



Graph-based semi-supervised learning via improving the quality of the graph dynamically

Jiye Liang¹ · Junbiao Cui¹ · Jie Wang¹ · Wei Wei¹

Received: 15 January 2020 / Revised: 26 January 2021 / Accepted: 5 April 2021 /
Published online: 13 May 2021

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

Abstract

Graph-based semi-supervised learning (GSSL) is an important paradigm among semi-supervised learning approaches and includes the two processes of graph construction and label inference. In most traditional GSSL methods, the two processes are completed independently. Once the graph is constructed, the result of label inference cannot be changed. Therefore, the quality of the graph directly determines the GSSL's performance. Most traditional graph construction methods make certain assumptions about the data distribution, resulting in the quality of the graph heavily depends on the correctness of these assumptions. Therefore, it is difficult to handle complex and various data distribution for traditional graph construction methods. To overcome such issues, this paper proposes a framework named Graph-based Semi-supervised Learning via Improving the Quality of the Graph Dynamically. In it, the graph construction based on the weighted fusion of multiple clustering results and the label inference are integrated into a unified framework to achieve their mutual guidance and dynamic improvement. Moreover, the proposed framework is a general framework, and most existing GSSL methods can be embedded into it so as to improve their performance. Finally, the working mechanism, the effectiveness in improving the performance of GSSL methods and the advantage compared with other GSSL methods based on dynamic graph construction methods of the proposal are verified through systematic experiments.

Keywords Semi-supervised learning · Graph construction · Clustering · Label inference

Editor: Zhi-Hua Zhou.

✉ Jiye Liang
ljy@sxu.edu.cn

Junbiao Cui
945546899@qq.com

Jie Wang
812849431@qq.com

Wei Wei
weiwei@sxu.edu.cn

¹ Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

1 Introduction

As the branch that best reflects the intelligence in the field of artificial intelligence, machine learning has attracted considerable attention in the past few decades (Michalski and Anderson 1984). Machine learning has achieved a huge success in a variety of tasks, especially in supervised learning tasks such as classification and regression. Most successful supervised learning, such as supervised deep learning (Lecun et al. 2015), requires sufficient labeled samples.

However. In many practical tasks, it is hard to obtain sufficient labeled samples because the labeling process is too costly, while a large number of unlabeled samples can be easily obtained. Although these unlabeled samples are unable to provide clear supervision information, they contain important information about the data distribution. Therefore, the unlabeled samples are helpful to improve the performance of the learner. It is the motivation and the ultimate goal of semi-supervised learning to enhance the generalization ability of learners by using a large number of inexpensive unlabeled samples. For this reason, semi-supervised learning has received much attention in the past few decades (Chapelle et al. 2006; Zhu and Goldberg 2009; Triguero et al. 2015; Van Engelen and Hoos 2020). Various types of semi-supervised learning methods have been proposed, forming four important semi-supervised learning paradigms: the generative semi-supervised learning method (Shahshahani and Landgrebe 1994; Cozman and Cohen 2002), the co-training style semi-supervised learning method (or the disagreement-based semi-supervised learning method) (Blum and Mitchell 1998; Wang and Zhou 2010), the semi-supervised SVM method (Joachims 1999; Chapelle et al. 2008) and the graph-based semi-supervised learning method (Zhu et al. 2003; Zhou et al. 2003). Meanwhile, semi-supervised learning has also been extensively studied in other fields such as regression (Zhou and Li 2005), clustering (Wagstaff et al. 2001; Basu et al. 2002; Zeng and Cheung 2012), dimensionality reduction (Zhang et al. 2007) and feature selection (Sheikhpour et al. 2017; Sechidis and Brown 2018), etc.

In recent years, deep learning (Lecun et al. 2015) has also made great progress in the semi-supervised learning field, just as it does in the supervised learning field. On the one hand, the neural network method has been applied to the three semi-supervised learning paradigms: the generative method (Kingma et al. 2014; Dai et al. 2017; Li et al. 2017), the co-training style method (Chen et al. 2018) and the graph-based method (Kipf and Welling 2017; Li et al. 2018; Jiang et al. 2019). On the other hand, the methodology of semi-supervised learning, using unlabeled samples to enhance the generalization performance of the learner, has also been applied to deep learning to train the deep neural networks (Li et al. 2018; Weston et al. 2008; Lee 2013) or design new neural networks (Rasmus et al. 2015; Park et al. 2018; Berthelot et al. 2019). For more content, we recommend readers with the recent survey article (Van Engelen and Hoos 2020). These methods not only enrich the semi-supervised learning field but improve the performance of semi-supervised learning in the related tasks.

The graph-based semi-supervised learning (GSSL) is an important semi-supervised learning paradigm, and its core assumption is that similar samples on the graph should possess the same label. Due to the good flexibility (various relationships between samples can be captured by constructing a specific graph), the high interpretability and good generalization performance, many methods in this framework have been proposed and have some success (Zhu et al. 2003; Zhou et al. 2003; Belkin et al. 2006; Subramanya and Bilmes 2011). Moreover, it is still an active research area in semi-supervised

learning (Berton et al. 2017; Rustamov and Klosowski 2018). The study of GSSL methods can be divided into two aspects:

(1) Label inference

The label inference in GSSL mainly focuses on how to carry out label learning based on the supervised information provided by labeled samples and the similarities between samples provided by the graph. There are many successful methods, such as the traditional semi-supervised learning method using Gaussian fields and harmonic functions (Zhu et al. 2003), the graph label propagation method (Zhou et al. 2003), the LapSVM and LapRLS method based on manifold regularization (Belkin et al. 2006) and the class probability distribution measure propagation on the graph (Subramanya and Bilmes 2011).

(2) Graph construction

For the GSSL method, it is critical and very difficult to construct high-quality graphs (De Sousa et al. 2013).

Recent research indicates that the key to the success of the GSSL is to construct high-quality graphs rather than design better label inference algorithms (Jebara et al. 2009; Berton and de Andrade Lopes 2014; Li et al. 2016). The GSSL's core assumption is that similar samples on the graph should share the same class label. According to this criterion, if the similarities between samples on the graph are consistent with their true class labels, the class labels of the unlabeled samples can be correctly predicted through the smoothness constraint on the graph. Conversely, if the similarity on the graph are contrary to the true class labels of the samples, the unlabeled samples will be given the wrong class labels by the label inference. Related experimental results of such cases are found in the literature (Belkin and Niyogi 2008; Karlen et al. 2008). In these cases, the utilization of unlabeled samples will lead to a negative effect: deteriorating the performance of the learner, an occurrence known as unsafe phenomena in semi-supervised learning (Li et al. 2016; Li and Zhou 2015; Wei et al. 2018). Therefore, the quality of the graph is extremely important for the performance of the GSSL method.

