
ylyang@mail.xidian.edu.cn

Zhiwen Hua  
zwhua@stu.xidian.edu.cn

<sup>1</sup> School of Mathematics and Statistics, Xidian University, Xi'an, 710071, China

assumes that the samples in a small local neighborhood have similar properties, that is, their labels should also be similar.

Since the quality of the graph affects the performance of subsequent algorithms, how to build a real and reliable graph is the core of GSSL. The most common graph construction methods are the  $k$  nearest neighbor ( $k$ NN) and  $\varepsilon$ -neighborhoods methods, but both rely heavily on the neighbor parameters  $k$  and  $\varepsilon$ , and on parameter  $\sigma$  of the Gaussian kernel function [27, 40]. How to choose the appropriate parameter values is still a problem to be solved. Recently, self-expressiveness based graph-learning methods have received much attention. Such methods include local linear representation [28, 32], low-rank representation [13, 52] and  $l_1$  graph [9, 35]. In these methods, each sample is represented by a linear combination of other samples, and the coefficients are used as edge weights.

Label propagation aims to predict the labels of unlabeled samples. Depending on whether the algorithm can process out-of-samples, the existing label propagation algorithms can be divided into two types: transductive and inductive model. The transductive model predicts the existing unlabeled samples, while the inductive model can process new samples directly and efficiently. Typical transductive models include the local and global consistency (LGC) algorithm [47], Gaussian fields and harmonic function (GFHF) [49], linear neighborhood propagation (LNP) [31], special label propagation (SLP) [22], sparse neighborhood propagation (SparseNP) [43], adaptive label propagation by double matrix decomposition (ALP-MD) [42], classic inductive models include flexible manifold embedding (FME) [23], discriminative sparse FME (SparseFME) [44], and joint sparse graph and flexible embedding (JSGFE) [8]. FME, SparseFME and JSGFE learn a linear projection to predict the labels of new samples.

Although the abovementioned label propagation methods have achieved good results, they usually have at least one of the following shortcomings. (1) Most of these methods separate graph construction and label propagation into two independent steps, that is, they pre-construct the graph and then perform label propagation. This cannot ensure an overall optimal performance, and does not make full use of the correlation between graph structure and label propagation. (2) Real-world data always contains noise, and most of these methods operate directly on the raw data without considering removing noise. (3) The resulting soft label usually contains mixed symbols that do not ensure sufficient sparsity and discrimination.

This paper proposes a new label propagation method named robust and sparse label propagation (RSLP). The RSLP algorithm has the following three characteristics:

- 1) RSLP decomposes the raw data into a low-rank clean part and a sparse error part.
- 2) RSLP jointly performs label propagation and graph construction in the clean data space, and ensures that the learned similarity matrix is globally optimal by minimizing both the label reconstruction error and the sample reconstruction error.
- 3) RSLP enhances the discrimination of soft labels by introducing  $l_{2,1}$ -norm regularization to the predicted label matrix.

The rest of this paper is organized as follows. Section 2 introduces related work, including some notations, graph construction methods, and label propagation algorithms. Section 3 proposes our RSLP model and its optimization steps. In Section 4, we discuss the connection between RSLP and other algorithms. Section 5 describes our experimental setup and presents the results. Finally, Section 6 summarizes the paper.

## 2 Related work

In this section, we introduce the notations used in this paper and some closely related work. Given a dataset  $X = \{X_L \cup X_U\} \in \mathcal{R}^{d \times n}$  with  $n$  samples and  $d$  features, where  $X_L = [x_1, x_2, \dots, x_l] \in \mathcal{R}^{d \times l}$  is a labeled dataset with  $l$  samples,  $X_U = [x_{l+1}, x_{l+2}, \dots, x_{l+u}] \in \mathcal{R}^{d \times u}$  is an unlabeled dataset with  $u$  samples, and each column vector  $x_i \in \mathcal{R}^{d \times 1}$  is a sample. Let  $Y = [y_1, y_2, \dots, y_n] \in \mathcal{R}^{c \times n}$  be the initial label matrix with  $y_{ij} = 1$  if  $x_j$  is labeled as  $i$ , and  $y_{ij} = 0$  otherwise, where  $c$  is the number of classes.

The graph construction methods regard all samples in the dataset  $X$  as nodes in the graph, and construct edges according to the similarity between samples. In this way, one can obtain a weighted neighborhood graph  $G = \{X, S\}$  with node set  $X$  and similarity matrix  $S \in \mathcal{R}^{n \times n}$ , where  $S_{ij}$  represents the similarity between sample  $x_i$  and  $x_j$ . Based on the structure of the graph, the label propagation algorithm propagates the known label information and obtains the predicted label matrix  $F = [f_1, f_2, \dots, f_n] \in \mathcal{R}^{c \times n}$ . In addition,  $e$  is a vector with all elements as 1,  $(s)_+$  represents  $\max(s, 0)$ , and  $\text{tr}(\cdot)$  and  $^T$  denote the trace and transpose of a matrix, respectively.

For a matrix  $A$ , let  $A_{i\cdot}$ ,  $A_{\cdot j}$ , and  $A_{ij}$  denote the  $i$ -th row vector,  $j$ -th column vector, and the element in the  $i$ -th row and  $j$ -th column of  $A$ , respectively. The  $l_1$ -norm, nuclear norm, Frobenius norm, and  $l_{2,1}$ -norm of  $A$  are defined as follows [17, 44]:

$$\|A\|_1 = \sum_i \sum_j |A_{ij}| \quad (1)$$

$$\|A\|_* = \sum_i \sigma_i \quad (2)$$

where  $\sigma_i$  is the  $i$ -th singular value of  $A$ .

$$\begin{aligned}\|A\|_F^2 &= \sum_i \sum_j A_{ij}^2 = \sum_i \|A_{i.}\|_2^2 = \sum_j \|A_{.j}\|_2^2 \\ &= \text{tr}(A^T A) = \text{tr}(A A^T)\end{aligned}\quad (3)$$

$$\|A\|_{2,1} = \sum_i \left[ \sum_j A_{ij}^2 \right]^{1/2} = \sum_i \|A_{i.}\|_2 = 2\text{tr}(A^T H A)\quad (4)$$

where  $H$  is a diagonal matrix with  $H_{ii} = 1/2\|A_{i.}\|_2$ .

## 2.1 Graph construction

The graph construction method is the core of GSSL. The most common graph construction methods are the  $k$  nearest neighbor( $k$ NN) and  $\varepsilon$ -neighborhoods methods. These two methods control the number of edges in the graph through the neighbor parameters  $k$  and  $\varepsilon$ . For the  $k$ NN graph, if node  $x_i$  is a member of  $x_j$ 's  $k$ -nearest neighbor or vice versa, an edge is created between  $x_i$  and  $x_j$ . For the  $\varepsilon$ -neighborhoods graph, if the distance between two nodes is less than the threshold  $\varepsilon$ , the nodes are connected by an edge. The corresponding edge weights are given through the Gaussian kernel function:

$$S_{ij} = \begin{cases} \exp(-\|x_i - x_j\|^2 / 2\sigma^2), & \text{if } x_i \text{ and } x_j \text{ are connected,} \\ 0, & \text{otherwise.} \end{cases}\quad (5)$$

Based on the theory of local linear reconstruction [28], some graph construction methods have been proposed to avoid the influence of parameter on the graph structure. The main idea of these methods is that each sample can be represented by a linear combination of other samples, and the coefficients are used as the edge weights. Wang et al. [31] developed a non-negative local linear reconstruction model to ensure that the obtained edge weights are all non-negative.

In order to ensure the sparseness of the graph, the  $l_1$  graph [5] learns the weights by solving the  $l_1$  regularization problem, that is, by calculating the sparsest reconstruction coefficients of each sample separately. Qiao et al. [26] proposed the sparsity preserving projections (SPP) method based on the sparse representation theory for learning the weight matrix and low-dimensional embedding of data. Yan et al. [39] used sparse coding to construct a graph structure that is more robust to noise, and introduced a semi-supervised learning method based on the  $l_1$  graph. Weng et al. [35] proposed a graph construction method based

on data self-representation and Laplacian smoothness, and combined with an adaptive coding scheme to improve the method to obtain sparse graphs. Dornaika et al. [9] proposed a novel sparse graph construction method using Laplacian smoothness, and applied it to semi-supervised classification.

Unlike the  $l_1$  graph, the goal of the low-rank graph is to obtain the jointly lowest-rank representation of all samples. Therefore, the low-rank graph construction methods can better obtain global data structures [27]. Zheng et al. [46] proposed the low-rank representation with local constraint method to capture both the global and local structure of the data. Peng et al. [25] proposed an enhanced low-rank representation via sparse manifold adaption, which can explicitly consider the local manifold structure of the data. Based on the idea that edges between adjacent points in a graph should have large weights, Fei et al. [13] introduced a novel low rank representation model with adaptive distance penalty. Zhuang et al. [52] incorporated the label information of the observed samples into the low-rank representation model to construct a more efficient graph structure for the semi-supervised learning problem. The objective functions of the  $l_1$  graph and low-rank graph can be unified into the following model [27]:

$$\min_S \|X - XS\|_a + \lambda \|S\|_b. \quad (6)$$

For the  $l_1$  graph,  $\|\cdot\|_a$  and  $\|\cdot\|_b$  represent  $\|\cdot\|_F^2$  and  $\|\cdot\|_1$ , respectively; and for the low-rank graph, they represent  $\|\cdot\|_{2,1}$  and  $\|\cdot\|_*$ .

