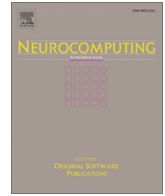




Contents lists available at ScienceDirect

## Neurocomputing

journal homepage: [www.elsevier.com/locate/neucom](http://www.elsevier.com/locate/neucom)

## A semi-supervised learning algorithm via adaptive Laplacian graph

Yuan Yuan, Xin Li, Qi Wang\*, Feiping Nie

School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P.R. China

## ARTICLE INFO

## Article history:

Received 9 October 2019

Revised 15 July 2020

Accepted 22 September 2020

Available online xxxx

Communicated by Liu Wei

## Keywords:

Semi-supervised learning

Adaptive graph

Pattern recognition

## ABSTRACT

Many semi-supervised learning methods have been developed in recent years, especially graph-based approaches, which have achieved satisfactory performance in the practical applications. There are two points that need to be noticed. Firstly, the quality of the graph directly affects the final classification accuracy. However, graph-based algorithms mostly use  $k$ -Nearest Neighbor to construct the graph. And the directly constructed graph is inaccurate due to outliers and erroneous features in the data. Secondly, the amount of labeled data is a small part of all data. It cannot be guaranteed that all categories of data are included in the labeled data and the labels of data are not totally correct in practice. To address the aforementioned problems, we propose a new graph-based semi-supervised method named ALGSSL via adaptive Laplacian graph. In the algorithm, we adaptively update the graph to reduce the sensitiveness of the construction of initial graph. Meanwhile, we use the regularization parameters to set confidence on existing labels, which can reduce the impact of the error labels on the result and discover the new category. Experiments on three toy datasets and nine benchmark datasets demonstrate the proposed method can achieve good performance.

© 2020 Elsevier B.V. All rights reserved.

## 1. Introduction

With the rapid development of computer network technology and the diversification of people's access to data, the information available to people is rapidly growing exponentially. This is a huge challenge in how to get what people need with huge amounts of data. Machine learning is one of the most important ways to solve the problem.

Traditional machine learning can be divided into two categories: unsupervised learning and supervised learning. In unsupervised learning, samples are clustered according to the similarity among the data. Without labeled samples, the results of these algorithms are uncertain. In supervised learning, a model is established to predict unlabeled data by learning from a large amount of labeled data. The trained model will have weak generalization ability and be easily over-fitting when the number of labeled data is too small. In many practical applications, unlabeled data is abundant and readily available while there are a few labeled data. And the labeling process is time consuming and expensive. It has become one of the most concerned issues to use limited labeled data and abundant unlabeled data to improve performance in machine learning. This motivates a hot research area of semi-supervised learning (SSL) [1].

Most of the semi-supervised learning algorithms are based on some manifold or clustering assumptions. These assumptions are reasonable because the data points from the same manifold generally share the same label in the majority of real-world problems. One of the typical algorithms is the graph-based semi-supervised algorithm [2–4]. Constructing a graph associated with the weight matrix is an significant step in graph-based SSL. This graph can characterize the geometry of the data, which is usually a  $k$ -Nearest Neighbor ( $k$ NN) graph. It defines a weight matrix on the nearest  $k$  neighbors found throughout the data [5,6]. However, the traditional  $k$ NN graph is very sensitive to noises. The noises affect the quality of the constructed graph and further affect the classification accuracy. It is important to construct a method that does not rely on the initial graph in semi-supervised learning.

Moreover, it is inaccurate to construct a graph directly, because there are outliers and some redundant features in the data [7–10]. A lot of scholars have discovered the shortcomings of using initial graph and tried to improve it in many areas. In multi-view clustering, Cai et al. [11] and Li et al. [41] considered that the original data contained noises and outlying entries, and they made graphs unreliable and inaccurate. In data clustering, Zhang et al. [12] thought that the initial affinity matrix was not optimal in practical application and Chen et al. [13] took graph learning as the part of clustering. In hyperspectral feature selection, Wang et al. [14] thought using the original feature space directly was not appropriate because of the noisy and irrelevant features. In hyperspectral

\* Corresponding author.

E-mail address: [crabwq@gmail.com](mailto:crabwq@gmail.com) (Q. Wang).

imagery, Ziemann et al. [15] believed that it was necessary to build an adaptive graph so as to handle the particular characteristics of hyperspectral images. Therefore, it is necessary to adaptively update the graph after learning an initial graph in semi-supervised learning.

Lots of semi-supervised learning algorithms [16–18] assume that labels of data are correct and use it as a priori knowledge to classify unlabeled samples. This is correct and feasible on existing datasets. However, it is unavoidable that some data categories are wrong when the data is labeled in practice. Since a very small portion of all data is marked, some categories are not included in the labeled data. Serious mistakes can occur if we still believe that the existing labels are correct and categorized according to them. It is necessary to solve the above two problems in actual applications. Nie et al. [19] proposed an algorithm called GGSSL to solve the problem of error labels by using regular items. Experiments show that this algorithm can correct the error labels, but it does not solve the problem that the graph-based semi-supervised algorithms are sensitive to the initial graph. Therefore, the focus of our paper is to reduce the sensitivity of the algorithm to the initial graph and to have some processing capability for the error labels.

In this paper, we propose a new graph-based semi-supervised learning algorithm named ALGSSL via adaptive Laplacian graph. The contributions of this paper are as follows. 1) A method that can construct naturally sparse and computationally efficient graphs is used for semi-supervised learning, which is insensitive to noise. 2) This approach adaptively updates the graph in semi-supervised learning. This can correct the inaccuracy in the direct graph construction method caused by the redundant and error features of the data and the outliers, and reduce the sensitivity to the initial image. 3) The approach uses the regularization parameters as the confidence of the labels. By changing the regularization parameters, the algorithm does not classify the unlabeled data exactly according to the original label. Therefore, this can increase tolerance of the error labels, and discover the new category at the same time.

## 2. Related work

According to the principle of algorithm, the existing semi-supervised learning algorithms based on traditional machine learning can be roughly divided into three categories.

The first type of algorithms take the generation model as a classifier and treat the probability that unlabeled data belongs to each class as a set of missing parameters. Then the expectation maximization (EM) algorithm is used for label estimation and model parameter estimation in general.

Miller applied the EM algorithm to semi-supervised learning [20]. Nigam [21] successfully used the algorithm based on the generated model in the text classification. Fujino et al. [22] extended the hybrid generation model by adding an offset correction term and optimizing it using the principle of maximizing entropy. One disadvantage of these approaches is that errors in the early prediction process accumulate as the iteration progresses. One possible solution is to adopt a heuristic strategy, which is to label only unlabeled samples with confidence greater than a certain threshold. Another disadvantage is that the methods cannot analyze its convergence under general conditions. This type of methods are still popular because of its conceptual and computational simplicity.

The second type of algorithms are the approaches based on cooperative training. Such approaches assume that the dataset has two independent and fully redundant views. In the process of learning, multiple learners are used to select several unlabeled examples with high reliability to label each other to update the model. The dataset is hard to satisfy the requirement that it has a sufficiently redundant view in the real problems.

It was originally proposed by Blum and Mitchell in 1998 [23]. Goldman and Zhou [24] proposed a cooperative training algorithm without sufficient redundant views. Although the algorithm no longer requires full redundant views, it has many limitations on classifier types. They also used 10-fold cross validation several times to better estimate tag confidence, which consumed a lot of time. In order to solve the problems in the above algorithm, Zou and Li [25] proposed a Tri-training algorithm which does not require different types of classifiers. The difference between this algorithm and the above algorithms is mainly that three classifiers are used. In [7] Zhan and Zhang adapt co-training strategy to implement semi-supervised multi-label learning.

The third type of algorithms are the approaches based on graph. These make direct or indirect use of the manifold assumptions. This means that when the samples are close on the manifold, they may have the same label. They usually build a graph based on a measure of similarity of existing data. The nodes in the graph correspond to (labeled or unlabeled) data, and the edges are the similarity between data. Then they define the objective function to be optimized and use the decision function as a regularization term to find the optimal model parameters.

Zhou et al. [26] used local and global consistency to classify unlabeled samples based on label information from labeled samples. Belkin et al. [27] proposed a regularization framework that can take advantage of the geometric nature of the boundary distribution. Karasuyama et al. [28] proposed an adaptive edge weighting strategy to optimize the edge weights by minimizing the local linear reconstruction error to obtain higher quality graph and enhance the performance of the algorithm.

In recent studies, deep learning has been applied to semi-supervised learning with good results. Kingma et al. [29] proposed a semi-supervised learning framework based on a generation model, which transformed the semi-supervised learning problem into a classification problem with a specific lack of input task. Chang et al. [30] used the generated confrontation network to generate training samples and then used the simulated samples to train the classifier. Atarashi et al. [31] proposed a generative model of the labeling process in crowdsourcing. It leverages unlabeled data effectively by introducing latent features and a data distribution. Meng et al. [32] proposed a method called SGDNNMF (semi-supervised graph regularized deep NMF with bi-orthogonal), which can obtain the mapping between the low-dimensional learning representation and the original high-dimensional data.