However, it is extremely difficult to construct a high-quality graph in GSSL because there are no operational metric for evaluating the quality of the graph. The quality of the graph can only be evaluated indirectly by the classification accuracy of the result of label inference on the graph, which is a post-mortem verification method and cannot provide any guidance to the graph construction. This is why the construction of a high-quality graph is difficult in GSSL. Fortunately, this difficulty provides us with enlightenment: why not let the graph construction and label inference guide each other to achieve their common improvement? Motivated by this insight. In this paper we integrate graph construction and label inference into an optimization model.

Before describing the details of the proposed method, we need to briefly review the existing graph construction methods in GSSL. The basic task of the graph construction is measuring the similarities between samples. According to how the similarities between samples are computed, the existing graph construction methods in GSSL can be roughly divided into two categories:

(2.1) Distance metric-based methods

The distance metric-based graph construction methods measure the similarities between samples by computing a certain distance between them, and intuitively, a pair of sample with a smaller distance should have a higher similarity. Among such methods, the most commonly used graph construction methods include the kNN graph and the ϵ -ball neighborhood graph (Zhu et al. 2003) based on the Euclidean distance. Meanwhile, the Gaussian kernel weighting method is also popular for graph construction in GSSL. In some cases on the kNN graph, the degree of the node varies greatly, which will deteriorate the quality of the graph. To solve this problem, the b -matching graph, in which the degree of each node is constrained to be b , was used in GSSL (Jebara et al. 2009).

Instead of using the Euclidean distance, the graph construction method based on the manifold hypothesis measure the similarities between samples through the geodesic distance. The key to this kind of method is how to compute the geodesic distances between samples accurately. The classic method uses the length of the shortest path on the Euclidean distance-based kNN graph to approximate the geodesic distance (Tenenbaum et al. 2000). Nevertheless, this approximation method has the problems named “short circuit” and “open circuit” over manifolds due to the inherent defects of the kNN graph. For relieving the “short circuit” problem over manifolds, a method for detecting and correcting the weight of the “short circuit” edge was proposed in Ghazvininejad et al. (2011) and was used to better compute the geodesic distances between samples.

In addition, some studies note that the valuable supervision information provided by the labeled samples should also be used for the graph construction in GSSL. In Berton and de Andrade Lopes (2014), the graph construction based on informativeness of labeled instance (GBILI) method was proposed. In which the distances between samples and the sum of distances between the sample and the all labeled samples are considered jointly to guide the edge generation on the graph. The GBILI method makes the labeled nodes tend to connect more edges so that the label information can spread to the unlabeled samples effectively. To further improve the robustness of the GBILI method, the literature (Berton et al. 2017) proposed a robust graph construction method considering the label information and proved that the graph constructed by this method is the optimal graph for modeling the smoothness hypothesis under certain conditions.

The basic principle of the distance metric-based graph construction methods is that a pair of sample with a smaller distance should have a higher similarity. In which the distance metric needs to be chosen in advance. If the distance metric is chosen inappropriately, the corresponding graph will not correctly reflect the similarities between samples, resulting in performance deterioration of the subsequent GSSL. At the same time, the quality of the graph is also affected significantly by the choice of parameters (such as the number of neighbors and the distance threshold), which also affects the performance of the GSSL. Furthermore, once the distance metric and parameters are selected, the corresponding graph will be fixed, thus it is unable to deal with various data distribution adaptively and poses difficulties in guaranteeing the performance of the GSSL.

(2.2) Data representation-based methods

The data representation-based graph construction methods measure the similarities between samples by the representation coefficients between samples that are obtained by solving a certain data representation model. The literature (Wang and Zhang 2008)

reconstructed a sample using a convex combination of the sample's k nearest neighbors, and designed the linear neighborhood propagation (LNP) algorithm to propagate the label on the graph. Inspired by the strong discriminating power of sparse representation (Wright et al. 2009), the ℓ_1 graph (Yan and Wang 2009) was constructed by using the absolute value of linear representation coefficients learned by sparse representation to measure the similarities between samples. After that, the literature (Cheng et al. 2010) added a nonnegative constraint on the sparse representation coefficients to better measure the similarity. In this literature, the representation coefficient matrix can be regarded as a graph for spectral clustering, subspace learning and GSSL. Since these above mentioned three methods all optimize each sample's representation coefficients individually, the representation coefficient matrix cannot capture the global information of the data distribution.

Inspired by the low-rank representation of data (Liu et al. 2013), the literature (Zhuang et al. 2011) proposed a method that implements a sparse and low-rank representation learning simultaneously. In this method, the ℓ_1 norm and nuclear norm regularization term are both applied to the representation coefficient matrix of all samples to capture the local and global structure simultaneously. When the data representation coefficients are obtained, the absolute values of the representation coefficients are used to measure the similarities between samples. Similar to the distance metric-based graph construction methods, the supervision information is also applied in the data representation-based graph construction method. In literature (Zhuang et al. 2017), semi-supervised low-rank representation (SSLRR) was proposed to construct a graph for GSSL. In which the representation coefficients between two labeled samples with different class labels are constrained to 0.

The data representation-based methods can learn the adjacent structure and edge weight of the graph and have the robustness to the noise data. However, this kind of method will be unable to reveal the data distribution correctly when the data distribution does not satisfy the subspace hypothesis, resulting in the inability to guarantee the performance of the subsequent GSSL.

It can be seen from the above analysis that the distance metric-based and the data representation-based graph construction methods heavily depend on their corresponding assumptions. If the assumption is incorrect, the two kinds of methods mentioned above will be unable to correctly capture the similarity that is consistent with the data distribution, which could result in the deterioration of the performance of the subsequent GSSL. However, the data distribution is complex and varies from data to data in practice, so it is hard to measure the true similarities between samples adaptively by using a specific assumption. Thus, to build a high-quality graph, it is necessary to propose a graph construction method that can alleviate the issues caused by the complex and various data distribution and can discover the potential data distribution adaptively.