However, most methods construct a graph structure from raw data, which are often corrupted in practice. In order to learn reliable graph from noisy data, Kang et al. [17] proposed a robust graph construction(RGC) method. The main idea of RGC is that the clean data matrix is low-rank and the noisy data matrix is sparse. RGC decomposes the raw data into a clean part  $C$  and an error part  $E$  to adaptively remove the noise, and then learns the similarity matrix  $S$  on the clean data, thereby improving the quality of the graph. Moreover, RGC optimizes  $C$  and  $S$  in the same objective function, and enhances each other by alternate optimization. The model is represented as follows:

$$\begin{aligned}\min_{C,E,S} & \|C\|_* + \alpha \|E\|_1 + \beta \text{tr}(CLC^T) + \gamma \|S\|_F^2 \\ \text{s.t. } & X = C + E, e^T S = e^T, 0 \leq S \leq 1,\end{aligned}\quad (7)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are trade-off parameters; and  $L = D - (S + S^T)/2$  is the graph Laplacian matrix, where  $D$  is a diagonal matrix with  $d_{ii} = \sum_j [(s_{ij} + s_{ji})/2]$ .

## 2.2 Label propagation

The purpose of label propagation is to predict the labels of unlabeled samples based on the graph structure and known label information, that is, to obtain the predicted

label matrix  $F$ . Note that  $F$  should satisfy two conditions: (1) for labeled samples, the predicted label should be as close to the true label as possible; and (2) for all samples, the predicted label should be smooth on the graph. Many label propagation algorithms can be summarized by the following framework:

$$\min_F \text{fit}(F, Y) + \text{smooth}(F). \quad (8)$$

The first term is a fitting term that measures the difference between the true labels and the predicted labels. The second term is a smoothing term that smooths the predicted label matrix so that the labels of adjacent samples are similar. The difference between most label propagation algorithms is the definitions of the fitting term and the smoothing term. A brief introduction of some classic label propagation algorithms is as follows.

Zhu et al. [49] proposed a label propagation algorithm based on the Gaussian random field model, and analyzed the connections with random walks, electric networks, and spectral graph theory. Zhou et al. [47] used the symmetric normalized weight matrix to iteratively propagate label information, and allowed the initial label information to be changed, which enhances the robustness of the algorithm. Wang et al. [31] introduced the linear neighborhood propagation (LNP) method which assumes that each sample can be reconstructed by the weighted linear combination of its nearest neighbors. The core of the algorithm is the calculation of weights between samples. The objective functions of these three algorithms can be written as follows:

$$\min_F \left( (F - Y)U(F - Y)^T \right) + \text{tr}(FLF^T), \quad (9)$$

where  $L$  is the graph Laplacian matrix  $L = D - S$  or the normalized Laplacian matrix  $L = I - D^{-1/2}SD^{-1/2}$ .  $U$  is a diagonal matrix with the first  $l$  and the remaining  $n - l$  elements being  $\lambda_l$  and  $\lambda_u$ , respectively, and is used to balance the fitting and smoothing terms.

Zhang et al. [41] improved LNP algorithm by fully considering the label information of samples when calculating the weights. If two samples belong to the same class, they will be closer to each other in the feature space. Nie et al. [22] proposed a special label propagation algorithm, which can discover the potential novel classes and output the probability that the sample belongs to the labeled classes or the novel class. Nie et al. [23] incorporated the loss function used to measure the degree of mismatch between embedded features and soft labels into the existing label propagation framework. By minimizing the objective function, a linear projection classifier for predicting the label of new sample can be obtained. Gong et al. [15] used the teaching and learning framework to handling ambiguous but critical samples to prevent inaccurate propagation.

Based on embedded representation, Zhang et al. [45] proposed an inductive embedded label propagation model, which enhances robustness by considering the unfavorable features contained in the samples. Du et al. [11] imposed the existing supervised information on the regularizer to enhance the constraint on the labels, and introduced the maximum correlation criterion to restrain labeling noise. Zhang et al. [42] introduced the idea of double matrix decomposition into the label propagation framework to remove the noise in the data and labels. Dornaika et al. [8] introduced a flexible graph-based semi-supervised learning framework by integrating manifold smoothness, sparse regression and large margin concept. Zoidi et al. [53] proposed positive and negative label propagation framework by considering the additional information that the sample should not be assigned to a label. Wang et al. [33] improved the conventional anchor graph regularization method [20] to process large-scale data problems quickly and accurately.

In order to make the predicted label matrix sufficiently sparse and discriminative, Zhang et al. [43] proposed nonnegative sparse neighborhood propagation (SparseNP). SparseNP adds the  $l_{2,1}$ -norm regular term on the label matrix  $F$ , so that the obtained soft labels are more discriminative. The specific model is as follows:

$$\begin{aligned} \min_F & \text{tr}((F - Y)UD(F - Y)^T) + \text{tr}(F\bar{L}F^T) + \gamma \|F^T\|_{2,1} \\ \text{s.t. } & F \geq 0, e^T F = e^T, \end{aligned} \quad (10)$$

where  $\bar{L} = (I - S)^T(I - S)$ , and  $U$  and  $D$  are diagonal matrices. SparseNP introduces nonnegative and sum-to-one constraints to the label matrix  $F$ ; hence, the resulting soft labels are probability values.

Zhang et al. [44] proposed an improved FME method called discriminative sparse flexible manifold embedding (SparseFME). Compared to FME, SparseFME uses the  $l_{2,1}$ -norm instead of the noise-sensitive Frobenius norm to measure regression residuals, which represents the difference between embedded features and soft labels. Its model is as follows:

$$\begin{aligned} \min_{F, P, b} & \text{tr}(FLF^T) + \text{tr}((F - Y)UD(F - Y)^T) + \alpha \|F^T\|_{2,1} \\ & + \beta (\|P\|_{2,1} + \gamma \|X^T P + eb^T - F^T\|_{2,1}) \\ \text{s.t. } & F \geq 0, e^T F = e^T, \end{aligned} \quad (11)$$

where  $L = I - D^{-1/2}SD^{-1/2}$  is the normalized Laplacian matrix,  $U$  and  $D$  are diagonal matrices,  $P$  is a projection matrix, and  $b$  is a bias vector.

In this paper, we refer to those methods that regularize the  $l_{2,1}$ -norm on the label matrix as sparse label propagation algorithms. There are two main advantages of using the  $l_{2,1}$ -norm: (1)  $l_{2,1}$ -norm is robust to noise; and (2)  $\|F^T\|_{2,1}$

makes many elements in each column of  $F$  tend to 0, and thus the resulting soft labels are more discriminative.

### 3 The proposed method

#### 3.1 Robust and sparse label propagation

Based on the existing graph construction methods and label propagation algorithms, we propose a novel robust and sparse label propagation (RSLP) model. It has three core improvements compared to previous methods. First, real-world data often contain noise, which may reduce the accuracy of the algorithm. Unlike most previous algorithms that execute on the raw data directly, the RSLP algorithm decomposes the raw data into a clean part and a noise part, and uses the clean data to perform graph construction and label propagation. Second, most existing GSSL algorithms learn the structure of the graph in advance, and then perform label propagation on the graph. This two-step strategy does not guarantee that the predefined graph is optimal for label propagation. Conversely, our RSLP algorithm seamlessly combines graph learning and label propagation by jointly estimating the similarity matrix and the predicted label matrix so that the obtained graph is globally optimal. Third, in order to enhance the discrimination of soft labels, we perform  $l_{2,1}$ -norm regularization on the predicted label matrix to make the soft labels of each sample sufficiently sparse.

The proposed RSLP jointly learns the clean data  $C$ , the similarity matrix  $S$  and the predicted label matrix  $F$ . The objective function of RSLP is defined as follows:

$$\begin{aligned} \min_{F, S, C, E} & \operatorname{tr}((F - Y)U(F - Y)^T) + \alpha(\|C\|_* + \|E\|_1) \\ & + (\|F - FS\|_F^2 + \|C - CS\|_F^2 + \beta\|S\|_F^2) + \gamma\|F^T\|_{2,1} \\ \text{s.t. } & X = C + E, S \geq 0, S_{ii} = 0, e^T S = e^T, F \geq 0, e^T F = e^T. \end{aligned} \quad (12)$$

where  $Y$  is the initial label matrix,  $U$  is a diagonal matrix, and  $X$  and  $E$  represent the raw data matrix and the noise data matrix, respectively.  $\alpha$ ,  $\beta$  and  $\gamma$  are balancing parameters.