## 3. Method description

### 3.1. Notation description

All matrices are written in uppercase, while the elements of matrices are written in lowercase. For matrix  $A$ , the  $i$ -th row and  $ij$ -th element are represented in  $a_i$  and  $a_{ij}$ . The trace of  $A$  is denoted as  $\text{Tr}(A)$ . The Frobenius norm of matrix  $A$  is denoted as  $\|A\|_F$ .

### 3.2. Proposed method

Given a data point set  $X = \{x_1, x_2, \dots, x_l, x_{l+1}, \dots, x_n\}$  and its corresponding label category set  $C = \{1, 2, \dots, c\}$ . Here the first  $l$  points have labels and their label  $p_i (1 \leq i \leq l)$  belongs to  $C$ , and the rest of points  $x_{l+1}, \dots, x_n$  are unlabeled.  $F$  refers to the soft label matrix. The weight matrix of initial graph is denoted as  $W$ . A classic semi-supervised graph-based loss function as shown in Eq. (1).

$$\min_F \text{tr}(F^T \tilde{L} F), \quad (1)$$

where  $\tilde{L}$  is the normalized Laplacian matrix for  $W$  and  $\tilde{L} = \tilde{D} - \tilde{W}$ .  $\tilde{W}$  is the normalized weights of the initial graph. It can be calculated by  $\tilde{W} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$ .  $\tilde{D}$  is the normalized degree matrix of  $W$  and a diagonal matrix with entries  $\tilde{d}_i = \sum_j \tilde{w}_{ij}$ . Eq. (1) is equivalent to  $\sum_{i,j=1}^n \tilde{s}_{ij} \|f_i - f_j\|_F^2$ , and it aims to set more similar soft labels for two points with larger weight values.

Most datasets are now manually annotated. It is inevitable that a small part of the data is marked incorrectly. To solve this problem and make the algorithm more robust, Nie et al. add a fitting term in [19], which measures the difference between the soft labels and the initial labels. From this, the objective function is obtained as follows:

$$\min_F \text{Tr}(F^T \tilde{L} F) + \text{Tr}(F - Y)^T U \tilde{D} (F - Y), \quad (2)$$

where  $Y$  is the initial label matrix of the labeled data.  $U$  is a diagonal matrix with entries  $\mu_i > 0$ , where  $\mu_i$  is a regularization parameter for data  $x_i$ . By setting the regularization parameters  $\mu_i$ , we can control the similarity between the soft labels and the initial labels. Therefore, the algorithm can get correct results when the initial label is wrong.

As the size of the dataset becomes larger, not only will there be error labels in the dataset, but there may also be unlabeled categories in the unlabeled data. Set the initial label matrix to try to solve this problem. Assume that the initial label of all unlabeled samples is  $c + 1$ . By setting the regularization parameters of the unlabeled data, the label of the unlabeled data may be the same as the original label ( $c + 1$  category), that is, the data of the new category is found.

Based on the above assumption, the label category set is  $\tilde{C} = \{1, 2, \dots, c + 1\}$ . Define  $F = [f_1^T, f_2^T, \dots, f_n^T]^T \in \mathbb{R}^{n \times (c+1)}$  as the soft label for points, where  $f_i \in \mathbb{R}^{c+1}$  ( $1 \leq i \leq n$ ) are row vectors and  $0 \leq f_{ij} \leq 1$  ( $1 \leq i \leq n, 1 \leq j \leq c + 1$ ). The initial label matrix is denoted as  $Y = [y_1^T, y_2^T, \dots, y_n^T]^T \in \mathbb{R}^{n \times (c+1)}$ , where  $y_i \in \mathbb{R}^{c+1}$  ( $1 \leq i \leq n$ ) are row vectors. For labeled data, if the label of  $X_i$  is  $j$  then  $y_{ij} = 1$  or else  $y_{ij} = 0$ . For unlabeled data,  $y_{ij} = 1$  when  $j = c + 1$ , otherwise  $y_{ij} = 0$ .

Furthermore, graph-based semi-supervised learning models strongly depend on the initial graph. If the quality of the initial graph is low, the effect of the semi-supervised learning method will be poor. And the models usually use  $k$ NN graph method to initialize the graph, which is sensitive to discrete points and noise. To solve this problem, we not only need an initial graph method that is more suitable for semi-supervised learning, but also need to reduce the sensitivity of the model to the initial graph. The method of initializing graph is detailed in Section 3.4.

Learning an adaptive graph is an effective way to reduce the dependency of the model on the initial graph. The weight matrix of the adaptive graph to be learned is denoted as  $S$ . To avoid that some rows of  $S$  are always 0 during the iteration, we constrain the sum of each row of  $S$  to 1 (i.e.  $\sum_j s_{ij} = 1$ ). Therefore, the final objective function about  $S$  and  $F$  can be defined as

$$\min_{S \geq 0, \sum_j s_{ij} = 1, F} \|S - W\|_F^2 + \lambda \left( \text{Tr}(F^T \tilde{L}_S F) + \text{Tr}(F - Y)^T U \tilde{D}_S (F - Y) \right), \quad (3)$$

where  $\lambda$  is penalty parameter,  $\tilde{L}_S$  is the normalized Laplacian matrix for  $S$  and  $\tilde{L}_S = \tilde{D}_S - \tilde{S}$ .  $\tilde{S}$  is the normalized weights of the graph to be learned. It can be calculated by  $\tilde{S} = D_S^{-\frac{1}{2}} S D_S^{-\frac{1}{2}}$ .  $\tilde{D}_S$  is the normalized degree matrix of  $S$  and a diagonal matrix with entries  $\tilde{d}_i = \sum_j \tilde{s}_{ij}$ .

### 3.3. Optimization algorithm

We apply the alternative optimization approach to solve this problem. When  $S$  is fixed, the cost function Eq. (3) becomes

$$\min_F \text{Tr}(F^T \tilde{L}_S F) + \text{Tr}(F - Y)^T U \tilde{D}_S (F - Y). \quad (4)$$

Let the derivative of Eq. (4) be zero, i.e.,

$$2\tilde{L}_S F + 2U \tilde{D}_S (F - Y) = 0. \quad (5)$$

To make it easier to simplify formulas, we define the variable  $I_\alpha$  which is a diagonal matrix with entries  $\alpha_i = 1/(1 + \mu_i)$  ( $i = 1, 2, \dots, n$ ). Then  $F$  can be derived as

$$\begin{aligned} F &= (\tilde{L}_S + U \tilde{D}_S)^{-1} U \tilde{D}_S Y = (I - \tilde{D}_S^{-1} \tilde{S} + U)^{-1} I_\alpha U Y \\ &= (I - \tilde{D}_S^{-1} \tilde{S} I_\alpha)^{-1} (I - I_\alpha) Y. \end{aligned} \quad (6)$$

When  $F$  is fixed, the cost function (3) becomes

$$\min_{S \geq 0, \sum_j s_{ij} = 1} \|S - W\|_F^2 + \lambda \text{Tr}(F^T \tilde{L}_S F). \quad (7)$$

To facilitate the solution, we rewrite Eq. (7) in another form:

$$\min_{S \geq 0, \sum_j s_{ij} = 1} \sum_{i,j} (s_{ij} - w_{ij})^2 + \lambda \sum_{i,j} \|f_i - f_j\|_2^2 s_{ij}. \quad (8)$$

Notice that Eq. (8) is independent for each  $i$  and each  $j$ . So the problem can be solved by solving each  $i$  individually:

$$\min_{s_{ij} \geq 0, \sum_j s_{ij} = 1} \sum_j (s_{ij} - w_{ij})^2 + \lambda \sum_j \|f_i - f_j\|_2^2 s_{ij}. \quad (9)$$

Let  $v_{ij} = \|f_i - f_j\|_2^2$  be the element of matrix  $V$ ,  $v_i$  is the row vector of the matrix  $V$  (and similarly for  $s_i$  and  $w_i$ ). Eq. (9) can be rewritten as:

$$\min_{s_{ij} \geq 0, \sum_j s_{ij} = 1} \left\| s_i - \left( w_i - \frac{\alpha}{2} v_i \right) \right\|_2^2. \quad (10)$$

Eq. (10) can be solved by the algorithm proposed in [33].

### 3.4. Learning an initial graph

It is well known that the  $k$ NN graph is very classic, but it is sensitive to noise points. Therefore, it is very important to use a new graph that is more suitable for semi-supervised learning.

We use the method is proposed in [34] to initialize graph. The initial graph  $W$  learned by this method is naturally sparse and computationally efficient for graph-based semi-supervised. This method uses the L2-norm of each row of  $W$  as the regularization to learn the affinity values of  $W$  [34]. Based on this method, we can get an initial graph with  $c$  connected components.