Considering the above requirements, in this paper, we turn to the domain of the clustering ensemble to find a solution. In the field of clustering, the clustering ensemble is a popular way to improve the quality and robustness of the final clustering results. By integrating multiple clustering results into a final clustering result, the clustering ensemble can obtain a stronger (Bai et al. 2018) or a more robust clustering result (Zhao et al. 2017). Among a large number of clustering ensemble methods, the similarity-based method, fusing many base clusterings to construct a sample similarity matrix (Fred and Jain 2005), is flexible and effective. The reason why this method is effective is that different types of base clusterings can capture different types of data distribution. Furthermore, by fusing multiple different clustering results, a robust similarity measure can be obtained for the subsequent clustering, which improves the quality and robustness of the final clustering result.

Inspired by this idea, we propose a graph construction method that measures the similarities between samples by weighted fusion of multiple clustering results. First, different

clustering algorithms with different settings are run to obtain multiple clustering results, which can capture complex and various data distribution. Then, the weighted fusion of these clustering results is used for graph construction and the weights of the multiple clustering results are adjusted dynamically according to the “must link”y and “cannot link” constraints provided by the labeled samples and the result of label inference on the graph. During the process of learning, the dynamic graph construction and label inference on the graph are optimized alternately, which achieves the dynamic improvement of the quality. In summary, the contributions of the proposal in this paper include the following three points:

1. The weighted fusion of multiple clustering results is used to construct the graph in GSSL. By fusing multiple different clustering results, this method can alleviate the issues caused by complex and various data distribution effectively.
2. The graph construction and label inference on the graph are integrated into a unified optimization model, which realizes the mutual guidance between these two processes. In the optimization model, the supervision information and the iterative intermediate results are rationally utilized, which dynamically improves the quality of the graph during the learning process.
3. The proposed method is a general framework and many existing GSSL methods can be embedded into this framework to improve their performance.

The rest of this paper is organized as follows. Section 2 introduces some basic notions and the related works. Including the general framework of GSSL and some representative graph construction methods. Section 3 provides the description of the GSSL-IQGD framework proposed in this paper. In sect. 4, we explain why the proposal is effective by using three toy examples on artificial data sets and verify it’s effectiveness by comparing it with other classic GSSL methods on ten benchmark data sets. The conclusion and further work prospects are given in Sect. 5.

2 Notations and related works

2.1 Formalization of the problem

For a given semi-supervised classification task, let $D_l = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$ denote the l labeled samples and $D_u = \{\mathbf{x}_j\}_{j=l+1}^n$ denote the u unlabeled samples, where $n = l + u$ and $\mathbf{x}_i \in \mathbb{R}^d$ is the d dimension description for the i th sample, and $y_i \in \{1, 2, \dots, c\}$ is the class label of the i th labeled sample, and c is the number of categories.

For convenience of discussion, the class label of the sample is described in the form of a matrix. Let $\mathbf{F} \in \{0, 1\}^{n \times c}$ be the label matrix, where

$$f_{ik} = \begin{cases} 1, & \text{if } (1 \leq i \leq l) \wedge (y_i = k) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

and let $\mathbf{Z} \in \mathbb{R}^{n \times c}$ be the predicting label matrix, where z_{ik} represents the membership degree of the i th sample to the k th category. In the rest of this article, let \mathbf{f}_i and \mathbf{Z}_i be the i th row of the matrix \mathbf{F} and \mathbf{Z} , respectively. More notations are included in Table 1.

Table 1 Definition of main notations

Notations	Domain	Description
l, u and n	\mathbb{N}	# labeled, unlabeled and total samples
d and c	\mathbb{N}	# features and classes
\mathbf{F}	$\{0, 1\}^{n \times c}$	Known label matrix, see formula (1)
\mathbf{f}_i	$\{0, 1\}^{1 \times c}$	The i th row of matrix \mathbf{F}
f_{ik}	$\{0, 1\}$	Entry in the i th row and the k th column of \mathbf{F}
\mathbf{Z}	$\mathbb{R}^{n \times c}$	Predicting label matrix
\mathbf{Z}_i	$\mathbb{R}^{1 \times c}$	The i th row of matrix \mathbf{Z}
z_{ik}	\mathbb{R}	Entry in the i th row and the k th column of \mathbf{Z}
m	\mathbb{N}	# clustering results
$\mathbf{R}^{(t)}$	$\{0, 1\}^{n \times n}$	Matrix representation of the t th clustering result, see formula (18)
$\mathbf{r}_i^{(t)}$	$\{0, 1\}^{1 \times n}$	The i th row of matrix $\mathbf{R}^{(t)}$
$r_{ij}^{(t)}$	$\{0, 1\}$	Entry in the i th row and the j th column of $\mathbf{R}^{(t)}$
\mathbf{W}	$\mathbb{R}^{n \times n}$	Edge weight matrix of graph
\mathbf{w}_i	$\mathbb{R}^{1 \times n}$	The i th row of matrix \mathbf{W}
w_{ij}	\mathbb{R}	Entry in the i th row and the j th column of \mathbf{W}
$tr(\cdot)$	\mathbb{R}	Trace of the square matrix \cdot

2.2 Graph smoothness term and label inference of GSSL

For a semi-supervised classification task described in Sect. 2.1, the GSSL method first converts the data into a graph $G = (V, E, \mathbf{W})$, where $V = \{v_i\}_{i=1}^n$ is the vertices set, and the vertex v_i corresponds to the sample \mathbf{x}_i . Additionally, E is the edges set. The nonnegative matrix \mathbf{W} represents the weight of each edge in E and $w_{ij} = 0$ means there is no edge between vertex v_i and v_j . For an undirected graph, we have $\mathbf{W} = \mathbf{W}^T$.

The graph smoothness term is an important component of the GSSL method. In general, the smoothness loss term on the graph can be written as:

$$L_{\text{smooth}}(\mathbf{Z}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} d(\mathbf{Z}_i, \mathbf{Z}_j), \tag{2}$$

where $d(\cdot, \cdot)$ is a certain distance or dissimilarity metric, $d(\mathbf{Z}_i, \mathbf{Z}_j)$ measures the difference between the prediction results of sample \mathbf{x}_i and \mathbf{x}_j , and w_{ij} reflects the similarities between them. The effect of minimizing Eq. (2) can be explained as follows: the more similar the samples are on the graph, the closer their prediction labels should be. Based on the smoothness assumption, GSSL learns labels for unlabeled samples.

In general, the GSSL framework (Zhou et al. 2003) can be written as:

$$\min_{\mathbf{Z}} L(\mathbf{Z}) = \gamma_{\text{fit}} L_{\text{fit}}(\mathbf{Z}) + \gamma_{\text{smooth}} L_{\text{smooth}}(\mathbf{Z}), \tag{3}$$

where $L_{\text{fit}}(\mathbf{Z})$ is the fitting loss on the known class labels, and $L_{\text{smooth}}(\mathbf{Z})$ is the smoothness loss on the graph. γ_{fit} and γ_{smooth} are two hyper-parameters¹ trading off the fitting loss and smoothness loss, respectively.