The term  $\operatorname{tr}((F - Y)U(F - Y)^T)$  is used to make the predicted labels and the real labels as consistent as possible. The term  $(\|C\|_* + \|E\|_1)$  regularizes the nuclear norm on  $C$  and  $l_1$ -norm on  $E$ , so that the obtained clean data are low-rank and the noise data are sparse.  $\|F - FS\|_F^2$  and  $\|C - CS\|_F^2$  represent the label reconstruction error and the sample reconstruction error, respectively. RSLP ensures that the obtained weights are optimal for both data

representation and label propagation by minimizing them jointly. Note that we are using clean data here so that the learned similarity matrix is more accurate. The term  $\|F^T\|_{2,1}$  guarantees that  $F$  is sparse in columns, that is, many elements in the predicted soft label of each sample tend to zero, so the discrimination of soft labels is enhanced.

In the constraints of the objective function of (12),  $X = C + E$  decomposes the raw data into the clean part  $C$  and the noise part  $E$ ,  $S_{ii} = 0$  is used to avoid trivial solution  $S = I$ , and the remainder of the constraints of the objective function in (12) are used to keep the probability interpretation of the obtained similarity matrix and label matrix.

It is worth noting that the proposed RSLP algorithm can be optimized by alternately performing the following three steps.

- (1) Removing noise from the raw data by matrix decomposition: Given the similarity matrix  $S$ , we aim to learn the clean data  $C$  from raw data. The objective function for this step is as follows:

$$\begin{aligned} \min_{C, E} & \|C - CS\|_F^2 + \alpha(\|C\|_* + \|E\|_1) \\ \text{s.t. } & X = C + E. \end{aligned} \quad (13)$$

- (2) Learning the robust similarity matrix: Given the clean data  $C$  and the label matrix  $F$ , this step focus on learning the similarity matrix  $S$  by the following objective function:

$$\begin{aligned} \min_S & \|F - FS\|_F^2 + \|C - CS\|_F^2 + \beta\|S\|_F^2 \\ \text{s.t. } & S \geq 0, S_{ii} = 0, e^T S = e^T. \end{aligned} \quad (14)$$

- (3) Conducting robust and sparse label propagation: with the similarity matrix  $S$  known, we can propagate label information to unlabeled samples. The label matrix  $F$  can be obtained by minimizing the following problem:

$$\begin{aligned} \min_F & \operatorname{tr}((F - Y)U(F - Y)^T) + \|F - FS\|_F^2 + \gamma\|F^T\|_{2,1} \\ \text{s.t. } & F \geq 0, e^T F = e^T. \end{aligned} \quad (15)$$

#### 3.2 Optimization

In this section, the optimization process of RSLP is described. By introducing the auxiliary variable  $Z$  to (12), we can obtain the following equivalent formula:

$$\begin{aligned} \min_{F, S, C, E, Z} & \operatorname{tr}((F - Y)U(F - Y)^T) + \alpha(\|C\|_* + \|E\|_1) \\ & + (\|F - FS\|_F^2 + \|Z - ZS\|_F^2 + \beta\|S\|_F^2) + \gamma\|F^T\|_{2,1} \\ \text{s.t. } & X = C + E, S \geq 0, S_{ii} = 0, e^T S = e^T, F \geq 0, e^T F = e^T, Z = C. \end{aligned} \quad (16)$$



Note that problem (16) can be solved by the altering direction method of multipliers (ADMM) [2]. The augmented Lagrangian function can be obtained by removing the equality constraints on  $X$  and  $Z$ :

$$\begin{aligned} L(C, E, S, F, Z, M_1, M_2) = & \operatorname{tr}((F - Y)U(F - Y)^T) \\ & + \alpha(\|C\|_* + \|E\|_1) \\ & + (\|F - FS\|_F^2 + \|Z - ZS\|_F^2 + \beta\|S\|_F^2) + \gamma\|F^T\|_{2,1} \\ & + \frac{\mu}{2} \left( \|C + E - X + \frac{M_1}{\mu}\|_F^2 + \|C - Z + \frac{M_2}{\mu}\|_F^2 \right) \\ \text{s.t. } & S \geq 0, S_{ii} = 0, e^T S = e^T, F \geq 0, e^T F = e^T. \end{aligned} \quad (17)$$

where  $\mu$  is a penalty parameter, and  $M_1$  and  $M_2$  are the matrices of Lagrangian multipliers.

Solving this involves the following alternating and iterative steps. In each step, one variable is updated while other variables are fixed.

Step 1: Update  $C$ . When other variables are fixed, (17) becomes the following problem:

$$\min_C \alpha\|C\|_* + \mu\|C - G\|_F^2, \quad (18)$$

where  $G = [Z + X - E - (M_1 + M_2)/\mu]/2$ . According to singular value shrinkage, problem (18) has a closed solution  $C = \bar{U} \operatorname{diag}\left(\left(\sigma - \frac{\alpha}{2\mu}\right)_+\right) \bar{V}^T$ , where  $\bar{U} \operatorname{diag}(\sigma) \bar{V}^T$  is the singular value decomposition of  $G$ .

Step 2: Update  $E$ . When other variables are fixed, we have

$$\min_E \alpha\|E\|_1 + \frac{\mu}{2} \left\| E - \left( X - C - \frac{M_1}{\mu} \right) \right\|_F^2, \quad (19)$$

which has a closed solution, i.e.,  $e_{ij} = (|q_{ij}| - \alpha/\mu)_+ \cdot \operatorname{sign}(q_{ij})$ , where  $Q = X - C - \frac{M_1}{\mu}$ .

Step 3: Update  $S$ . With other variables fixed, (17) is reduced to

$$\begin{aligned} \min_S & \|F - FS\|_F^2 + \|Z - ZS\|_F^2 + \beta\|S\|_F^2 \\ \text{s.t. } & S \geq 0, S_{ii} = 0, e^T S = e^T. \end{aligned} \quad (20)$$

Using the same method as in [42], we can get the following equivalent problem by writing  $e^T S = e^T$  into (20):

$$\begin{aligned} \min_S & \left\| \begin{pmatrix} F \\ Z \\ e^T \end{pmatrix} - \begin{pmatrix} F \\ Z \\ e^T \end{pmatrix} S \right\|_F^2 + \beta\|S\|_F^2 \\ \text{s.t. } & S \geq 0, S_{ii} = 0. \end{aligned} \quad (21)$$

Let  $M = [F; Z; e^T]$ , and by taking the derivative with respect to  $S$ , we have

$$\frac{\partial}{\partial S} = -2M^T M + 2M^T M S + 2\beta S. \quad (22)$$

By setting  $\frac{\partial}{\partial S} = 0$ , we can obtain

$$S = \left( \left( M^T M + \beta I \right)^{-1} M^T M \right)_+, S_{ii} = 0. \quad (23)$$

Step 4: Update  $F$ . With other variables fixed, the formula for solving  $F$  is as follows:

$$\begin{aligned} \min_F & \operatorname{tr}((F - Y)U(F - Y)^T) + \|F - FS\|_F^2 + \gamma\|F^T\|_{2,1} \\ \text{s.t. } & F \geq 0, e^T F = e^T. \end{aligned} \quad (24)$$

According to the definition of  $l_{2,1}$ -norm in (4), we have  $\|F^T\|_{2,1} = 2\operatorname{tr}(F B F^T)$ , where  $B_{i,i} = \frac{1}{2\|F_{:,i}\|_2}$ . Problem (24) then turns into

$$\begin{aligned} \min_F & \operatorname{tr}((F - Y)U(F - Y)^T) + \|F - FS\|_F^2 + 2\gamma\operatorname{tr}(F B F^T) \\ \text{s.t. } & F \geq 0, e^T F = e^T. \end{aligned} \quad (25)$$

By taking the derivative with respect to  $F$ , we have

$$\frac{\partial}{\partial F} = 2FU - 2YU + 2FA + 4\gamma FB. \quad (26)$$

where  $A = (I - S)(I - S)^T$ . By setting  $\frac{\partial}{\partial F} = 0$ , we can obtain

$$F = YU(U + A + 2\gamma B)^{-1}. \quad (27)$$

As discussed in [43–45], the solution of  $F$  under the nonnegative constraint and the sum-to-one constraint can be obtained by

$$F = \left( YU(U + A + 2\gamma B)^{-1} \right)_+, F_{ij} = F_{ij}/(e^T F_{:,j}). \quad (28)$$

Step 5: Update  $Z$ . When the other variables are fixed, (17) is reduced to

$$\min_Z \|Z - ZS\|_F^2 + \frac{\mu}{2} \left\| C - Z + \frac{M_2}{\mu} \right\|_F^2. \quad (29)$$

By taking the derivative with respect to  $Z$  and setting  $\frac{\partial}{\partial Z} = 0$ , we can obtain

$$Z = (\mu C + M_2)(2A + \mu I)^{-1}. \quad (30)$$

where  $A = (I - S)(I - S)^T$ .

Based on the alternating optimization process described above, we summarize the details of the proposed RSLP method in Algorithm 1. If the norm of the difference of the label matrix between two adjacent iterations is less than 0.01, we consider that RSLP has achieved convergence and stop iterating.