Here,  $w_i$  represents the row vector of the weight matrix  $W$ . Constraint  $w_i$  has  $c$  nonzero values, that is,

$$\|w_i\|_0 = c. \quad (11)$$

The problem we have to solve is

$$\min_{w_i^T \mathbf{1} = 1, w_i \geq 0, w_{ii} = 0} \sum_{j=1}^n \|x_i - x_j\|_2^2 w_{ij} + \gamma \sum_{j=1}^n w_{ij}^2. \quad (12)$$

We define

$$e_{ij} = \|x_i - x_j\|_2^2. \quad (13)$$

Without loss of generality, we suppose  $e_{i1}, e_{i2}, \dots, e_{in}$  are arranged in ascending order. Then, Eq. (12) can be simplified as

$$\min_{w_i^T \mathbf{1} = 1, w_i \geq 0, w_{ij} = 0} \frac{1}{2} \left\| w_i + \frac{e_i}{2\gamma} \right\|_2^2. \quad (14)$$

Using the method of Laplace transform, KKT condition and the constraint  $w_i^T \mathbf{1} = 1$ , we can derive

$$\hat{w}_{ij} = \left( -\frac{e_i}{2\gamma} + \eta \right)_+, \quad (15)$$

$$\eta = \frac{1}{c} + \frac{1}{2c\gamma} \sum_{j=1}^n e_{ij}. \quad (16)$$

According to Eq. (11), we know  $\hat{w}_{i,c} > 0$  and  $\hat{w}_{i,c+1} = 0$ , that is,

$$-\frac{e_{i,c}}{2\gamma} + \eta > 0, -\frac{e_{i,c+1}}{2\gamma} + \eta = 0. \quad (17)$$

According to Eq. (16) and Eq. (17), we get the maximal  $\gamma$ :

$$\gamma = \frac{1}{2} e_{i,c+1} - \frac{1}{2} \sum_{j=1}^n e_{ij}. \quad (18)$$

Then, we get the final expression of the weights  $w_{ij}$  as follows:

$$w_{ij} = \begin{cases} \frac{e_{i,c+1} - e_{ij}}{ce_{i,c+1} - \sum_{h=1}^c e_{ih}} & j \leq c \\ 0 & j > c \end{cases}. \quad (19)$$

The algorithm ALGSSL is summarized in Alg. 1.

---

**Algorithm 1.** ALGSSL

---

**Input:** the data point set  $X = \{x_1, x_2, \dots, x_n\}$ , the soft label matrix  $F$ , the label matrix  $Y$ .

1: Learn an initialize graph  $S$  according to Eq. (19).

2: Calculate the diagonal matrix  $D_S$ , for each  $0 \leq i \leq n$ ,  
 $d_i = \sum_j s_{ij}$ .

3: Compute the normalized weight matrix  $\tilde{S} = D_S^{-\frac{1}{2}} S D_S^{-\frac{1}{2}}$ , and the normalized Laplacian matrix  $\tilde{L}_S = \tilde{D}_S - \tilde{S}$ .

4: **while** no convergence **do**

5: Update  $F$  according to Eq. (6).

6: Update  $S$  according to Eq. (10).

7: **end while**

8: **return** the updated soft label matrix.

**Output:** the updated label matrix.

---

## 4. Experiments

In this section, we first verify the effect of the regularization parameter  $\alpha$  on the result labels with some toy examples. Then we validate the algorithm on several benchmark datasets.

In the experiments, the value of the regularization parameter  $\alpha_i$  in Eq. (6) is as follows. We set the parameter  $\alpha_i$  corresponding to the labeled data  $x_i$  to the same value  $\alpha_l$  and the parameter  $\alpha_i$  corresponding to the unlabeled data  $x_i$  to the same value  $\alpha_u$ .

### 4.1. Experimental on toy examples

The role of the regularization parameter  $\alpha$  is discussed in the first two toy experiments. Then the graph constructed method we used and  $k$ -Nearest Neighbor graph are compared in the third toy experiment.

The first toy example is created with two categories data and eight labeled points as shown in Fig. 1a. In this example, We label four points in category 1 and category 2, respectively. And one of the labeled points for each category is wrong, that is, six points

with correct label and two points with wrong label. It's inevitable that people make mistakes in marking. So we set up error labels in our experiment to simulate this situation. The algorithm should have the ability to handle these mislabels rather than fully trust that the labels are correct.

We found that adjusting parameter  $\alpha_i$  can adjust the confidence of the algorithm on existing labels. We let  $\alpha_l = 0$  and  $\alpha_u = 1$ , and the experimental results are shown in Fig. 1(b). The algorithm classifies the data according to the wrong label, and it is conceivable that the correct rate is low. Then we change the value of parameter  $\alpha_l$  to 0.99, and the experimental results are shown in Fig. 1(c). At this time, the algorithm does not completely classify according to the existing labels, but refers to the relevant features of the graph. Therefore, the algorithm can eliminate the interference of the wrong labels and correctly classify the data.

After a lot of experiments, we got the following conclusions about the parameter  $\alpha_l$  and the existing labels. For labeled data,  $\alpha_l = 0$  means the result labels are the same as the initial labels and will not change,  $\alpha_l > 0$  means that the result labels are different from the initial labels. When the initial labels are not completely trusted,  $\alpha_l$  is set to a larger value. In contrast, when the initial labels are completely correct,  $\alpha_l$  is set to zero.

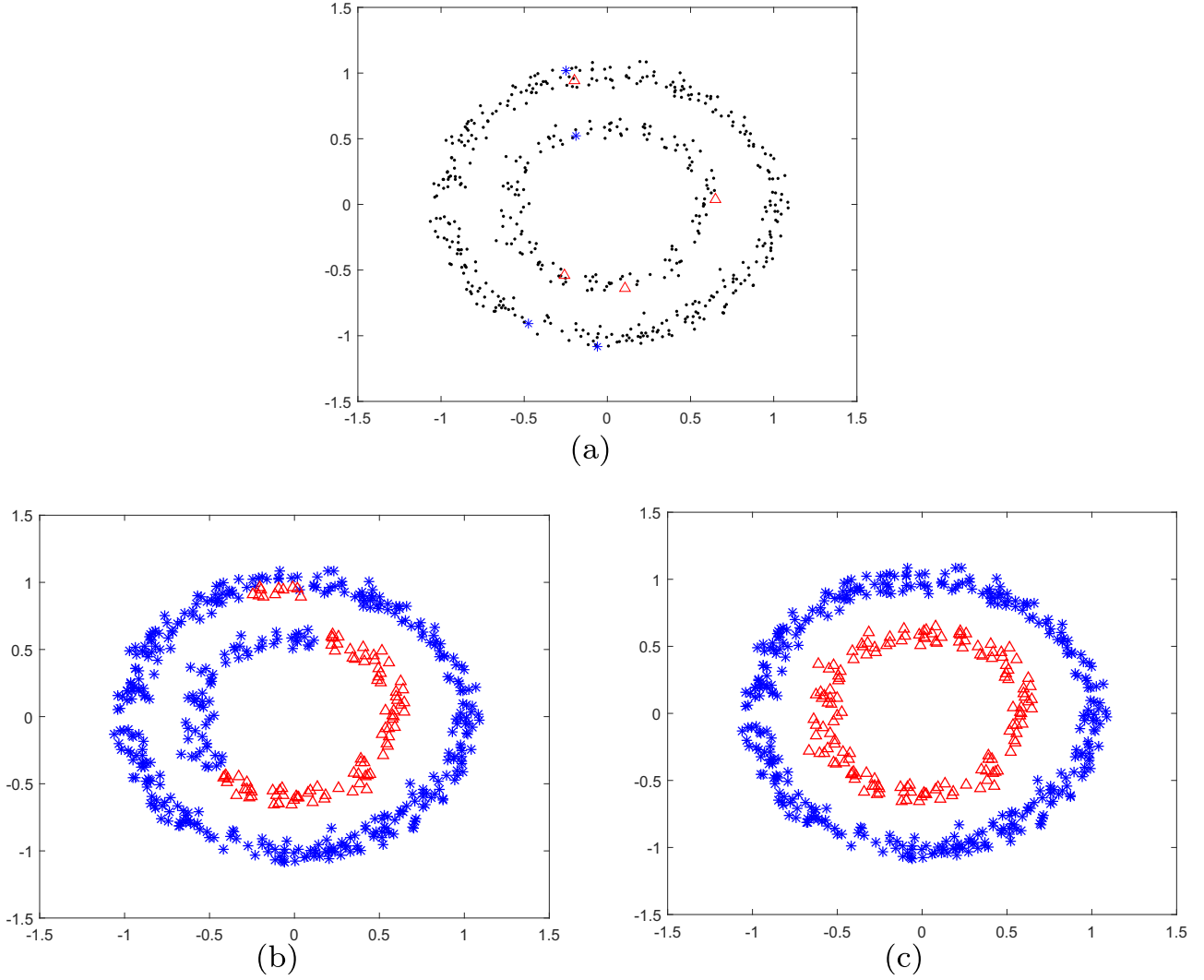
In Fig. 2 (a), the toy example is created with three categories and two labels. We simulated the lack of categories that would occur in a real dataset in this toy example. The number of unlabeled data is much greater than that of labeled data. In practical applications, the number of unlabeled data is far greater than the number of labeled data. There may be undiscovered new categories in unlabeled data. This toy example corresponds to the above situation.