Accordingly, the prediction function for the unlabeled sample \mathbf{x}_j is:

$$y_j = \arg \max_{k=1,2,\dots,c} z_{jk}. \tag{4}$$

It should be noted that. In essence, the model represented by (3) is the same as the regularization framework in Zhou et al. (2003). Most classical semi-supervised learning methods, such as the Harmonic (Zhu et al. 2003), LLGC (Zhou et al. 2003), LapRLS (Belkin et al. 2006), LapSVM (Belkin et al. 2006) and measure propagation (Subramanya and Bilmes 2011), can be described by this framework.

2.3 Semi-supervised classification with graph convolutional networks

Among all deep neural network based semi-supervised learning methods, the SSC-GCN (semi-supervised classification with graph convolution networks) proposed in 2017 (Kipf and Welling 2017) and its extensions (Li et al. 2018; Jiang et al. 2019) are closest to the GSSL. In addition to the input data described in Sect. (2.1), the graph $G = (V, E, \mathbf{W})$ is also given in advance in the SSC-GCN.

In the method, first, the symmetric and normalized graph Laplacian matrix $\hat{\mathbf{W}}$ is computed as:

$$\hat{\mathbf{W}} = \bar{\mathbf{D}}^{-1/2} \bar{\mathbf{W}} \bar{\mathbf{D}}^{-1/2}, \tag{5}$$

where $\bar{\mathbf{W}} = \mathbf{W} + \mathbf{I}$ and $\bar{\mathbf{D}} = \text{diag}(\bar{d}_1, \bar{d}_2, \dots, \bar{d}_n)$ is the degree matrix with $\bar{d}_i = \sum_{j=1}^n \bar{w}_{ij}, i = 1, 2, \dots, n$. Then, the spatial-based graph convolution is applied to the output of each layer of the neural network to obtain smooth hidden representations, i.e. the hidden representations of similar samples on the graph are close. At last, a two layers SSC-GCN model used in literature (Kipf and Welling 2017) is expressed in the following form:

$$\mathbf{Z} = \text{softmax} \left(\hat{\mathbf{W}} \text{ReLU} \left(\hat{\mathbf{W}} \mathbf{X} \mathbf{W}^{(0)} \right) \mathbf{W}^{(1)} \right), \tag{6}$$

where $\mathbf{X} = (\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T)^T \in \mathbb{R}^{n \times d}$ is the matrix arranged by the description vectors of n samples, $\mathbf{W}^{(0)} \in \mathbb{R}^{d \times h}$ and $\mathbf{W}^{(1)} \in \mathbb{R}^{h \times c}$ are parameters of the graph convolution network, h is the number of hidden neural units, $\text{ReLU}(\cdot) = \max(\cdot, 0)$ is the nonlinear activation function. And

$$\forall \mathbf{u} \in \mathbb{R}^c, \text{softmax}(\mathbf{u}) = \left(\frac{\exp(u_1)}{\sum_{k=1}^c \exp(u_k)}, \frac{\exp(u_2)}{\sum_{k=1}^c \exp(u_k)}, \dots, \frac{\exp(u_c)}{\sum_{k=1}^c \exp(u_k)} \right)$$

is applied to the final output of the network, so that the i th row of the predicting label matrix \mathbf{Z} corresponds to the membership degrees of the i th sample to all the categories. After getting the predicting label matrix \mathbf{Z} , the fitting loss

¹ For the convenience of discussion in this article, we added a hyper-parameter for every term of the objective function. Actually, one hyper-parameter is enough to balance the trade-off between two terms.

$$L_{\text{fit}}(\mathbf{Z}) = - \sum_{i=1}^l \sum_{k=1}^c f_{ik} \log z_{ik} \quad (7)$$

is evaluated over l labeled samples, then parameters $\mathbf{W}^{(0)}$ and $\mathbf{W}^{(1)}$ are trained by gradient descent.

In this paper we focus on how to construct a better graph to improve the performance of GSSL. In the SSC-GCN method, although the graph is given in advance, it can still be easily embedded in the proposed framework and its performance can be improved.

2.4 Graph construction method

The study of the GSSL method is mainly divided into two parts: graph construction and label inference. Recent research shows that the key to the success of the GSSL method is constructing a high-quality graph instead of designing a better label inference algorithm (Berton et al. 2017; De Sousa et al. 2013; Jebara et al. 2009; Zhuang et al. 2017). Therefore, this paper focuses on the graph construction method. The classical graph construction methods used in GSSL are briefly reviewed as follows.

2.4.1 Nearest neighbor graph

The “0–1” kNN graph and the weighted kNN graph are the most commonly used methods in GSSL (Zhu et al. 2003; Zhou et al. 2003). Formally, the entry of the edge weight matrix \mathbf{W} is defined as

$$w_{ij} = \begin{cases} 1, & \mathbf{x}_i \in kNN(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

or

$$w_{ij} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}}, & \mathbf{x}_i \in kNN(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases}. \quad (9)$$

Apart from these forms, there are some other kinds of the nearest neighbor graph that are also frequently used in GSSL, such as the ε -ball nearest neighbor graph, the mutual kNN graph and so on.

2.4.2 b – matching graph

To avoid the situation where the degree of some vertices in the kNN graph are very large while others is very small, the b -matching graph was proposed in the literature (Jebara et al. 2009), where the degree of each vertex is constrained to b . The corresponding optimization problem is as follows.

$$\begin{aligned}
 & \min_{\mathbf{W}} \sum_{i=1}^n \sum_{j=1}^n w_{ij} d(\mathbf{x}_i, \mathbf{x}_j) \\
 & \text{s.t. } w_{ij} \in \{0, 1\}, w_{ij} = w_{ji}, i, j = 1, 2, \dots, n \\
 & \sum_{j=1}^n w_{ij} = b, w_{ii} = 0, i = 1, 2, \dots, n
 \end{aligned} \tag{10}$$

where $d(\mathbf{x}_i, \mathbf{x}_j)$ is the distance between sample \mathbf{x}_i and \mathbf{x}_j . This optimization problem can be solved efficiently by the loopy belief propagation algorithm (Huang and Jebara 2007).