**Algorithm 1** Robust and sparse label propagation algorithm.

**Input:** Data set  $X = \{X_L \cup X_U\}$ ; Initial label matrix  $Y$ ; Parameters  $\alpha, \beta$  and  $\gamma$ .

- 1 **Initialize:**  $M_1^0 = M_2^0 = 0, E^0 = 0, F^0 = Y, Z^0 = X, k = 0$ .
- 2 **while not converge do**
  - 3 1. Update the matrix  $G$  by
  - 4  $G^{k+1} = [Z^k + X - E^k - (M_1^k + M_2^k) / \mu] / 2$ ;
  - 5 2. Update the clean data  $C$  as
  - 6  $C^{k+1} = \bar{U}^{k+1} \text{diag} \left( \left( \sigma - \frac{\alpha}{2\mu} \right)_+ \right) (\bar{V}^{k+1})^T$ ,
  - 7 where  $\bar{U}^{k+1} \text{diag}(\sigma) (\bar{V}^{k+1})^T$  is the singular value decomposition of  $G^{k+1}$ ;
  - 8 3. Update the matrix  $Q$  by  $Q^{k+1} = X - C^{k+1} - \frac{M_1^k}{\mu}$ ;
  - 9 4. Update the error data  $E$  as  $E^{k+1}$  by setting
  - 10  $e_{ij}^{k+1} = (|q_{ij}^{k+1}| - \alpha / \mu)_+ \cdot \text{sign}(q_{ij}^{k+1})$ ;
  - 11 5. Update the similarity matrix  $S$  as
  - 12  $S^{k+1} = \left( ((M^k)^T M^k + \beta I)^{-1} (M^k)^T M^k \right)_+$ ,  
 $S_{ii}^{k+1} = 0$ ,  
 where  $M^k = [F^k; Z^k; e^T]$ ;
  - 13 6. Update the matrix  $A$  by
  - 14  $A^{k+1} = (I - S^{k+1})(I - S^{k+1})^T$ ;
  - 15 7. Update the predicted label matrix  $F$  as
  - 16  $F^{k+1} = \left( YU(U + A^{k+1} + 2\gamma B^k)^{-1} \right)_+, F_{ij}^{k+1} = F_{ij}^{k+1} / (e^T F_{.j}^{k+1})$ ;
  - 17 8. Update the matrix  $Z$  by
  - 18  $Z^{k+1} = (\mu C^{k+1} + M_2^k) (2A^{k+1} + \mu I)^{-1}$ ;
  - 19 9. Update Lagrange multipliers  $M_1$  and  $M_2$  as
  - 20  $M_1^{k+1} = M_1^k + \mu(C^{k+1} + E^{k+1} - X)$ ,
  - 21  $M_2^{k+1} = M_2^k + \mu(C^{k+1} - Z^{k+1})$ ;
  - 22 10. Check the convergence condition:
  - 23 if  $\|F^{k+1} - F^k\|_F \leq \varepsilon$ , stop;
  - 24 else  $k = k + 1$ ;
- 24 **End while.**

**Output:** The predicted label matrix  $F^* = F^{k+1}$  and the similarity matrix  $S^* = S^{k+1}$ .

## 4 Connection with other related algorithms

In this section, we analyze the connection between RSLP and other related algorithms.

### 4.1 Connection with label propagation algorithms

We discuss the connection between RSLP and several label propagation algorithms, i.e., GFHF, LGC, LNP, and SparseNP.

If we assume that:

(1) the raw data are absolutely clean, and thus there is no need to decompose  $X$ , and (2) the structure of the graph is predefined, that is, the similarity matrix  $S$  is known, then the objective function of RSLP can be transformed into the following form:

$$\min_F \text{tr}((F - Y)U(F - Y)^T) + \text{tr}(F\bar{L}F^T) + \gamma \|F^T\|_{2,1} \quad (31)$$

$s.t. F \geq 0, e^T F = e^T$ .

where  $\bar{L} = (I - S)(I - S)^T$ . Comparing the above formula with the objective function of SparseNP in (10), we can find that they are very similar. Note that  $U$  and  $D$  in (10) and  $U$  in (31) are diagonal matrices, and by making  $U$  in (31) equal to  $UD$  in (10), the objective functions in these two equations are equivalent. Therefore SparseNP can be considered as a special case of RSLP.

Furthermore, if the parameter  $\gamma$  is set to 0 and the constraint conditions in the objective function are removed, formula (31) is converted into the following problem:

$$\min_F \text{tr}(F - Y)U(F - Y)^T + \text{tr}(F\bar{L}F^T) \quad (32)$$

which is similar to the general model of GFHF, LGC, and LNP expressed in formula (9), except that  $L$  in (9) uses the graph Laplacian matrix while (32) uses  $\bar{L} = (I - S)(I - S)^T$ .

### 4.2 Connection with RGC and principle component analysis

We also analyze the connection between the proposed RSLP algorithm and RGC, and principle component analysis (PCA).

We suppose that all samples are labeled, that is,  $F$  is known and is equal to  $Y$ ; thus, there is no need to calculate  $F$ . The model of RSLP in (12) changes to

$$\min_{C, E, S} \alpha (\|C\|_* + \|E\|_1) + \text{tr}(C\bar{L}C^T) + \beta \|S\|_F^2 \quad (33)$$

$s.t. X = C + E, S \geq 0, S_{ii} = 0, e^T S = e^T$ .

where  $\bar{L} = (I - S)(I - S)^T$ . Compared with the RGC model in (7), formula (33) sets the trade-off parameter of  $\text{tr}(C\bar{L}C^T)$  as 1 and makes the coefficients of  $\|C\|_*$  and  $\|E\|_1$  the same. In the constraint condition, there is an extra term  $S_{ii} = 0$  in (33).

The core idea of removing noise in RSLP and RGC is that the clean data are low-rank and the noise is sparse, and it derives from robust principle component analysis

(RPCA) [3]. Manifold RPCA (MRPCA) [29] assumes that the clean data are located on a smooth manifold, and adds a manifold smoothing regularization to the RPCA model. If the similarity matrix  $S$  is precalculated from raw data  $X$ , formula (33) can be reduced to

$$\begin{aligned} \min_{C, E} & \text{tr}(C\bar{L}C^T) + \alpha(\|C\|_* + \|E\|_1) \\ \text{s.t.} & X = C + E. \end{aligned} \quad (34)$$

which is the same as the MRPCA model except that  $\|C\|_*$  and  $\|E\|_1$  have the same parameters.

## 5 Experiments

In this paper, we propose a new robust and sparse label propagation algorithm for semi-supervised classification. In order to verify the effectiveness of the proposed method, we perform experiments on some real datasets and compare our algorithm with other label propagation algorithms in terms of classification performance. The algorithms used for comparison are GFHF [49], LGC [47], SLP [22], SparseNP [43], ALP-MD [42], and SparseFME [45].

### 5.1 Parameter setting

To ensure fairness of comparison, we construct the same  $k$ NN graph for the GFHF, LGC, SLP, SparseNP, and SparseFME. The neighbor parameter  $k$  is fixed at 8, and the bandwidth parameter  $\sigma$  of the Gaussian kernel function is automatically determined by the distance of neighbors [30]. The calculation of  $\sigma$  is as follows:

$$\sigma = \frac{1}{3n} \sum_{i=1}^n d(x_i, x_{i_k}), \quad (35)$$

where  $n$  is the number of samples, and  $d(x_i, x_{i_k})$  represents the distances between sample  $x_i$  and its  $k$ th nearest neighborhood  $x_{i_k}$ . Since ALP-MD and our RSLP algorithm combine the processes of graph construction and label propagation, there is no need to define the graph structure in advance.

SparseNP has one parameter  $\gamma$ , while SparseFME, ALP-MD, and our RSLP have three parameters, i.e.,  $\alpha$ ,  $\beta$ , and  $\gamma$ . For fair comparison, parameters  $\alpha$ ,  $\beta$ , and  $\gamma$  are all selected from  $\{10^{-6}, 10^{-3}, 10^0, 10^3, 10^6\}$ , as in [45] and [23]. The parameters  $\alpha$  of LGC and  $\alpha$  of SLP are both fixed at 0.99 throughout this paper, as suggested in [47]. The elements  $\lambda_l$  and  $\lambda_u$  in the diagonal matrix  $U$  of RSLP are set to  $10^5$  and

0, respectively. We use MATLAB to run all the experiments on a computer with Inter(R) Core(TM) i5-2450M CPU @2.5GHz 8G.

We use the accuracy (ACC) of classifying unlabeled samples to evaluate the performance of each algorithm. The ACC can be calculated as follows:

$$ACC = \frac{1}{u} \sum_{i=1}^u I(y_i, f_i), \quad (36)$$

where  $u$  is the number of unlabeled samples, and  $y_i$  and  $f_i$  are the truth label and predicted label of sample  $x_i$ , respectively.  $I(y_i, f_i)$  is an indicator function:  $I(y_i, f_i) = 1$  if  $y_i = f_i$ ; and  $I(y_i, f_i) = 0$ , otherwise. In all our experiments, we run the label propagation algorithm 10 times to obtain the averaged ACC for different numbers of labeled samples.

### 5.2 Experiments on simple datasets

In this subsection, we compare our RSLP with GFHF, LGC, SLP, SparseNP, ALP-MD and SparseFME on seven simple datasets: Appendicitis, Wine, Sonar, Heart, Seeds, Led7digit and Vehicle. As described in Table 1, these datasets have different number of samples, features and classes. For each dataset, we randomly select some samples from each class as labeled data, and use the rest as unlabeled data. Then we run the label propagation algorithm 10 times to get the averaged ACC as the evaluation standard.