We set the initial category of the unlabeled sample to the new category. So we can use parameter  $\alpha_u$  to change the confidence of the algorithm for the initial label of unlabeled data. We first let the parameter  $\alpha_u$  be 1 and  $\alpha_l$  be 1, then the result is shown in Fig. 2(b). From this result, we find that the algorithm classifies the data of the three categories into two categories (the initial number of categories is two), and the algorithm error rate is higher. We change the value of  $\alpha_u$  to 0.99999 and the result is shown in Fig. 2(c). From the results we can see that the algorithm under the new parameter settings can find data that does not belong to the initial categories and assign it to the new category.

The relationship between parameter  $\alpha_u$  and unlabeled data is summarized below. For unlabeled data,  $\alpha_u = 1$  means that the algorithm classifies the data according to known categories, just like most algorithms set.  $\alpha_u < 1$  means that the algorithm has the ability to discover data that does not belong to the existing categories and treat it as a new category. This approach is reasonable, it allows us to discover unknown categories in the data and effectively improve the robustness of the algorithm.

The above two experiments show that the algorithm can effectively find new categories and correct error tags by adjusting the regularization parameter  $\alpha$ .

The third dataset is similar to the second, except that different proportions of Gaussian noise are added to the third dataset when it is generated. This dataset has 500 points and there are three categories. We add 10%, 20% and 30% Gaussian noise to the dataset respectively, as shown in the Fig. 3. In this toy experiment, 100 samples are randomly selected from the dataset as labeled data. ALGSSL and KNN-ALGSSL are tested in these datasets. Here, KNN-ALGSSL refers to ALGSS method based on  $k$ -Nearest Neighbor graph, and its settings are the same as ALGSSL except the graph constructed method. We separately perform these two algorithms on the datasets four times to obtain the average error rates. The experimental results are shown in the Table 1. According to the results, we can know that the proposed constructed method is less



**Fig. 1.** (a) The toy example is created with two categories and eight labels. The known labels contains two error labels. (b) Classification results with  $\alpha_l = 0$  and  $\alpha_u = 1$ . (c) Classification results with  $\alpha_l = 0.99$  and  $\alpha_u = 1$ . Points represent unmarked data, triangles represent the points with category 1 and stars represent the points with category 2.

affected by noise than  $k$ -Nearest Neighbor graph, and it is more suitable for semi-supervised learning tasks.

Fig. 4(b)–(d) shows the ALGSSL algorithm's initial weight graph denoted as  $W_1$ , its iterative adaptive weight graph denoted as  $W_2$ , and the difference between the two graphs ( $W_2 - W_1$ ) in a dataset with 10% noise added. We compared the initial weight graph of the dataset noised by 10% (Fig. 4) and the original dataset (Fig. 4(a)). It can be known that the noise points are distributed outside the block diagonal of the weight graph. In Fig. 4(d), the dark blue points represent negative points, which means that the values of the points in the adaptive graph are smaller than the values of the corresponding points in the initial graph. Furthermore, the color of the points outside the block diagonal of graph (Fig. 4(d)) is mostly dark blue, that is, the adaptive graph is less noisy than the initial graph. In other words, the adaptive graph reduces the influence of noise on the proposed ALGSSL and improves its accuracy through adaptive learning.

#### 4.2. Experimental on benchmark datasets

To evaluate the performance of our algorithm, We compared our algorithm with some SSL algorithms, including TSVM [35], meanS3VM-iter [36], LDS [37], LGC [26], AEW [28], LEAD [38],

SaGSSL [39] and GGSSL (include GGSSL1 and GGSSL2) [19]. In addition, 1 NN (1 -Nearest Neighbor) algorithm and SVM algorithm are used as baseline algorithms.

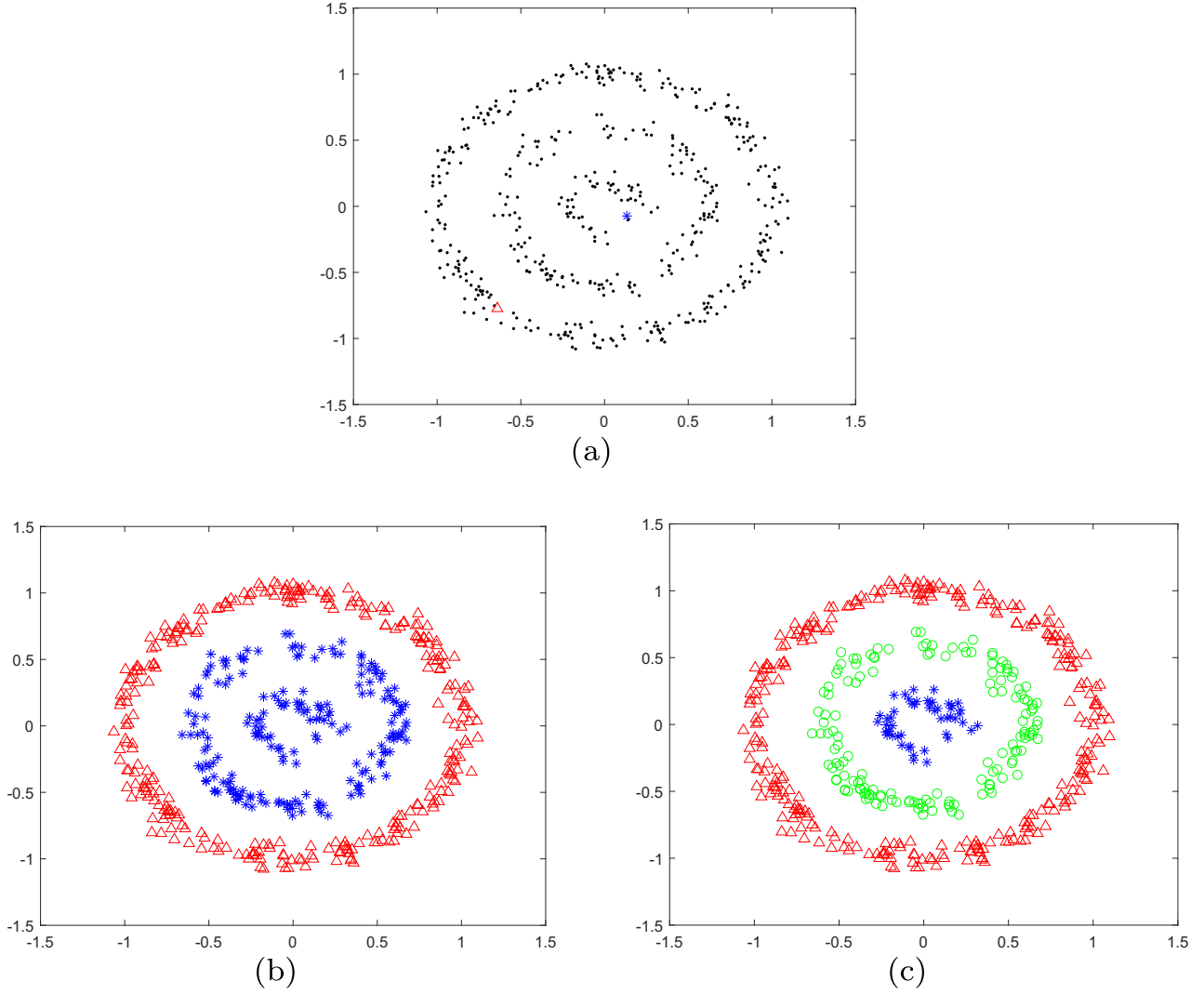
##### 4.2.1. Datasets descriptions

The benchmark datasets we used are provided by [40]. There are six datasets in the benchmark datasets. For each dataset, the benchmark provides 12 splits, each of which has 100 labeled data. In data sources, three of the datasets (g241c, g241d and Digit1) are man-made data and the rest (USPS, COIL and BCI) are from real data. In the distribution of data points, the first two datasets (g241c and g241d) are composed of multiple Gaussian distributions but not manifold structures, and the three datasets (Digit1, USPS and COIL) are manifold structures. The details of the benchmark datasets are shown in the Table 2.

##### 4.2.2. Experimental setups

The parameters  $C_1$  and  $C_2$  in meanS3VM-iter are respectively set to 100 and 0.1, and the Gaussian kernel parameter is selected via ten-fold cross-validation. For the LGC method, the number of neighbor  $k \in \{1, 2, \dots, 9\}$  and the regularization parameter  $\mu \in \{10^{-3}, 10^{-2}, \dots, 1\}$ . For the AEW method, the parameter





**Fig. 2.** (a) The toy example is created with three categories and two labels. (b) Classification results with  $\alpha_u = 1$  and  $\alpha_l = 1$ . (c) Classification results with  $\alpha_u = 0.99999$  and  $\alpha_l = 1$ . Points represent unmarked data, triangles represent the points with category 1, stars represent the points with category 2 and circles represent the points with category 3.