2.4.3 Linear neighbor graph

Unlike the method that directly uses the distances between samples to measure the similarity, the literature (Wang and Zhang 2008) proposed a linear representation-based similarity measure. In detail, for the sample \mathbf{x}_i , its k nearest neighbors $kNN(\mathbf{x}_i)$ are computed. And it is reconstructed by a convex combination of its $kNN(\mathbf{x}_i)$. The combination coefficients is used as the weights of the edges connected to node vertex v_i :

$$\begin{aligned}
 & \min_{\mathbf{W}} \sum_{i=1}^n \left\| \mathbf{x}_i - \sum_{\mathbf{x}_j \in kNN(\mathbf{x}_i)} w_{ij} \mathbf{x}_j \right\|_2^2 \\
 & \text{s.t. } w_{ij} \geq 0, i, j = 1, 2, \dots, n \\
 & \sum_{j=1}^n w_{ij} = 1, i = 1, 2, \dots, n \\
 & w_{ij} = 0, i = 1, 2, \dots, n, \mathbf{x}_j \notin kNN(\mathbf{x}_i).
 \end{aligned} \tag{11}$$

2.4.4 ℓ_1 graph

To mine the subspace structure of data. In the literature (Cheng et al. 2010), the ℓ_1 graph was used to learn the adjacency structure and the edge weights simultaneously. The similarities between samples are measured by the absolute value of the linear combination coefficients learned by the sparse representation. In detail, the i th sample is reconstructed by the remaining $n - 1$ samples:

$$\min_{\alpha} \|\alpha\|_1 \text{ s.t. } (\mathbf{X}_{\bar{i}}, \mathbf{I})\alpha = \mathbf{x}_i \tag{12}$$

where $\mathbf{X}_{\bar{i}} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times (n-1)}$ is a matrix of all samples except the i th sample. In addition, the identity matrix $\mathbf{I} \in \mathbb{R}^{d \times d}$ is used as the basis for reconstructing the noise on \mathbf{x}_i , which can improve the robustness of the model. Actually, the combination coefficient vector can be split into two segments:

$$\alpha = \begin{pmatrix} \alpha_{samp} \\ \alpha_{noise} \end{pmatrix} \in \mathbb{R}^{(n-1)+d}. \tag{13}$$

Accordingly, we have $\mathbf{x}_i = \mathbf{X}_{\bar{i}}\alpha_{samp} + \mathbf{I}\alpha_{noise} = \mathbf{X}_{\bar{i}}\alpha_{samp} + \alpha_{noise}$; that is, the segment $\alpha_{samp} \in \mathbb{R}^{n-1}$ contains the coefficients for reconstructing the i th sample with the rest of the

samples. Based on the above analysis, let α^* be the optimal solution of problem (12); then, the weights of the edges connected to vertex v_i are calculated by

$$w_{ij} = \begin{cases} |\alpha_j^*|, & j < i \\ 0, & j = i \\ |\alpha_{j-1}^*|, & j > i \end{cases} \quad (14)$$

2.4.5 LRR graph and SSLRR graph

The low-rank representation is a robust subspace structure recovery method proposed in the literature (Liu et al. 2013). Unlike the sparse representation model that learns each sample representation coefficients individually. In which the representation coefficients of all samples lack global constraint, the low-rank representation can better capture the global structure of the data by applying low-rank regularization to the representation coefficient matrix. Usually, the nuclear norm $\|\cdot\|_*$ is used to approximate the rank of a matrix, and the low-rank representation model can be written as follows.

$$\min_{\mathbf{R}, \mathbf{E}} \|\mathbf{R}\|_* + \lambda \|\mathbf{E}\|_{2,1} \quad s.t. \mathbf{X} = \mathbf{XR} + \mathbf{E}. \quad (15)$$

After obtaining the optimal representation matrix \mathbf{R}^* , the weight of the edge between vertex v_i and v_j can be calculated by the following formula (Zhuang et al. 2011)

$$w_{ij} = \frac{|r_{ij}^*| + |r_{ji}^*|}{2}, \quad (16)$$

where r_{ij}^* is the element in the i th row and the j th column of the matrix \mathbf{R}^* .

To better use the supervised information to improve the quality of the graph, a semi-supervised graph construction method based on the LRR graph (SSLRR) is proposed in literature (Zhuang et al. 2017). In the SSLRR method, the “cannot link” constraint is added into the low-rank representation model to enforce the representation coefficients between samples with different class labels to be 0:

$$\begin{aligned} & \min_{\mathbf{R}, \mathbf{E}} \|\mathbf{R}\|_* + \lambda \|\mathbf{E}\|_{2,1} \\ & s.t. \mathbf{X} = \mathbf{XR} + \mathbf{E} \\ & \sum_{j=1}^n r_{ji} = 1, \quad i = 1, 2, \dots, n \\ & r_{ij} = 0, \quad i, j = 1, 2, \dots, l, \quad y_i \neq y_j. \end{aligned} \quad (17)$$

Similar to the LRR graph, the weight of the edge between vertex v_i and v_j is calculated by the formula (16).

It can be seen from the above discussion that the quality of the graph depends heavily on the assumptions used in these methods, whether it is for graph construction methods based on the distance metric or data representation (some kind of distance metric for the former and subspace structure for the latter). If the data distribution does not meet the corresponding assumptions, the quality of the graph will be seriously degraded, resulting in the performance deterioration of the subsequent GSSL. Indeed, the data distribution tends

to be complex and various, so the graph construction method based on specific assumption encounters difficulty in adaptively capturing the similarities between samples that is consistent with the data distribution.

Motivated by the above analysis, a graph construction method by fusing multiple clustering results is proposed for the following reasons:

1. Clustering is a classic method to mine the structure of a data distribution, and different clustering algorithms are good at mining different data distribution structures. Thus we can use different clustering algorithms to capture various data distribution structures.
2. We can construct a high-quality graph by integrating multiple clustering results reasonably.

3 GSSL via improving the quality of the graph dynamically

From the above discussion, we can see that the quality of the graph directly affects the performance of the GSSL method. Traditional methods are based on certain specific assumption, so it is difficult to capture the complex and various data distribution. To address this problem, the method of fusing multiple clustering results is employed to elevate the quality of the graph, which can improve the performance of the GSSL method.

3.1 Measuring similarity via the weighted co-association matrix

In practice, the potential data distribution tends to be complex and varies from data to data. Clustering is a classical unsupervised learning method that aims to discover the data distribution structure. Many classical clustering algorithms (Jain 2010) have been proposed, and different algorithms are expert in dealing with different data distribution. Therefore, different clustering algorithms with different settings (for more implementation details, see the experimental section) can be used to obtain a candidate set that covers various data distribution.