**Table 1** Summary of data sets

Name	# Data	# Feature	# Classes
Appendicitis	106	7	2
Wine	178	13	3
Sonar	208	60	2
Heart	270	13	2
Seeds	210	7	3
Led7digit	500	7	10
Vehicle	846	18	4
Pendigits	1000	16	10
OPtdigits	1000	64	10
USPS	1000	256	10
AR	1680	1024	120
UMIST	574	10304	20
ExtendYaleB	2414	1024	38
COIL20	1440	1024	20



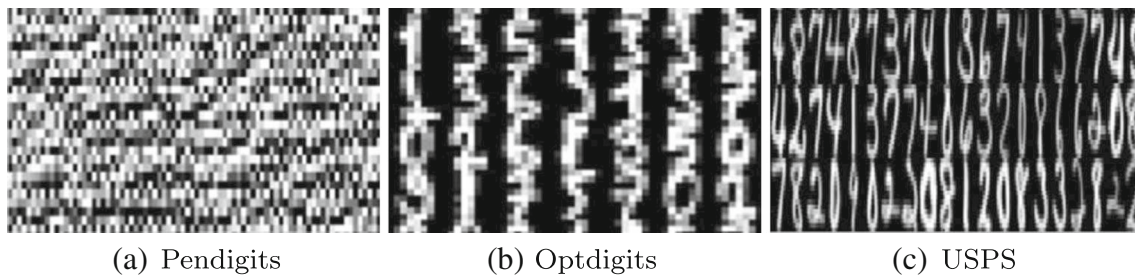
The experimental results are shown in Table 2. The number in parentheses after the dataset name indicates the number of labeled samples of each class, and the best result of each experiment is highlighted in bold. As can be seen from Table 2, the proposed RSLP algorithm achieves the

best results in most cases. GFHF, SparseNP, ALP-MD, and SparseFME achieve the best classification results in one, two, four, and one cases, respectively. In general, the classification accuracy increases with the number of labeled samples. Our RSLP method achieves good classification

**Table 2** Results of data sets from UCI

Datasets(l)	GFHF	LGC	SLP	SparseNP	ALP-MD	SparseFME	RSLP
Wine(2)	65.99	65.58	65.93	65.76	80.87	65.81	<b>84.94</b>
Wine(4)	67.47	69.58	67.47	62.71	80.96	67.23	<b>86.75</b>
Wine(6)	70.00	68.94	70.38	66.38	85.31	70.25	<b>91.25</b>
Wine(8)	67.60	71.49	67.79	65.91	88.51	67.47	<b>91.88</b>
Wine(10)	67.57	71.69	67.50	66.15	89.05	67.50	<b>92.23</b>
Sonar(1)	49.66	50.92	51.26	51.94	51.75	51.07	<b>57.14</b>
Sonar(2)	55.78	56.23	56.57	53.73	54.56	56.72	<b>66.91</b>
Sonar(3)	55.40	55.35	58.02	<b>59.21</b>	57.62	57.08	54.55
Sonar(4)	57.65	55.35	57.95	56.60	56.95	57.90	<b>62.70</b>
Sonar(5)	57.17	56.67	59.70	61.21	59.04	63.13	<b>70.46</b>
Led7digit(2)	61.90	58.17	35.58	<b>62.75</b>	60.15	60.27	61.08
Led7digit(4)	69.78	62.63	48.91	63.98	63.94	68.17	<b>69.94</b>
Led7digit(6)	<b>72.21</b>	64.71	54.00	63.77	65.30	71.66	69.41
Led7digit(8)	71.07	65.19	56.33	63.88	70.26	70.26	<b>73.88</b>
Led7digit(10)	72.00	68.00	59.58	66.50	71.85	71.70	<b>74.33</b>
Heart(2)	58.31	57.14	57.37	52.26	<b>67.44</b>	60.08	66.39
Heart(4)	60.27	59.43	59.77	54.69	<b>67.75</b>	60.42	66.95
Heart(6)	59.26	59.30	59.03	55.74	68.45	59.19	<b>69.96</b>
Heart(8)	60.16	60.83	60.08	59.13	70.79	60.59	<b>72.95</b>
Heart(10)	63.12	63.84	62.88	60.52	72.88	61.00	<b>75.52</b>
Appendicitis(1)	50.67	53.85	46.06	54.04	56.06	<b>61.73</b>	57.31
Appendicitis(2)	64.71	69.90	63.73	62.15	57.65	63.92	<b>79.71</b>
Appendicitis(3)	75.90	71.30	68.10	65.70	82.10	67.04	<b>87.00</b>
Appendicitis(4)	75.00	71.53	73.78	63.67	82.86	67.25	<b>87.35</b>
Appendicitis(5)	81.67	79.06	72.60	71.15	77.50	76.67	<b>88.02</b>
Seeds(1)	65.85	80.68	78.31	81.50	<b>85.07</b>	79.13	84.88
Seeds(2)	83.97	88.24	86.23	86.91	90.59	86.72	<b>91.42</b>
Seeds(3)	89.06	89.55	88.36	87.21	91.39	89.10	<b>91.94</b>
Seeds(4)	88.23	88.99	87.73	85.76	91.31	87.98	<b>92.93</b>
Seeds(5)	88.62	89.59	88.05	86.97	88.51	87.33	<b>92.15</b>
Vehicle(2)	38.91	40.61	41.71	44.01	<b>47.59</b>	41.77	44.38
Vehicle(4)	45.84	49.57	49.88	47.78	42.61	50.11	<b>50.84</b>
Vehicle(6)	47.45	50.02	50.22	52.46	48.75	50.63	<b>54.76</b>
Vehicle(8)	50.81	51.63	55.46	53.69	45.85	53.89	<b>57.99</b>
Vehicle(10)	51.68	51.61	55.48	55.41	47.66	55.50	<b>56.95</b>

The bold type is used to highlight the best results for each experiment.



**Fig. 1** Examples of handwritten digit images from the datasets used in our experiments

results on all seven datasets. This is because RSLP jointly performs graph construction and label propagation, and the resulting soft labels are more discriminative.

### 5.3 Experiments on handwritten digit recognition

In this subsection, we mainly compare the performance of our RSLP and other label propagation algorithms in handwritten digit recognition tasks.

The handwriting image datasets used in the experiment are Pen-Based Recognition of Handwritten Digits Data Set (Pendigits), Optical Recognition of Handwritten Digits (Optdigits) and USPS. Each datasets contains a total of 10 classes from 0 to 9, and their number of features are 16,

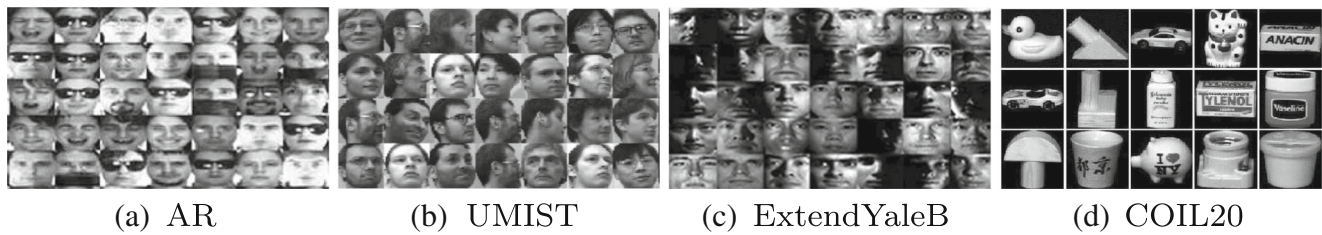
64, and 256, respectively. Figure 1a, b, and c show sample images from Pendigits, Optdigits and USPS, respectively. In order to test the robustness of each algorithm to noise, we corrupt the raw data by randomly selecting a quarter of the features to contaminate with random Gaussian noise. For each dataset, we randomly select 100 images from each of 10 classes (1000 images in total) for the experiments.

The experimental results are shown in Table 3. As can be seen from the table, the classification accuracies of all algorithms increase as the number of labeled samples increase, and RSLP achieves the best classification results in most cases. On Pendigits and USPS datasets, the RSLP algorithm performs significantly better than all other algorithms. On the Optdigits dataset, although the

**Table 3** Results of handwritten digit recognition data sets

Datasets(l)	GFHF	LGC	SLP	SparseNP	ALP-MD	SparseFME	RSLP
Pendigits(2)	24.54	23.38	25.00	23.82	64.33	42.92	<b>65.08</b>
Pendigits(4)	30.74	27.41	30.17	31.28	70.70	50.11	<b>74.28</b>
Pendigits(6)	32.41	29.89	32.57	35.91	68.16	52.11	<b>76.71</b>
Pendigits(8)	35.76	31.74	35.00	37.22	67.07	51.69	<b>73.95</b>
Pendigits(10)	40.05	33.00	39.85	39.09	64.70	52.70	<b>69.65</b>
Optdigits(1)	18.99	27.47	25.11	58.49	53.85	29.39	<b>63.18</b>
Optdigits(2)	37.06	40.10	46.50	63.42	65.62	51.91	<b>69.43</b>
Optdigits(3)	45.80	42.25	56.99	67.85	73.75	59.55	<b>74.23</b>
Optdigits(4)	56.76	49.65	62.25	68.58	75.39	63.55	<b>75.47</b>
Optdigits(5)	62.40	53.35	65.82	71.07	<b>77.34</b>	67.07	76.70
USPS(2)	54.17	47.08	63.25	62.35	58.71	63.67	<b>64.54</b>
USPS(4)	65.89	55.24	71.44	69.11	66.85	70.63	<b>71.91</b>
USPS(6)	71.23	55.93	74.84	74.94	71.50	74.61	<b>75.05</b>
USPS(8)	74.26	59.43	76.57	76.69	74.26	76.33	<b>77.14</b>
USPS(10)	75.88	62.98	78.58	78.55	76.13	78.00	<b>79.13</b>

The bold type is used to highlight the best results for each experiment.