$a_m \in [0.1\bar{d}, 10\bar{d}]$  where  $\bar{d}$  is average of Euclidean distances of all the sides. The parameters  $C_1, C_2$  and  $\beta$  in the LEAD method are respectively fixed to 1, 0.01 and 0.02. The kernel parameter of the LEAD method is selected via fivefold cross-validation among  $\{0.25, 0.5, 1, 2, 4\}\delta$  where  $\delta$  is the average distance between samples. For the SaGSSL method,  $\gamma_A$  and  $\gamma_1$  are obtained by 5-fold cross validation. The search region of  $\gamma_A$  and  $\gamma_1$  is  $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4\}$ . The parameter  $K$  is selected in the region  $[10, 60]$  to construct graphs. The parameter  $\sigma$  about computing the weights of edge is selected in the region  $\{2^{-4}\delta, 2^{-3}\delta, 2^{-2}\delta, 2^{-1}\delta, \delta, 2^1\delta, 2^2\delta, 2^3\delta, 2^4\delta\}$ , where  $\delta$  is the average of the distance between samples. The number of graphs  $T$  is set to 54. The setting of the kernel parameter is same as that of the LEAD method. For the GGSSL method, the parameter  $k \in \{5, 6, \dots, 10\}$  and the parameter  $\sigma = \sqrt{-\frac{\bar{d}}{\ln(s)}}$  where  $s \in \{10^{-4} \times \frac{1}{k}, 10^{-3} \times \frac{1}{k}, 10^{-2} \times \frac{1}{k}, 10^{-1} \times \frac{1}{k}, \frac{1}{k}\}$  and the definition of variable  $\bar{d}$  is the same as above. For our algorithm, the parameter  $k \in \{3, 4, \dots, 16\}$  and the parameter  $\lambda \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ .

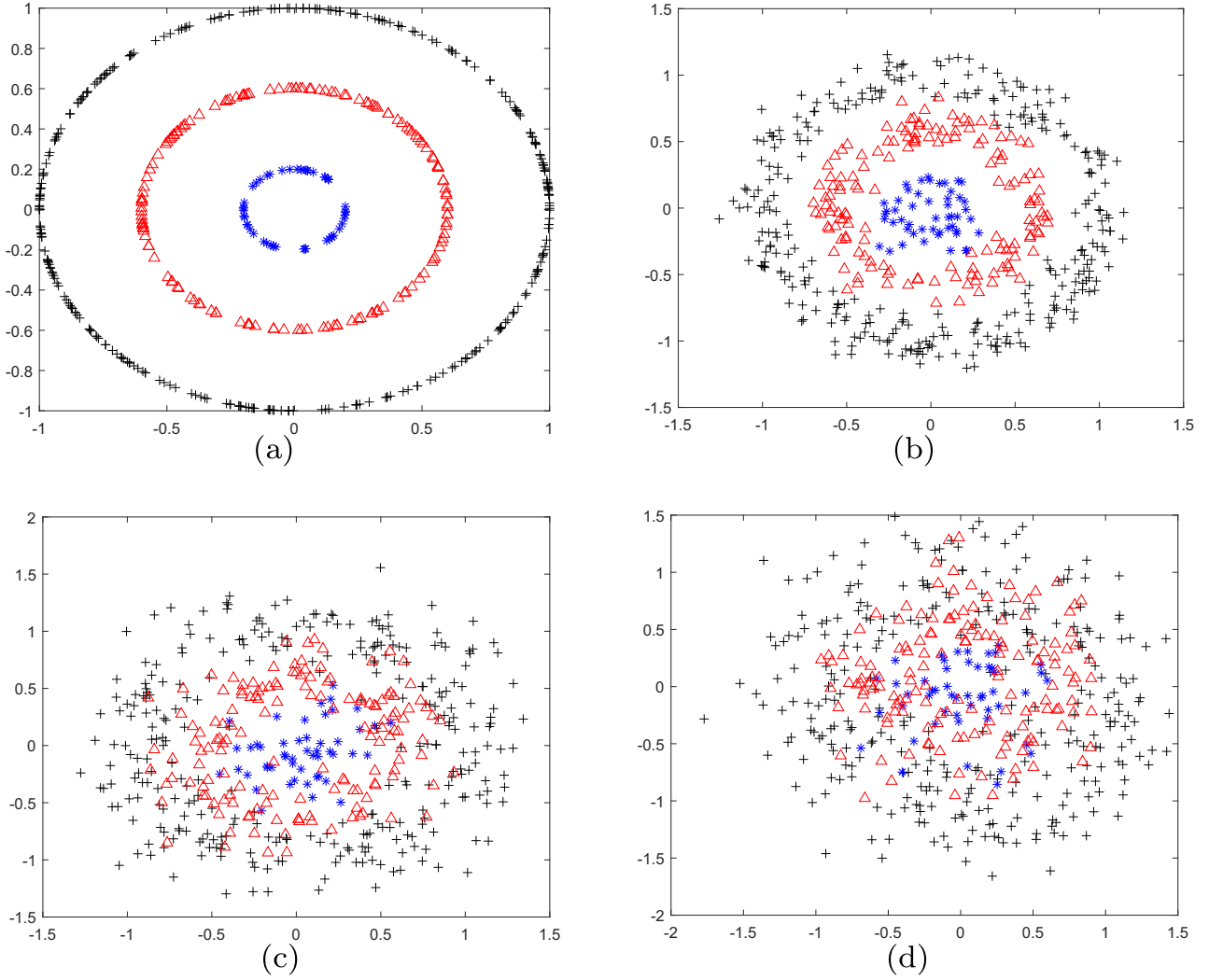
Both GGSSL and our algorithm set the regularization parameters  $\alpha_l$  to 0 and  $\alpha_u$  to 0.99999.

#### 4.2.3. Experimental results

For each dataset, the algorithms run on its 12 splits to obtain the average test error values. Table 3 shows the average errors of the algorithms in 6 datasets. The best results for the six datasets are highlighted in bold. Meanwhile, the convergence curves of the proposed algorithm are shown in Fig. 5.

KNN-ALGSSL refers to the ALGSSL method based on  $k$ -Nearest Neighbor graph. According to the experimental results, the average error rate of KNN-ALGSSL method is 4% higher than that of ALGSSL method in six datasets. This shows that the initial graph method we used is more suitable than  $k$ NN graph.

In the g241c dataset, the first three algorithms with low average error rates are LDS, TSVM and SVM. The LDS algorithm performs optimally with an average error rate of 18.04%. In the g241d dataset, the error rates of the algorithm TSVM, meanS3VM-iter and LDS are very similar. Among them, the TSVM algorithm has the lowest average error rate. The average error rate of the ALGSSL algorithm



**Fig. 3.** (a) The third toy dataset consists of 500 samples in three categories. (b) Add 10% Gaussian noise to the dataset. (c) Add 20% Gaussian noise to the dataset. (d) Add 30% Gaussian noise to the dataset.

**Table 1**

The average errors (%) with 100 samples of third toy dataset. The values in the table represent the average error rate of the algorithm running 4 times under the datasets.

Method	10% noise	20% noise	30% noise
KNN-ALGSSL	4.35	21.65	27.55
ALGSSL	1.75	15.50	24.40

is 40.17%, ranking eighth in the dataset g241c. In g241d, the average error rate of the ALGSSL algorithm is 37.1%, ranking seventh.