Assume that the clustering process produces m clustering results $\Pi = \{\pi_t\}_{t=1}^m$, where π_t is the t th clustering result, and let $\mathbf{R}^{(t)} \in \{0, 1\}^{n \times n}$ be the matrix derived from π_t , where

$$r_{ij}^{(t)} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same cluster in } \pi_t \\ 0, & \text{otherwise} \end{cases}. \quad (18)$$

The co-association matrix (Fred and Jain 2005) is defined as:

$$\mathbf{M}^{(\text{co})} = \frac{1}{m} \sum_{t=1}^m \mathbf{R}^{(t)}. \quad (19)$$

The element $m_{ij}^{(\text{co})}$ of matrix $\mathbf{M}^{(\text{co})}$ can be used to measure the similarity between \mathbf{x}_i and \mathbf{x}_j . It is widely accepted that the importance of different clustering result should be different. Thus the weighted co-association matrix can measure the similarity better:

$$\mathbf{W} = \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)} \quad \text{s.t. } \lambda \in \text{conv}_m, \quad (20)$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)^T$ is the weights vector of m clustering results, and $\text{conv}_m = \{\alpha \mid \alpha \in \mathbb{R}^m, \alpha_t \geq 0, t = 1, 2, \dots, m, \sum_{t=1}^m \alpha_t = 1\}$ is the simplex. Obviously, $w_{ij} \in [0, 1]$, and the larger the value is, the greater the similarity between x_i and x_j is.

The difference between formula (19) and (20) is that the weight of each clustering result in formula (19) is equal, while in formula (20) the weights are obtained by optimization (which will be shown in the subsequent discussion). From the perspective of the clustering ensemble, the quality of each clustering result is different. Naturally, they should be given different weights in the fusion process. However, in unsupervised scenario, how to evaluate the quality of the clustering result is an open problem. Therefore, the equal weight strategy adopted by formula (19) is not a bad choice. In this paper, some criteria are used to evaluate the quality of the clustering results. First, the supervision information provided by the labeled samples can be used to evaluate the quality of each clustering result (see Sect. 3.2), making higher quality clustering results obtain greater weights. Second, the result of label inference can be used as the pseudo label to evaluate the quality of each clustering result, which can dynamically adjust the weights of clustering results (see Sect. 3.3).

3.2 Refining the weights by means of supervision information

Unlike the clustering ensemble, the supervision information provided by the labeled samples can be used to evaluate the clustering result in semi-supervised learning. We can directly obtain the “must link” and “cannot link” constraint (Wagstaff et al. 2001; Zeng and Cheung 2012) through the labeled samples. Let ML and CL be the sets of “must link” sample pairs and “cannot link” sample pairs, respectively. The definitions of ML and CL are

$$ML = \{(\mathbf{x}_i, \mathbf{x}_j) \mid i, j = 1, 2, \dots, l, y_i = y_j\}, \quad (21)$$

and

$$CL = \{(\mathbf{x}_i, \mathbf{x}_j) \mid i, j = 1, 2, \dots, l, y_i \neq y_j\}. \quad (22)$$

For the sake of the this discussion, their matrix representations are defined as

$$\mathbf{M} \in \{0, 1\}^{n \times n}, \quad m_{ij} = \begin{cases} 1, & (\mathbf{x}_i, \mathbf{x}_j) \in ML \\ 0, & \text{otherwise} \end{cases}, \quad (23)$$

and

$$\mathbf{C} \in \{0, 1\}^{n \times n}, \quad c_{ij} = \begin{cases} 1, & (\mathbf{x}_i, \mathbf{x}_j) \in CL \\ 0, & \text{otherwise} \end{cases}. \quad (24)$$

If the sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ meets the “must link” constraint, then on the graph G their corresponding weight w_{ij} should be large. We can use the following optimization problem to achieve this goal.

$$\min_{\lambda} - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in ML} w_{ij} \quad \text{s.t.} \quad \mathbf{W} = \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)}, \quad \lambda \in \text{conv}_m. \quad (25)$$

If the sample pair $(\mathbf{x}_i, \mathbf{x}_j)$ meets the “cannot link” constraint, then on the graph G , their corresponding node pair should ideally be disconnected. However, this goal corresponds to a discrete optimization problem that is very difficult to implement by numerical optimization

techniques. In this paper, we use the following optimization problem to approximate the necessary condition for achieving this goal.

$$\min_{\lambda} \sum_{(x_i, x_j) \in CL} \mathbf{w}_i \mathbf{w}_j^T \quad s.t. \quad \mathbf{W} = \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)}, \quad \lambda \in \mathbf{conv}_m. \tag{26}$$

From formula (20) we know that the weight w_{ij} on the graph G is nonnegative. Therefore, the inner product of the weight vectors $\mathbf{w}_i \mathbf{w}_j^T$ is also nonnegative. The inner product $\mathbf{w}_i \mathbf{w}_j^T = 0$ means that the intersection of the set of nodes that directly connect to the i th node and the set of nodes that directly connect to the j th node is empty, i.e., $\{v_k | w_{ik} \neq 0, k = 1, 2, \dots, n\} \cap \{v_k | w_{jk} \neq 0, k = 1, 2, \dots, n\} = \phi$, which is a necessary condition for that the node v_i and v_j are disconnected on the graph G .

Combining optimization problems (25) and (26), we obtain the following optimization problem.

$$\begin{aligned} \min_{\lambda} \quad & -\gamma_{ml} \sum_{(x_i, x_j) \in ML} w_{ij} + \gamma_{cl} \sum_{(x_i, x_j) \in CL} \mathbf{w}_i \mathbf{w}_j^T \\ s.t. \quad & \mathbf{W} = \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)}, \quad \lambda \in \mathbf{conv}_m. \end{aligned} \tag{27}$$

where the γ_{ml} and γ_{cl} are two nonnegative trade-off hyper-parameters.