**Fig. 2** Examples of images from the face and object recognition datasets used in our experiments

classification accuracy of RSLP is not as good as that of ALP-MD when the number of labeled samples is five, the RSLP algorithm is superior to the ALP-MD algorithm in other cases. Note that although ALP-MD does not achieve the optimal classification effect, overall it performs well on the Pendigits and Optdigits datasets. From the results obtained on all three datasets, we can see that our RSLP algorithm is the most robust to noise, followed by the ALP-MD algorithm. This is because both algorithms decompose the raw data into a clean part and an error part, thereby reducing the impact of noise on the performance of the algorithm.

## 5.4 Experiments on face and object recognition

In this section, we focus on evaluating the performance of our method in face and object recognition tasks. Three popular benchmark face recognition datasets and an object recognition dataset are used, namely, AR, UMIST, Extend YaleB and COIL20. AR, UMIST, and Extend YaleB contain face image data of 120, 20, and 38 individuals, respectively, and COIL20 contains image data of 20 target objects. Figure 2 shows some samples from these datasets. Following the experimental settings employed in [8] and

**Table 4** Results of face and object recognition data sets

Datasets(l)	GFHF	LGC	SLP	SparseNP	ALP-MD	SparseFME	RSLP
AR(2)	75.79	65.97	85.69	86.19	91.17	83.80	<b>93.04</b>
AR(4)	88.64	73.53	94.12	94.31	97.26	93.35	<b>97.73</b>
AR(6)	92.49	79.51	96.46	96.57	98.63	95.99	<b>98.87</b>
AR(8)	94.44	84.38	97.35	97.35	99.19	97.19	<b>99.22</b>
AR(10)	95.23	86.65	97.67	97.63	99.17	97.54	<b>99.33</b>
UMIST(1)	21.93	21.84	28.21	32.49	20.81	31.77	<b>42.08</b>
UMIST(2)	29.18	35.64	34.38	41.07	34.72	55.75	<b>57.87</b>
UMIST(3)	34.46	46.26	43.85	49.09	59.14	66.07	<b>68.00</b>
UMIST(4)	40.69	57.69	53.40	56.36	67.53	73.83	<b>75.32</b>
UMIST(5)	43.04	58.00	58.76	60.84	71.84	76.92	<b>80.57</b>
ExtendYaleB(2)	70.23	76.83	84.98	84.00	78.86	81.31	<b>85.05</b>
ExtendYaleB(4)	81.45	84.25	88.51	86.63	<b>90.16</b>	86.45	89.75
ExtendYaleB(6)	83.89	86.28	89.73	87.69	<b>91.73</b>	88.00	90.86
ExtendYaleB(8)	84.96	87.43	90.32	88.57	<b>92.32</b>	88.93	92.05
ExtendYaleB(10)	86.58	88.68	91.00	89.44	93.05	89.91	<b>93.63</b>
COIL20(2)	39.68	34.50	42.81	43.59	65.28	<b>71.63</b>	68.93
COIL20(4)	48.06	36.08	53.69	53.79	74.75	<b>79.42</b>	77.82
COIL20(6)	52.75	44.33	59.75	59.89	77.58	80.38	<b>81.48</b>
COIL20(8)	58.14	50.41	61.32	62.54	81.16	81.95	<b>84.26</b>
COIL20(10)	61.58	50.52	66.42	67.53	82.44	83.22	<b>85.78</b>

The bold type is used to highlight the best results for each experiment.

[45], we preprocess these four datasets using principal component analysis (PCA), and set the PCA variability at 98%.

Table 4 shows the experimental results of all tested algorithms on these datasets. We can see that the proposed RSLP algorithm achieves the best classification results in most cases, which further illustrates the advantages of our algorithm. The classification accuracy of RSLP is higher than that of ALP-MD, indicating that the decomposition of the original data and the label propagation on the clean data significantly improve the performance of the algorithm. The superiority of the RSLP algorithm over the SparseNP algorithm reflects the benefits of joint execution graph construction and label propagation.

### 5.5 Convergence analysis

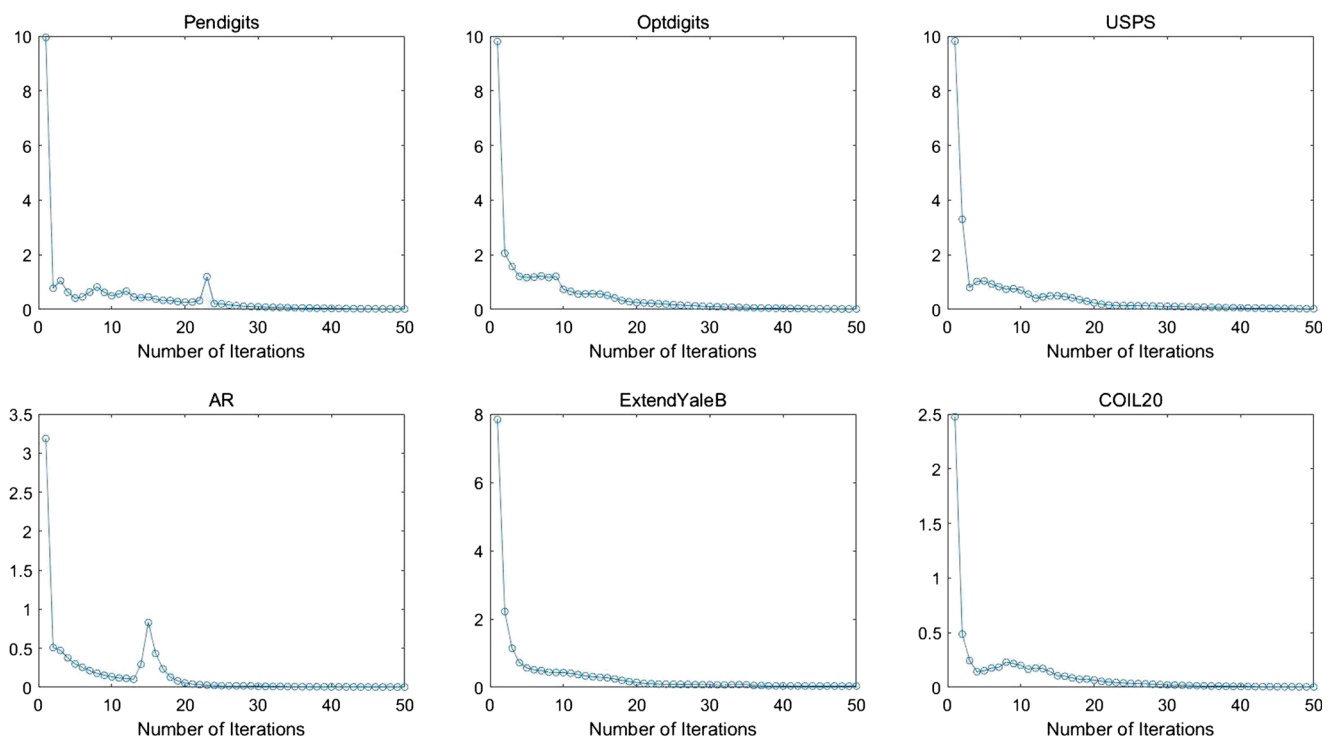
In this subsection, we demonstrate the convergence behavior of RSLP. Six datasets mentioned earlier are applied for this analysis (three handwritten digit recognition datasets, two face recognition datasets, and one object recognition dataset). For each of these datasets, we fix the number of labeled samples per class at 4. Figure 3 shows the convergence curve of the experimental results.

The horizontal axis is the number of iterations, and the vertical axis is the norm of the difference of label matrices obtained by two adjacent iterations, i.e.,  $\|F^{t+1} - F^t\|_F$ . We can see that the difference decreases rapidly with the increase in the number of iterations, that is to say, the optimization process of our algorithm converges very quickly.