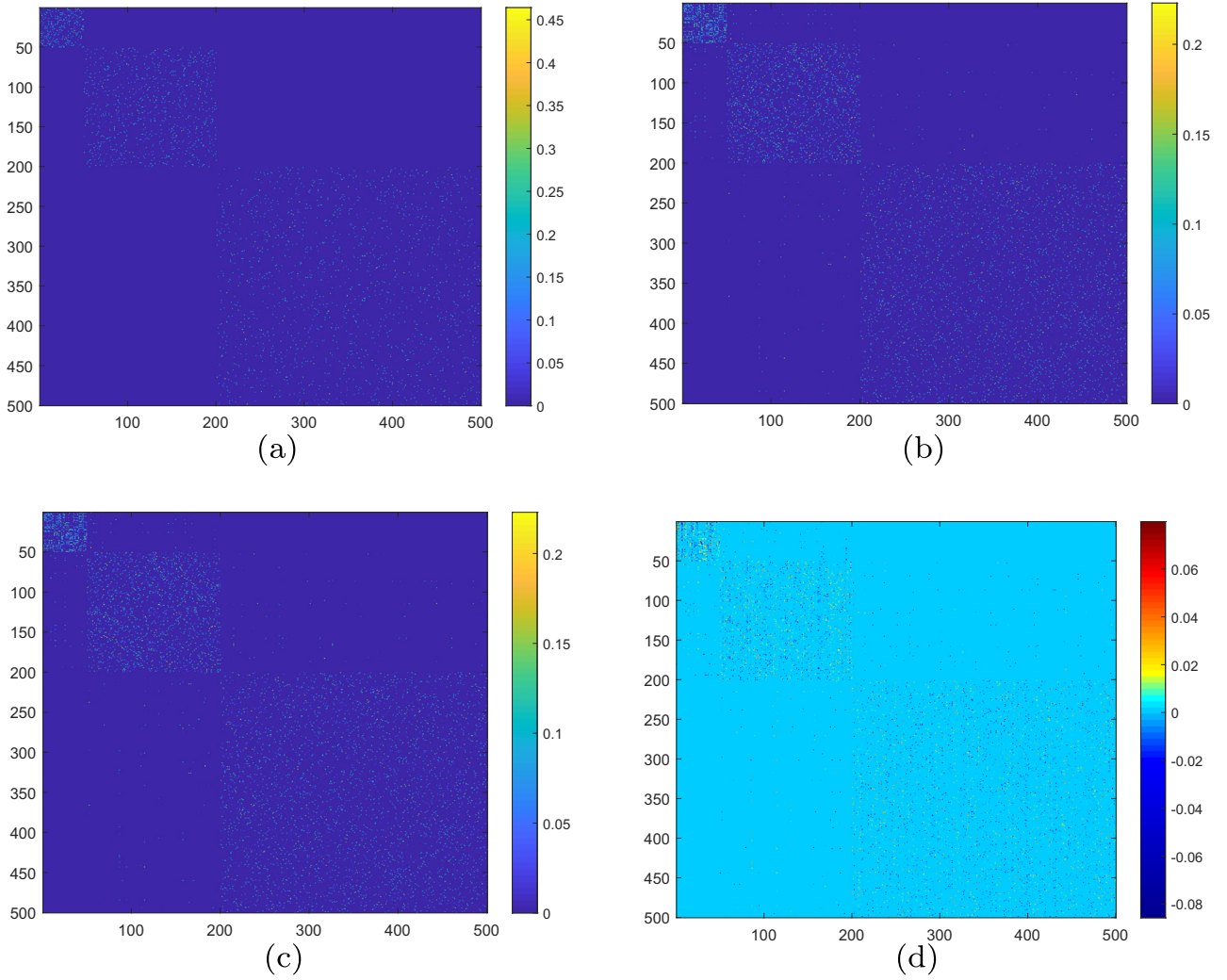
In Digit1 and COIL, the three algorithms ALGSSL, GGSSL2 and GGSSL1 rank first, second and third respectively. The average error rate of ALGSSL in Digit1 is 1.71%, and that of ALGSSL in COIL is 8.66%. In the dataset USPS, the ALGSSL algorithm, the LDS algorithm and the GGSSL2 algorithm ranked the top three with an average error rate of 4.34%, 4.96% and 5.29% respectively.

In the last dataset BCI, the ALGSSL algorithm performs best with an average error rate of 34.29%, followed by the SVM algorithm with an average error rate of 34.31%. The third is the meanS3VM-iter algorithm, whose average error rate is 35.12%.

**Discussion:** In the first two datasets (g241c and g241d), the experimental results of our algorithm ranked seventh and eighth

respectively in the 12 algorithms, that is, the ranking is below the middle. At the same time, it can be seen from Table 3 that graph-based semi-supervised algorithms (such as AEW, LEAD, SaGSSL, and GGSSL) do not perform well on the first two datasets, and their average error rates are almost 40%. By studying the properties of datasets, the two datasets have the following settings at the initial generation: these hold the cluster assumption but do not hold the manifold assumption. Fig. 6 shows the two-dimensional projection images of the first two datasets (g241c and g241d), where ‘.’ represents the data of the first category and ‘+’ represents the data of the second category. It can be seen from Fig. 6 that some of the distribution areas of the two categories are coincident. At the same time, graph-based semi-supervised learning algorithms mostly classify data on data-generated graphs. The artificially set conditions of the datasets impose great inconvenience on the graph-based approach.

The ALGSSL algorithm has the lowest average errors in dataset Digit1, USPS and COIL with flow structure, among which Digit1 is artificial dataset and USPS and COIL are real dataset. This indicates that our algorithm can perform well in datasets of flow structures. In addition, it also performs very well in dataset BCI. This implies that our algorithm has good scalability on non-obvious manifold structure datasets.



**Fig. 4.** (a) The initial weight graph  $W_0$  of the third toy dataset. (b) The initial weight graph  $W_1$  of the dataset with 10% noise. (c) The adaptive weight graph  $W_2$  after algorithm iteration. (d)  $W_2 - W_1$ .

**Table 2**

The details of the benchmark datasets.

ID	Dataset	Classes	Dimension	Points	Artificial
1	g2421c	2	241	1500	Yes
2	g2421d	2	241	1500	Yes
3	Digit1	2	241	1500	Yes
4	USPS	2	241	1500	No
5	COIL	6	241	1500	No
6	BCI	2	117	400	No

#### 4.3. Experimental on large-scale datasets

In order to verify the effectiveness of the ALGSSL algorithm on the large-scale datasets with more categories, it is tested on the following three datasets.

Mnist is database of handwritten digits, and all images have been size-normalized and fixed to  $28 \times 28$ . MSRA25 dataset contains 1,700 gray-scale images of faces from 12 people, and each image is cropped to a  $16 \times 16$  size. Olivetti face dataset consists of 100 gray-scale images of 10 people, and then 900 images are augmented from the original images by three in-plane rotations and three scalings. The details of these three datasets are shown in Table 4.

In the experiments, the labeled data are one hundred samples randomly selected from datasets. The methods are repeated 10 times and the average error rates are used as the final results. Moreover, the other settings of the experiment are the same as in Section 4.2. The experimental results are shown in Table 5.

By comparing the results in Table 5, it can be known that ALGSSL has the lowest error rate in these three datasets. Therefore, the ALGSSL method can also perform well on large-scale datasets.

#### 4.4. Experimental on corrupted datasets

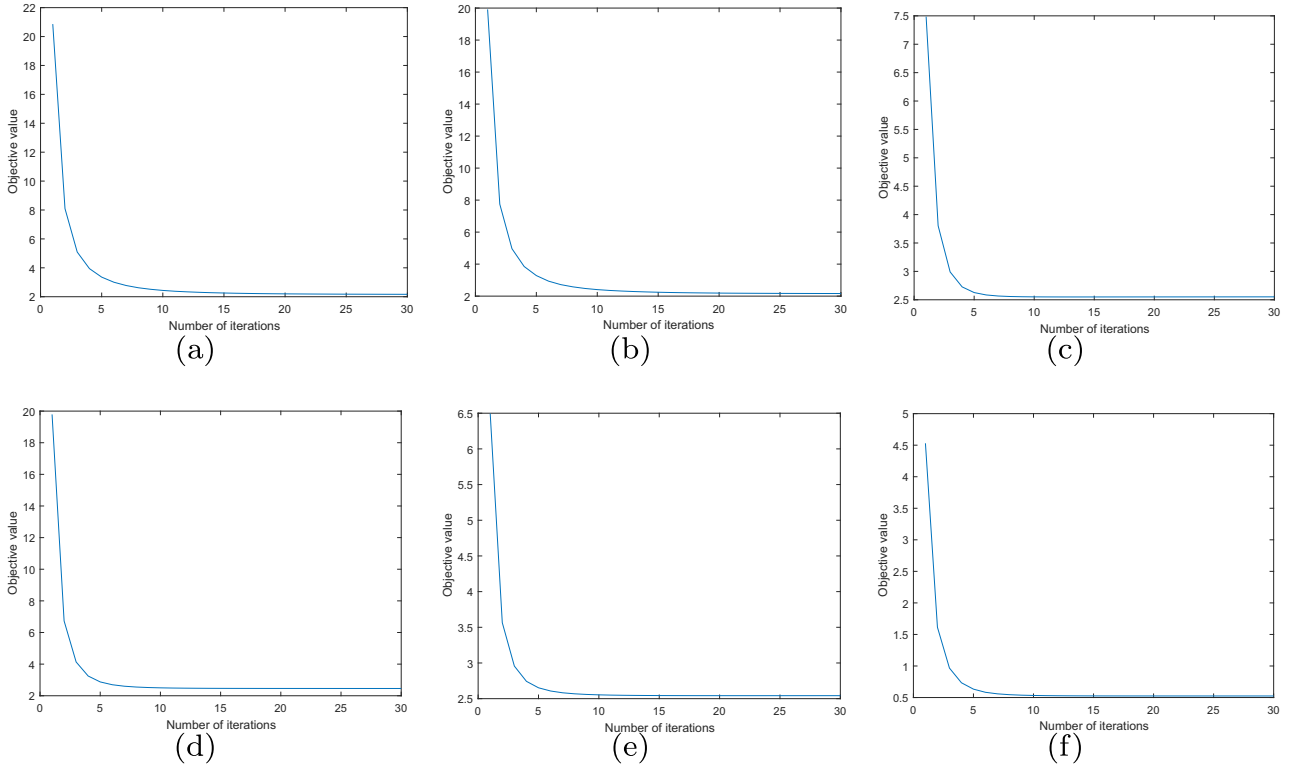
The effectiveness of adaptive graph is verified by taking corrupted datasets as an example. Datasets g241c, g241d, Digit1, USPS



**Table 3**

The average errors (%) with 100 samples. The values in the table represent the average error rate of the algorithm running 12 times under the datasets.