3.3 Optimizing the quality of the graph and the class label iteratively

In this section, the most commonly used squared loss is chosen as an example to show how to integrate the graph construction by fusing multiple clustering results and the label inference into a unified framework to achieve their mutual guidance and dynamic improvement. Formally, the squared fitting loss and smoothness terms can be written as follows.

$$L_{\text{fit}}(\mathbf{Z}) = \sum_{i=1}^n \|\mathbf{f}_i - \mathbf{Z}_i\|_2^2, \tag{28}$$

$$L_{\text{smooth}}(\mathbf{Z}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2. \tag{29}$$

It should be noted that the method proposed in this paper is a general framework for improving the quality of the graph dynamically in GSSL. Other GSSL methods, such as the Harmonic method (Zhu et al. 2003) (in which the squared loss is used as the fitting and smoothness loss), LLGC (Zhou et al. 2003) (in which the squared loss is used as the fitting and smoothness loss), LapSVM (Belkin et al. 2006) (in which the hinge loss is used as the fitting loss and the squared loss is used as the smoothness loss) and measure propagation (Subramanya and Bilmes 2011) (in which the KL divergence is used as the fitting and smoothness loss), can be embedded in this framework without modification and their performance can be improved.

By integrating formula (27), (28) and (29), the final form of the proposed model is given below.

$$\begin{aligned}
\min_{\mathbf{Z}, \lambda} L(\mathbf{Z}, \lambda) &= \gamma_{\text{fit}} \sum_{i=1}^n \|\mathbf{f}_i - \mathbf{Z}_i\|_2^2 + \frac{\gamma_{\text{smooth}}}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 \\
&\quad - \gamma_{\text{ml}} \sum_{(x_i, x_j) \in ML} w_{ij} + \gamma_{\text{cl}} \sum_{(x_i, x_j) \in CL} \mathbf{w}_i \mathbf{w}_j^T \\
\text{s.t. } \mathbf{W} &= \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)}, \quad \lambda \in \text{conv}_m.
\end{aligned} \tag{30}$$

In the rest of this paper, the method corresponding to the above formula is named SLLI-IQGD (Squared Loss Label Inference via Improving the Quality of the Graph Dynamically).

In formula (30), label inference and graph construction are integrated into an optimization model. In this model, the graph is constructed by fusing multiple clustering results, and the fusion weights are learned iteratively under the joint guidance of three kinds of information: “must link”, “cannot link” and the pseudo label generated by label inference. As a result, the quality of the graph and the result of label inference are improved dynamically.

3.4 Model solution

The alternating optimization method is used to solve the optimization problem (30), which contains fixing λ and updating \mathbf{Z} , and fixing \mathbf{Z} and updating λ .

3.4.1 Fixing λ and updating \mathbf{Z}

When λ is fixed, the optimization problem (30) can be written as follows.

$$\begin{aligned}
\min_{\mathbf{Z}} L(\mathbf{Z}) &= \gamma_{\text{fit}} \sum_{i=1}^n \|\mathbf{Z}_i - \mathbf{f}_i\|_2^2 + \frac{\gamma_{\text{smooth}}}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 \\
&= \gamma_{\text{fit}} \text{tr}((\mathbf{Z} - \mathbf{F})(\mathbf{Z} - \mathbf{F})^T) + \gamma_{\text{smooth}} \text{tr}(\mathbf{Z}^T \mathbf{L} \mathbf{Z}),
\end{aligned} \tag{31}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix of graph G and $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n)$ is the diagonal matrix with $d_i = \sum_{j=1}^n w_{ij}$, $i = 1, 2, \dots, n$. The differential of $L(\mathbf{Z})$ w.r.t \mathbf{Z} is:

$$\frac{\partial L(\mathbf{Z})}{\partial \mathbf{Z}} = 2\gamma_{\text{fit}} \mathbf{Z} - 2\gamma_{\text{fit}} \mathbf{F} + 2\gamma_{\text{smooth}} \mathbf{L} \mathbf{Z}. \tag{32}$$

Let the differential be $\mathbf{0}$, and we can obtain the following formula for updating \mathbf{Z} .

$$\mathbf{Z}^* = \gamma_{\text{fit}} (\gamma_{\text{smooth}} \mathbf{L} + \gamma_{\text{fit}} \mathbf{I})^{-1} \mathbf{F}. \tag{33}$$

Since the matrix \mathbf{L} is a positive semidefinite matrix and both γ_{fit} and γ_{smooth} are greater than 0, the matrix $(\gamma_{\text{smooth}} \mathbf{L} + \gamma_{\text{fit}} \mathbf{I})$ is an invertible matrix.

The optimization problem (31) is actually a label propagation algorithm on the graph under the regularization framework proposed in Zhou et al. (2003). Different from the literature (Zhou et al. 2003), the weight matrix \mathbf{W} is the weighted fusion of multiple clustering results. It can be seen from the solving process of the above subproblem that the weighted co-association matrix obtained by the previous iteration guides the learning of the label of the unlabeled samples through the smoothness loss term.

3.4.2 Fixing \mathbf{Z} and updating λ

When \mathbf{Z} is fixed, the optimization problem (30) can be written as follows.

$$\begin{aligned} \min_{\lambda} L(\lambda) &= \frac{\gamma_{\text{smooth}}}{2} \sum_{i=1}^n \sum_{j=1}^n w_{ij} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 - \gamma_{\text{ml}} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in ML} w_{ij} \\ &\quad + \gamma_{\text{cl}} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in CL} \mathbf{w}_i \mathbf{w}_j^T \end{aligned} \tag{34}$$

$$s.t. \mathbf{W} = \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)}, \lambda \in \text{conv}_m.$$

By substituting the first constraint, the matrix representation of the ‘‘must link’’ constraint defined in formula (23) and the ‘‘cannot link’’ constraint defined in (24) into the above objective function, the optimization problem (34) can be converted equivalently into the following form.

$$\begin{aligned} \min_{\lambda} L(\lambda) &= \sum_{t=1}^m \lambda_t \frac{\gamma_{\text{smooth}}}{2} \left(\sum_{i=1}^n \sum_{i=1}^n r_{ij}^{(t)} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2 \right) \\ &\quad - \sum_{t=1}^m \lambda_t \gamma_{\text{ml}} \left(\sum_{i=1}^l \sum_{i=1}^l r_{ij}^{(t)} m_{ij} \right) \\ &\quad + \gamma_{\text{cl}} \sum_{i=1}^l \sum_{i=1}^l c_{ij} \left(\lambda^T \begin{pmatrix} \mathbf{r}_i^{(1)} \\ \mathbf{r}_i^{(2)} \\ \vdots \\ \mathbf{r}_i^{(m)} \end{pmatrix} \right) \left(\lambda^T \begin{pmatrix} \mathbf{r}_j^{(1)} \\ \mathbf{r}_j^{(2)} \\ \vdots \\ \mathbf{r}_j^{(m)} \end{pmatrix} \right)^T \end{aligned} \tag{35}$$