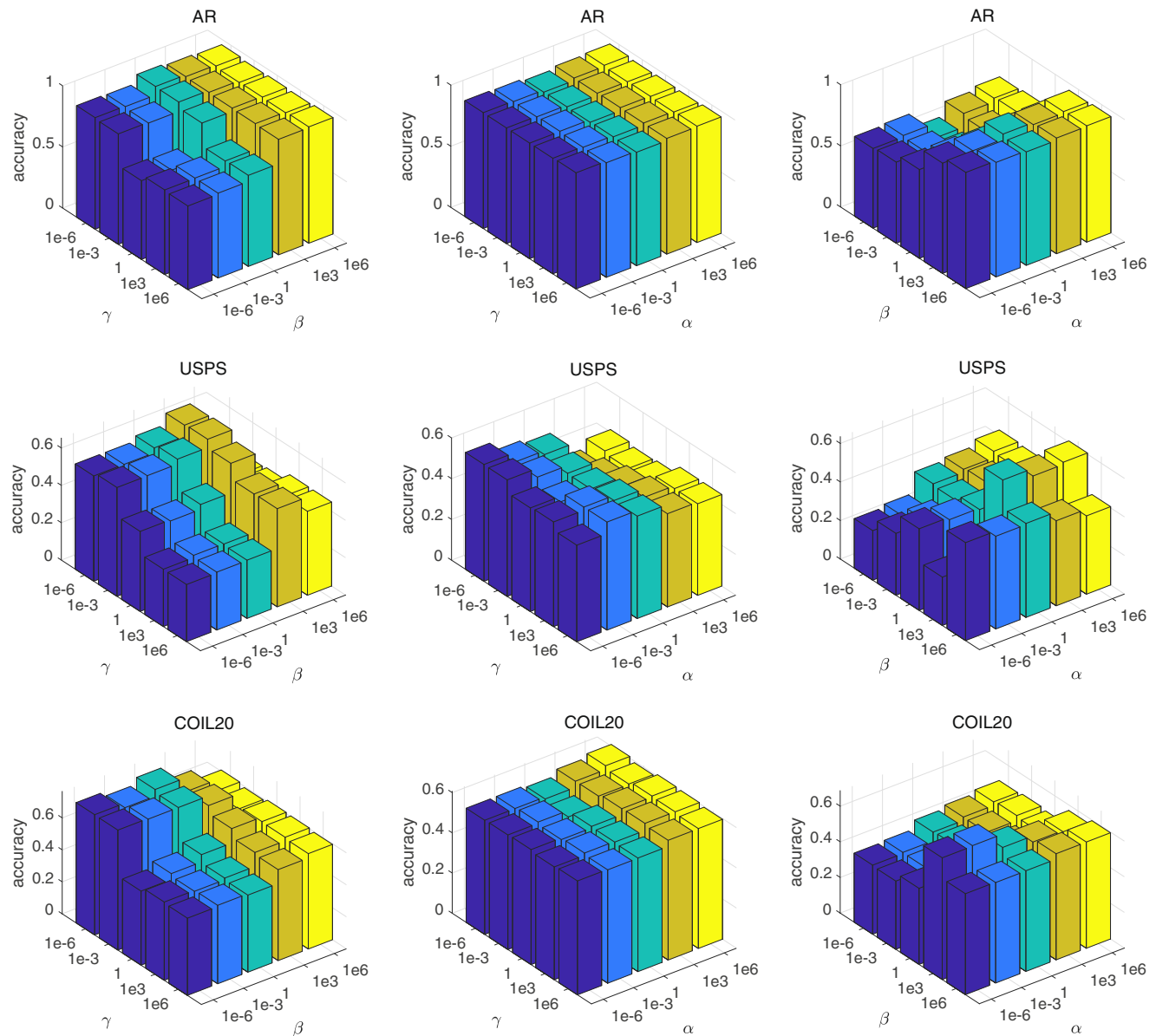
### 5.6 Parameter analysis

Analyzing parameter sensitivity is very important because parameters may affect the performance of algorithm. We use USPS, AR, and COIL20 datasets to analyze the influence of RSLP parameters (i.e.,  $\alpha$ ,  $\beta$ , and  $\gamma$ ) on classification performance. The number of labeled samples per class is set to 4. We fix one parameter to analyze the effects of the other two, that is, we set  $\alpha = 10^6$  to change  $\beta$  and  $\gamma$ ; set  $\beta = 10^6$  to change  $\alpha$  and  $\gamma$ ; and set  $\gamma = 1$  to change  $\alpha$  and  $\beta$ .

The experimental results are shown in Fig. 4. When  $\alpha$  is fixed, and  $\beta$  and  $\gamma$  are small, our algorithm achieves better classification results. When we fix  $\beta$ , it can be seen that RSLP is insensitive to the other two parameters. When  $\gamma$  is fixed, taking larger  $\alpha$  and  $\beta$  helps to obtain higher classification accuracy.



**Fig. 3** Convergence behavior of RSLP on image datasets



**Fig. 4** Accuracy of RSLP under various parameters

## 6 Conclusion

In this paper, we propose a novel GSSL model termed robust and sparse label propagation algorithm, it aims at enhancing the robustness to noise and improving classification performance of label propagation methods. RSLP simultaneously removes noise, constructs a similarity matrix, and predicts unknown labels. Specifically, RSLP decomposes the raw data into a low-rank clean part and a sparse error part. In order to ensure that the learned similarity matrix is globally optimal for representation and classification, RSLP jointly performs label propagation and graph construction in the clean data space. Furthermore, the proposed RSLP performs  $l_{2,1}$ -norm regularization on

the predicted label matrix to enhance the discrimination of soft labels. Experimental results on simple datasets and image recognition datasets show that our RSLP algorithm is superior to existing algorithms in terms of classification accuracy. In our future work, we will study how to extend RSLP into an inductive model to process out-of-samples more effectively.

**Acknowledgements** This work was supported by National Natural Science Foundation of China grant 61573266.

## References

1. Appice A, Guccione P, Malerba D (2017) A novel spectral-spatial co-training algorithm for the transductive classification