Method	g241c	g241d	Digit1	USPS	COIL	BCI
1-NN	40.28	37.49	6.12	7.64	23.27	44.83
SVM	23.11	24.64	5.53	9.75	22.93	34.31
TSVM	18.64	<b>22.42</b>	6.15	9.77	21.07	35.53
meanS3VM-iter	25.64	22.48	9.41	13.40	20.40	35.12
LDS	<b>18.04</b>	23.74	3.46	4.96	13.72	43.97
LGC	28.18	28.70	2.38	7.08	14.97	44.81
AEW	45.83	45.14	2.43	11.34	10.69	42.22
LEAD	39.66	37.35	4.74	7.73	12.25	45.26
SaGSSL	37.39	35.75	2.68	6.75	11.75	44.75
GGSSL1	45.49	42.74	2.39	6.38	9.48	45.44
GGSSL2	44.19	40.95	2.29	5.29	8.79	45.61
KNN-ALGSSL	44.32	41.66	6.78	10.16	12.35	35.20
ALGSSL	40.17	37.10	<b>1.71</b>	<b>4.34</b>	<b>8.66</b>	<b>34.29</b>



**Fig. 5.** The convergence curves of ALGSSL method on six datasets. (a) g241c (b) g241d (c) Digit1 (d) USPS (e) COIL (f) BCI.

and BCI which has two categories are used to construct corrupted datasets. For each dataset, we randomly select 3 samples from the labeled data of category 1 and change their labels, and do the same for the labeled data of category 2. That is, six of the labeled samples in each dataset are incorrectly labeled. The other settings of the experiment are the same as in Section 4.2. The experimental results are shown in Table 6.

Compared with the results in Table 3, the error rates of the AGSSL method in the five corrupted datasets change by an average of 0.19%. This shows that the results of the ALGSSL method on the corrupted datasets have not significantly deteriorated. Simultaneously, ALGSSL method has the lowest error rate among corrupted datasets Digit1, USPS, and BCI.

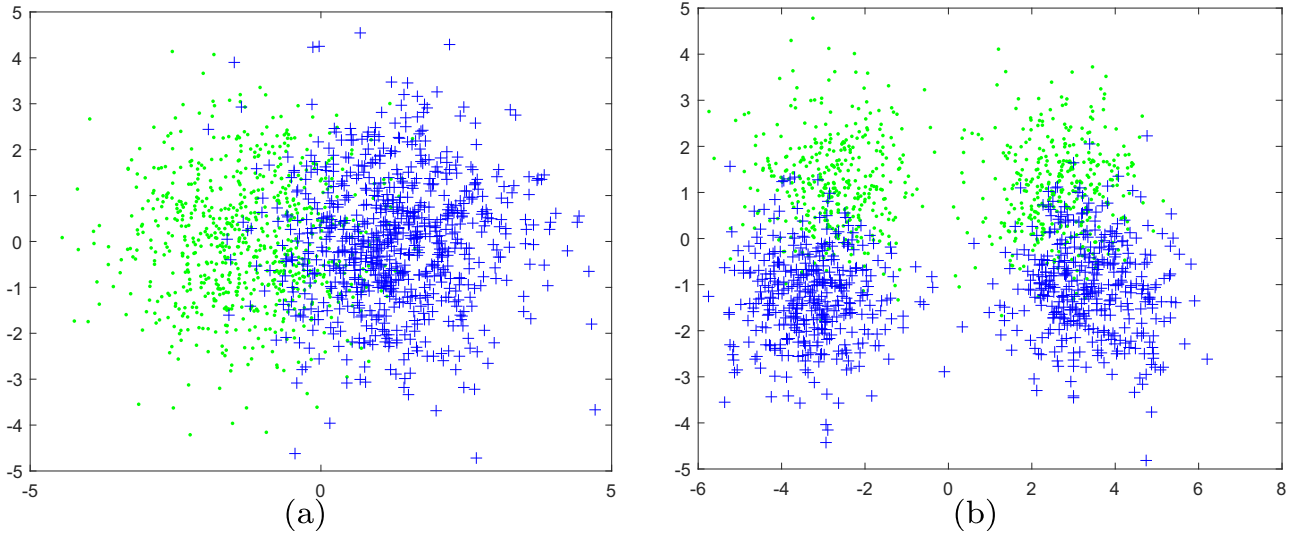
#### 4.5. Discovering the new category

We prepare an additional experiment to demonstrate the ability of the algorithm to discover new categories. In the above

comparison algorithms, only GGSSL can discover new categories, so we compare it with the GGSSL algorithm (including GGSSL1 and GGSSL2).

We need to remove the labels from several categories and treat them as unknown categories in unlabeled data. Then we want the algorithms to classify data from unknown categories into a new category. COIL has six categories and the rest of the datasets only have two categories in the above benchmark datasets, so we use COIL for this experiment. The label information for the first three categories of COIL is used, and the remaining three categories are used as unknown categories.

In the algorithm, the labeled data is one hundred randomly selected data in the first three categories of COIL, and the remaining data of COIL is unlabeled data. The experiments are repeated 12 times to calculate the average of the error rates. The relevant parameter settings about these algorithms are the same as in previous section. To be fair, we use the same graph construction algorithm as AGSSL in the experiments of GGSSL1 and GGSSL2. As



**Fig. 6.** (a) The two-dimensional projection of g241c. ‘.’ represents the points with category 1, ‘+’ represents the points with category 2. (b) The two-dimensional projection of g241d. ‘.’ represents the points with category 1, ‘+’ represents the points with category 2.

**Table 4**

The details of the datasets.

ID	Dataset	Classes	Dimension	Points
1	Mnist	10	784	3495
2	MSRA25	12	256	1799
3	Olivetti face	10	256	900

**Table 5**

The average error rate (%) of corrupted datasets. The values in the table represent the average error rate of the algorithm running 10 times under the datasets.

Method	Mnist	MSRA25	Olivetti face
1-NN	29.09	4.80	13.75
SVM	33.84	1.03	15.77
AEW	14.36	1.02	29.82
LEAD	28.47	4.66	12.37
GGSSL1	13.93	0.77	10.14
GGSSL2	13.91	0.75	10.43
KNN-ALGSSL	14.38	0.93	11.95
ALGSSL	<b>13.43</b>	<b>0.62</b>	<b>8.15</b>

shown in the Table 7, we record the error rates of the new category data and the overall data.

Experimental results show that the error rate of our algorithm is lower than GGSSL1 and GGSSL2. Referring to the relevant information in Table 3, it can be known that the average error rate of ALGSSL is 8.66% in COIL under normal experimental conditions. We can find that the average error rate of our algorithm increase by only 0.38% after artificially removing the labels of the three categories.

**Table 6**

The average error rate (%) of corrupted datasets. Six of the labeled samples in each dataset are incorrectly labeled.

Method	g241c	g241d	Digit1	USPS	BCI
1-NN	40.28	38.04	7.78	7.55	45.21
SVM	<b>23.11</b>	<b>24.81</b>	5.75	9.85	35.11
AEW	46.23	45.58	2.74	11.64	41.95
LEAD	35.37	34.14	4.58	7.45	36.51
GGSSL1	45.49	42.72	2.45	6.41	45.59
GGSSL2	44.19	41.02	2.37	5.37	45.76
ALGSSL	40.33	36.48	<b>1.73</b>	<b>4.36</b>	<b>34.42</b>

**Table 7**

The average error rate (%) of COIL datasets after manual removal of three category label information. The data for the first three categories is represented by “the known categories”, and the data for the last three categories is represented by “the new category”.

Method	The known categories	The new category	All data
GGSSL1	18.73	1.27	10.00
GGSSL2	18.60	1.36	9.98
ALGSSL	<b>17.58</b>	<b>0.50</b>	<b>9.04</b>

There may be multiple unlabeled categories of data in the dataset, but they are classified into a new category in the proposed algorithm. It is feasible and effective for the following two points. First, the new category is not a real category. A sample of the new category in this paper is a sample that is different from the samples of the known categories. In the proposed algorithm, the samples are classified according to their similarities among the labeled samples. When the similarities between the samples and all known category samples are small, the samples will be classified into the new category. Second, in practice, if the experimental results show that there are a large number of samples in the new category, it means that the annotation of the dataset we used is problematic. According to the experimental results, the annotation of datasets can be dynamically and accurately modified.

## 5. Conclusion

In this paper, we propose a graph-based semi-supervised learning algorithm via adaptive Laplacian graph. The algorithm updates the graph as part of semi-supervised learning rather than using the

fixed initial graph. It is insensitive to the initial graph and reduce the inaccuracy caused by the outliers and the error features of the data in the direct graph construction. By adjusting the regularization parameters, new categories can be found and error labels can be corrected. Besides, we use a sparse construction graph approach. The initial map construction is more suitable for semi-supervised learning tasks. In the experiments of the benchmark datasets, our algorithm is better than the comparison algorithms except for two datasets which hold the clustering assumption but do not hold the manifold assumption. In the experiments with the dataset of unknown categories, the proposed method finds almost all the data from unknown categories and its average error rates for overall data increase by only 2.66%. According to those experiments, our proposed ALGSSL is demonstrated to be effective.