- of hyperspectral imagery data. *Pattern Recogn* 63:229–245. <https://doi.org/10.1016/j.patcog.2016.10.010>
2. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers
  3. Candès EJ, Li X, Ma Y, Wright J (2011) Robust principal component analysis. *J ACM* 58(3):11. <https://doi.org/10.1145/1970392.1970395>
  4. Chen D, Wang W, Gao W, Zhou Z (2018) Tri-net for semi-supervised deep learning. In: Lang J (ed) *Proceedings of the twenty-seventh international joint conference on artificial intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden*, ijcai.org, 2014–2020. <https://doi.org/10.24963/ijcai.2018/278>
  5. Cheng B, Yang J, Yan S, Fu Y, Huang TS (2010) Learning with  $\ell^1$ -graph for image analysis. *IEEE Trans Image Process* 19(4):858–866. <https://doi.org/10.1109/TIP.2009.2038764>
  6. Cheng G, Han J, Lu X (2017) Remote sensing image scene classification: Benchmark and state of the art. *Proc IEEE* 105(10):1865–1883. <https://doi.org/10.1109/JPROC.2017.2675998>
  7. Das D, Lee CSG (2020) A two-stage approach to few-shot learning for image recognition. *IEEE Trans Image Process* 29:3336–3350. <https://doi.org/10.1109/TIP.2019.2959254>
  8. Dornaika F, Trounitsi YE (2019) Joint sparse graph and flexible embedding for graph-based semi-supervised learning. *Neural Netw* 114:91–95. <https://doi.org/10.1016/j.neunet.2019.03.002>
  9. Dornaika F, Weng L (2019) Sparse graphs with smoothness constraints: Application to dimensionality reduction and semi-supervised classification. *Pattern Recogn* 95:285–295. <https://doi.org/10.1016/j.patcog.2019.06.015>
  10. Dornaika F, Dahbi R, Bosaghzadeh A, Ruichek Y (2017) Efficient dynamic graph construction for inductive semi-supervised learning. *Neural Netw* 94:192–203. <https://doi.org/10.1016/j.neunet.2017.07.006>
  11. Du B, Xinyao T, Wang Z, Zhang L, Tao D (2018) Robust graph-based semisupervised learning for noisy labeled data via maximum correntropy criterion. *IEEE Trans Cybern* 49(4):1440–1453. <https://doi.org/10.1109/TCYB.2018.2804326>
  12. Fang X, Han N, Wong WK, Teng S, Wu J, Xie S, Li X (2019) Flexible affinity matrix learning for unsupervised and semisupervised classification. *IEEE Trans Neural Netw* 30(4):1133–1149. <https://doi.org/10.1109/TNNLS.2018.2861839>
  13. Fei L, Xu Y, Fang X, Yang J (2017) Low rank representation with adaptive distance penalty for semi-supervised subspace classification. *Pattern Recogn* 67:252–262. <https://doi.org/10.1016/j.patcog.2017.02.017>
  14. Fox-Roberts P, Rosten E (2014) Unbiased generative semi-supervised learning. *J Mach Learn Res* 15(1):367–443
  15. Gong C, Tao D, Liu W, Liu L, Yang J (2017) Label propagation via teaching-to-learn and learning-to-teach. *IEEE Trans Neural Netw Learn Syst* 28(6):1452–1465. <https://doi.org/10.1109/TNNLS.2016.2514360>
  16. Hua Z (2020) Yang, Y. Node influence-based label propagation algorithm for semi-supervised learning. *Neural Comput Appl*, Qiu, H. <https://doi.org/10.1007/s00521-020-05078-0>
  17. Kang Z, Pan H, Hoi SCH, Xu Z (2019) Robust graph learning from noisy data. *IEEE Trans Syst Man Cybern* 1–11. <https://doi.org/10.1109/TCYB.2018.2887094>
  18. Kingma DP, Mohamed S, Rezende DJ, Welling M (2014) Semi-supervised learning with deep generative models. In: *Advances in neural information processing systems*, pp 3581–3589
  19. Liu C, Hsiao W, Lee C, Chang T, Kuo T (2016) Semi-supervised text classification with universum learning. *IEEE Trans Cybern* 46(2):462–473. <https://doi.org/10.1109/TCYB.2015.2403573>
  20. Liu W, He J, Chang SF (2010) Large graph construction for scalable semi-supervised learning. In: *ICML*, pp 679–686. <https://icml.cc/Conferences/2010/papers/16.pdf>
  21. Ma L, Ma A, Ju C, Li X (2016) Graph-based semi-supervised learning for spectral-spatial hyperspectral image classification. *Pattern Recognit Lett* 83:133–142. <https://doi.org/10.1016/j.patrec.2016.01.022>, advances in Pattern Recognition in Remote Sensing
  22. Nie F, Xiang S, Liu Y, Zhang C (2010) A general graph-based semi-supervised learning with novel class discovery. *Neural Comput and Applic* 19(4):549–555. <https://doi.org/10.1007/s00521-009-0305-8>
  23. Nie F, Xu D, Tsang IWH, Zhang C (2010) Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Trans Image Process* 19(7):1921–1932. <https://doi.org/10.1109/TIP.2010.2044958>
  24. Nie F, Shi S, Li X (2020) Semi-supervised learning with auto-weighting feature and adaptive graph. *IEEE Trans Knowl Data Eng* 32(6):1167–1178. <https://doi.org/10.1109/TKDE.2019.2901853>
  25. Peng Y, Lu BL, Wang S (2015) Enhanced low-rank representation via sparse manifold adaption for semi-supervised learning. *Neural Netw* 65:1–17. <https://doi.org/10.1016/j.neunet.2015.01.001>, <https://www.sciencedirect.com/science/article/pii/S0893608015000027>
  26. Qiao L, Chen S, Tan X (2010) Sparsity preserving projections with applications to face recognition. *Pattern Recognit* 43(1):331–341. <https://doi.org/10.1016/j.patcog.2009.05.005>. <https://www.sciencedirect.com/science/article/pii/S0031320309001964>
  27. Qiao L, Zhang L, Chen S, Shen D (2018) Data-driven graph construction and graph learning: A review. *Neurocomputing* 312:336–351. <https://doi.org/10.1016/j.neucom.2018.05.084>
  28. Roweis ST, Saul LK (2000) Nonlinear dimensionality reduction by locally linear embedding. *Science* 290(5500):2323–2326. <https://doi.org/10.1126/science.290.5500.2323>
  29. Shahid N, Kalofolias V, Bresson X, Bronstein M, Vandergheynst P (2015) Robust principal component analysis on graphs. In: 2015 IEEE international conference on computer vision (ICCV), pp 2812–2820
  30. de Sousa CA (2015) An overview on the gaussian fields and harmonic functions method for semi-supervised learning. In: 2015 international joint conference on neural networks (IJCNN). IEEE, pp 1–8. <https://doi.org/10.1109/IJCNN.2015.7280491>
  31. Wang F, Zhang C (2008) Label propagation through linear neighborhoods. *IEEE Trans Knowl Data Eng* 20(1):55–67. <https://doi.org/10.1109/TKDE.2007.190672>
  32. Wang J, Wang F, Zhang C, Shen H, Quan L (2009) Linear neighborhood propagation and its applications. *IEEE Trans Pattern Anal Mach Intell* 31(9):1600–1615. <https://doi.org/10.1109/TPAMI.2008.216>
  33. Wang M, Fu W, Hao S, Tao D, Wu X (2016) Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Trans Knowl Data Eng* 28(7):1864–1877. <https://doi.org/10.1109/TKDE.2016.2535367>
  34. Wei D, Yang Y, Qiu H (2020) Improving self-training with density peaks of data and cut edge weight statistic. *Soft Comput* :1–16. <https://doi.org/10.1007/s00500-020-04887-8>
  35. Weng L, Dornaika F, Jin Z (2016) Graph construction based on data self-representativeness and laplacian smoothness. *Neurocomputing* 207:476–487. <https://doi.org/10.1016/j.neucom.2016.05.021>
  36. Wu D, Shang M, Luo X, Xu J, Yan H, Deng W, Wang G (2018) Self-training semi-supervised classification based on density peaks of data. *Neurocomputing* 275:180–191. <https://doi.org/10.1016/j.neucom.2017.05.072>
  37. Wu H, Prasad S (2018) Semi-supervised deep learning using pseudo labels for hyperspectral image classification. *IEEE Trans Image Process* 27(3):1259–1270. <https://doi.org/10.1109/TIP.2017.2772836>

38. Xie Y, Wang H, Yu B, Zhang C (2020) Secure collaborative few-shot learning. *Knowl-Based Syst* 106157:203. <https://doi.org/10.1016/j.knosys.2020.106157>
39. Yan S, Wang H (2009) Semi-supervised learning by sparse representation. pp 792–801. <https://doi.org/10.1137/1.9781611972795.68>
40. Yu J, Kim SB (2018) Consensus rate-based label propagation for semi-supervised classification. *Inform Sci* 465:265–284. <https://doi.org/10.1016/j.ins.2018.06.074>
41. Zhang C, Wang S, Li D, Yang J, Chen H (2015) Prior class dissimilarity based linear neighborhood propagation, vol 83, pp 58–65. <https://doi.org/10.1016/j.knosys.2015.03.011>
42. Zhang H, Zhang Z, Li S, Ye Q, Zhao M, Wang M (2018a) Robust adaptive label propagation by double matrix decomposition. In: 2018 24th International conference on pattern recognition (ICPR). pp 2160–2165. <https://doi.org/10.1109/ICPR.2018.8545594>
43. Zhang Z, Zhang L, Zhao M, Jiang W, Liang Y, Li F (2015) Semi-supervised image classification by nonnegative sparse neighborhood propagation. In: Proceedings of the 5th ACM on international conference on multimedia retrieval, pp 139–146. <https://doi.org/10.1145/2671188.2749292>
44. Zhang Z, Zhang Y, Li F, Zhao M, Zhang L, Yan S (2017) Discriminative sparse flexible manifold embedding with novel graph for robust visual representation and label propagation. *Pattern Recogn* 61:492–510. <https://doi.org/10.1016/j.patcog.2016.07.042>
45. Zhang Z, Li F, Jia L, Qin J, Zhang L, Yan S (2018b) Robust adaptive embedded label propagation with weight learning for inductive classification. *IEEE Trans Neural Netw Learn Syst* 29(8):3388–3403. <https://doi.org/10.1109/TNNLS.2017.2727526>
46. Zheng Y, Zhang X, Yang S, Jiao L (2013) Low-rank representation with local constraint for graph construction. *Neurocomputing* 122:398–405
47. Zhou D, Bousquet O, Lal TN, Weston J, Olkoph BS (2004) Learning with local and global consistency. In: Advances in neural information processing systems, vol 16
48. Zhou ZH, Li M (2010) Semi-supervised learning by disagreement. *Knowl Inf Syst* 24(3):415–439. <https://doi.org/10.1007/s10115-009-0209-z>
49. Zhu X, Ghahramani Z, Lafferty JD (2003) Semi-supervised learning using gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 912–919
50. Zhu X (2005) Lafferty J, Semi-supervised learning with graphs, Rosenfeld R
51. Zhuang L, Zhou Z, Gao S, Yin J, Lin Z, Ma Y (2017) Label information guided graph construction for semi-supervised learning. *IEEE Trans Image Process* 26(9):4182–4192. <https://doi.org/10.1109/TIP.2017.2703120>
52. Zhuang L, Zhou Z, Gao S, Yin J, Lin Z, Ma Y (2017) Label information guided graph construction for semi-supervised learning. *IEEE Trans Image Process* 26(9):4182–4192. <https://doi.org/10.1109/TIP.2017.2703120>
53. Zoidi O, Tefas A, Nikolaidis N, Pitas I (2018) Positive and negative label propagations. *IEEE Trans Circ Syst Video Technol* 28(2):342–355. <https://doi.org/10.1109/TCSVT.2016.2598671>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Zhiwen Hua** received his B.S. in Information and Computer Science from North University of China, Shanxi, China in 2018. He is a graduate at Xidian University, Xi'an, China at present. His current research interests include semi-supervised learning, data mining and data analysis.



**Youlong Yang** received his B.S. and M.S. in Mathematics from Shaanxi Normal University, Xi'an, China in 1990 and 1993 respectively, and Ph.D. in System Engineering from Northwestern Polytechnical University, Xi'an, China in 2003. Since 2004, he has been with the faculty at Xidian University, Xi'an, China. His research interests include Machine learning, Statistical data analysis and Probabilistic graphical models.