### CRediT authorship contribution statement

**Yuan Yuan:** Methodology, Writing - original draft, Writing - review & editing, Project administration. **Xin Li:** Software, Validation, Writing - original draft, Writing - review & editing. **Qi Wang:** Supervision, Writing - original draft, Writing - review & editing. **Feiping Nie:** Conceptualization.

### Declaration of Competing Interest

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

### Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 61632018, 61825603, U1864204 and 61773316.

### References

- [1] X. Zhu, Semi-supervised learning literature survey, University of Wisconsin-Madison Department of Computer Sciences, 2007.
- [2] M. Wang, W. Fu, S. Hao, D. Tao, X. Wu, Scalable semi-supervised learning by efficient anchor graph regularization, *IEEE Trans. Knowl. Data Eng.* 28 (7) (2016) 1864–1877.
- [3] B. Jiang, H. Chen, Y. Bo, et al., Scalable graph-based semi-supervised learning through sparse bayesian model, *IEEE Trans. Knowl. Data Eng.* 29 (12) (2017) 2758–2771.
- [4] M. Zhao, Z. Zhang, C. Zhan, W. Wang, Graph based semi-supervised classification via capped  $l_2$ ,  $l_1$ -norm regularized dictionary learning, in: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), IEEE, 2017, pp. 1019–1024.
- [5] J.B. Tenenbaum, V.D. Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science* 290 (5500) (2000) 2319–2323.
- [6] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (5500) (2000) 2323–2326.
- [7] W. Zhan, M.L. Zhang, Inductive semi-supervised multi-label learning with co-training, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 1305–1314.
- [8] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: AAAI, 2017, pp. 2408–2414.
- [9] X. Li, M. Chen, F. Nie, Q. Wang, Locality Adaptive Discriminant Analysis Locality Adaptive Discriminant Analysis, Locality Adaptive Discriminant Analysis, Proceedings of International Joint Conference on Artificial Intelligence (IJCAI) (2017) 2201–2207.
- [10] Q. Wang, M. Chen, F. Nie, X. Li, Detecting coherent groups in crowd scenes by multiview clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* 42 (1) (2018) 46–58.
- [11] F. Nie, G. Cai, J. Li, X. Li, Auto-weighted multi-view learning for image clustering and semi-supervised classification, *IEEE Transactions on Image Processing* 27 (3) (2017) 1501–1511.
- [12] L. Zhang, Q. Zhang, B. Du, J. You, D. Tao, Adaptive manifold regularized matrix factorization for data clustering, in: Twenty-sixth International Joint Conference on Artificial Intelligence, 2017, pp. 3399–3405.
- [13] M. Chen, Q. Wang, X. Li, Adaptive projected matrix factorization method for data clustering, *Neurocomputing* 306 (2018) 182–188.
- [14] J. Y. Wang, I. Almasri, X. Gao, Adaptive graph regularized nonnegative matrix factorization via feature selection, in: International Conference on Pattern Recognition, 2012, pp. 963–966.
- [15] A.K. Ziemann, D.W. Messinger, P.S. Wenger, An adaptive k-nearest neighbor graph building technique with applications to hyperspectral imagery, in: Image and Signal Processing Workshop, 2015, pp. 32–36.
- [16] L.P. Valem, D.C. Pedronette, F. Breve, I.R. Guilherme, Manifold correlation graph for semi-supervised learning, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–7.
- [17] Y. Kuznetsov, J. Stückler, B. Leibe, Semi-supervised deep learning for monocular depth map prediction, in: IEEE Conference on Computer Vision & Pattern Recognition, 2017.
- [18] R. Fierimonte, S. Scardapane, A. Uncini, M. Panella, Fully decentralized semi-supervised learning via privacy-preserving matrix completion, *IEEE Trans. Neural Networks Learn. Syst.* 28 (11) (2017) 2699–2711.
- [19] F. Nie, S. Xiang, Y. Liu, C. Zhang, A general graph-based semi-supervised learning with novel class discovery, *Neural Comput. Appl.* 19 (4) (2010) 549–555.
- [20] D.J. Miller, A mixture of experts classifier with learning based on both labelled and unlabeled data, *Adv. Neural Inf. Process. Syst.* 9 (5) (1997) 571–577.
- [21] K. Nigam, A.K. McCallum, S. Thrun, T. Mitchell, Text classification from labeled and unlabeled documents using EM, *Mach. Learn.* 39 (2–3) (2000) 103–134.
- [22] A. Fujino, N. Ueda, K. Saito, A hybrid generative/discriminative approach to semi-supervised classifier design, in: National Conference on Artificial Intelligence, 2005, pp. 764–769.
- [23] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Conference on Computational Learning Theory, 1998, pp. 92–100.
- [24] S.A. Goldman, Y. Zhou, Enhancing supervised learning with unlabeled data (2000) 327–334.
- [25] Z.H. Zhou, M. Li, Tri-training: exploiting unlabeled data using three classifiers, *IEEE Trans. Knowl. Data Eng.* 17 (11) (2005) 1529–1541.
- [26] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, Learning with local and global consistency, in: International Conference on Neural Information Processing Systems, 2003, pp. 321–328.
- [27] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, *J. Mach. Learn. Res.* 7 (Nov.) (2006) 2399–2434.
- [28] M. Karasuyama, H. Mamitsuka, Adaptive edge weighting for graph-based learning algorithms, *Mach. Learn.* 106 (2) (2017) 307–335.
- [29] D.P. Kingma, S. Mohamed, D.J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.
- [30] C.-Y. Chang, T.-Y. Chen, P.-C. Chung, Semi-supervised learning using generative adversarial networks, in: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2018, pp. 892–896.
- [31] K. Atarashi, S. Oyama, M. Kurihara, Semi-supervised learning from crowds using deep generative models, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
- [32] Y. Meng, R. Shang, F. Shang, L. Jiao, S. Yang, R. Stolkin, Semi-supervised graph regularized deep nmf with bi-orthogonal constraints for data representation, *IEEE Trans. Neural Networks Learn. Syst.* (2019).
- [33] J. Huang, F. Nie, H. Huang, A new simplex sparse learning model to measure data similarity for clustering, *IJCAI* (2015) 3569–3575.
- [34] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained laplacian rank algorithm for graph-based clustering, in: AAAI, 2016, pp. 1969–1976.
- [35] T. Joachims, *Learning to Classify Text Using Support Vector Machines*, Kluwer Academic Publishers, 2002.
- [36] Y.-F. Li, J. T. Kwok, Z.-H. Zhou, Semi-supervised learning using label mean, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 633–640.
- [37] O. Chapelle, Semi-supervised classification by low density separation, in: Tenth International Workshop on Artificial Intelligence and Statistics, 2004.
- [38] Y.-F. Li, S.-B. Wang, Z.-H. Zhou, Graph quality judgement: a large margin expedition, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 1725–1731.
- [39] H. Gan, Z. Li, W. Wu, Z. Luo, R. Huang, Safety-aware graph-based semi-supervised learning, *Expert Syst. Appl.* 107 (2018) 243–254.
- [40] O. Chapelle, B. Schölkopf, A. Zien, *Semi-supervised learning*, Springer Berlin Heidelberg, 2006.
- [41] X. Li, M. Chen, W. Qi, A multiview-based parameter free framework for group detection, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence* (2017) 4147–4153.



**Yuan Yuan** (M'05-SM'09) is currently a full professor with the School of Computer Science and Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P.R. China. She has authored or coauthored over 150 papers, including about 100 in reputable journals such as IEEE Transactions and Pattern Recognition, as well as conference papers in CVPR, BMVC, ICIP, and ICASSP. Her current research interests include visual information processing and image/video content analysis.



**Qi Wang** (M'15-SM'15) received the B.E. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2005 and 2010, respectively. He is currently a Professor with the School of Computer Science, and with the Center for OPTical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xi'an, China. His research interests include computer vision and pattern recognition.



**Xin Li** received the B.E. degree in Software Engineering from the Northwestern Polytechnical University, Xi'an 710072, Shaanxi, P.R. China, in 2017. She is currently pursuing the doctoral degree from the Center for Optical IMagery Analysis and Learning (OPTIMAL), Northwestern Polytechnical University, Xian, China. Her research interests include hyperspectral anomaly detection and semi-supervised clustering.



**Feiping Nie** received the PhD degree in computer science from Tsinghua University, China, in 2009. He is currently a full professor at Northwestern Polytechnical University, Xi'an, China. His research interests are machine learning and its application fields, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He has published more than 100 papers in the prestigious journals and conferences like TIP, TPAMI, TNNLS, TKDE, ICML, NIPS, KDD, etc. He is currently serving as an associate editor or a PC member for several prestigious journals and conferences in the related fields.