$$s.t. \lambda \in \text{conv}_m$$

Let

$$\begin{aligned} \mathbf{v}^{\text{smooth}} &= (v_1^{\text{smooth}}, v_2^{\text{smooth}}, \dots, v_m^{\text{smooth}})^T \in \mathbb{R}^m, \\ v_t^{\text{smooth}} &= \sum_{i=1}^n \sum_{j=1}^n r_{ij}^{(t)} \|\mathbf{Z}_i - \mathbf{Z}_j\|_2^2, \quad t = 1, 2, \dots, m, \end{aligned} \tag{36}$$

$$\begin{aligned} \mathbf{v}^{\text{ml}} &= (v_1^{\text{ml}}, v_2^{\text{ml}}, \dots, v_m^{\text{ml}})^T \in \mathbb{R}^m, \\ v_t^{\text{ml}} &= \sum_{i=1}^l \sum_{j=1}^l r_{ij}^{(t)} m_{ij}, \quad t = 1, 2, \dots, m, \end{aligned} \tag{37}$$

$$\mathbf{v} = \frac{\gamma_{\text{smooth}}}{2} \mathbf{v}^{\text{smooth}} - \gamma_{\text{ml}} \mathbf{v}^{\text{ml}} \in \mathbb{R}^m, \tag{38}$$

$$\mathbf{S} = \gamma_{\text{cl}} \sum_{i=1}^l \sum_{j=1}^l c_{ij} \mathbf{S}^{(ij)}, \tag{39}$$

$$\mathbf{S}^{(ij)} = \begin{pmatrix} \mathbf{r}_i^{(1)} \\ \mathbf{r}_i^{(2)} \\ \vdots \\ \mathbf{r}_i^{(m)} \end{pmatrix} \begin{pmatrix} \mathbf{r}_j^{(1)} \\ \mathbf{r}_j^{(2)} \\ \vdots \\ \mathbf{r}_j^{(m)} \end{pmatrix}^T \in \mathbb{R}^{m \times m}. \tag{40}$$

Obviously, \mathbf{S} is a symmetric matrix.

By using formula (36–39), the optimization problem (35) can be equivalently written as the following form.

$$\min_{\lambda} L(\lambda) = \lambda^T \mathbf{S} \lambda + \mathbf{v}^T \lambda \quad \text{s.t. } \lambda \in \mathbf{conv}_m \tag{41}$$

The optimization problem (41) is a standard quadratic programming problem whose convexity depends on whether the matrix is a positive semidefinite matrix. Because \mathbf{S} is a symmetric matrix, all of its eigenvalues are real. Let μ_{\min} be the smallest eigenvalue of \mathbf{S} .

If $\mu_{\min} \geq 0$, then \mathbf{S} is a positive semidefinite matrix, and the optimization problem (41) turns out to be a convex quadratic programming (CQP) problem, which can be solved by using a convex quadratic programming algorithm (Boyd and Vandenberghe 2004).

If $\mu_{\min} < 0$, then \mathbf{S} is not a positive semidefinite matrix, and the optimization problem (41) is not a convex quadratic programming problem. In this case, the optimization problem (41) can be converted into the following equivalent form.

$$\min_{\lambda} L(\lambda) = \lambda^T \mathbf{S}_+ \lambda + \mathbf{v}^T \lambda + \mu_{\min} \lambda^T \mathbf{I} \lambda \quad \text{s.t. } \lambda \in \mathbf{conv}_m, \tag{42}$$

where $\mathbf{S}_+ = \mathbf{S} - \mu_{\min} \mathbf{I}$ is a positive semidefinite matrix, and the first term of the objective function in problem (42) is convex w.r.t λ . The second term $\mathbf{v}^T \lambda$ is linear w.r.t λ . The third term $\mu_{\min} \lambda^T \mathbf{I} \lambda$ is concave w.r.t λ since $\mu_{\min} < 0$. Therefore, the optimization problem (42) is concave-convex quadratic programming (CCQP) problem. Such a problem can be solved by transforming the concave part of the objective function into a series of convex quadratic programming problems, for details, see the literature (Yuille and Rangarajan 2003).

As seen from subproblem (41), three factors jointly guide the learning of the weights of clustering results.

1. Through the graph smoothness term, the predicting label matrix \mathbf{Z} obtained in the last iteration provides guidance information that is encoded in $\mathbf{v}^{\text{smooth}}$.
2. The supervision information expressed by the “must link” constraint provides guidance information that is encoded in \mathbf{v}^{ml} .
3. The supervision information expressed by the “cannot link” constraint also provides guidance information that is encoded in \mathbf{S} .

3.5 The framework of the GSSL-IQGD algorithm

In this section, the framework of the GSSL-IQGD algorithm is described in Algorithm 1.

Algorithm 1 GSSL-IQGD(Graph-based Semi-supervised Learning via Improving the Quality of the Graph Dynamically)

Input: Semi-supervised data set: $D_l = \{(x_i, y_i)\}_{i=1}^l$, $D_u = \{x_j\}_{j=l+1}^n$; The strategy that produces clustering results; The label inference algorithm $A_{\text{label inference}}$; The hyper-parameters: γ_{fit} , γ_{smooth} , γ_{ml} and γ_{cl} ;
Output: The class labels of unlabeled samples: $\{y_j\}_{j=l+1}^n$.
 1: Generate m clustering results $\{\mathbf{R}^{(t)}\}_{t=1}^m$ using the input strategy;
 2: Compute \mathbf{v}^{ml} by formula (37), \mathbf{S} by formula (39) and it's smallest eigenvalue μ_{min} ;
 3: Initialize λ by $\lambda_t = \frac{1}{m}$, $t = 1, 2, \dots, m$ and the graph $\mathbf{W} = \frac{1}{m} \sum_{t=1}^m \mathbf{R}^{(t)}$;
 4: **repeat**
 5: Update \mathbf{Z} by calling $A_{\text{label inference}}$ with \mathbf{W} , γ_{fit} and γ_{smooth} as input;
 6: Update $\mathbf{v}^{\text{smooth}}$ and \mathbf{v} by formula (36) and (38) respectively;
 7: **if** $\mu_{\text{min}} \geq 0$ **then**
 8: Update λ by CQP [60];
 9: **else**
 10: Update λ by CCQP [61];
 11: **end if**
 12: Update graph by $\mathbf{W} = \sum_{t=1}^m \lambda_t \mathbf{R}^{(t)}$;
 13: **until** λ no longer changes // In practice, set a threshold on $\|\lambda^{(k+1)} - \lambda^{(k)}\|_2$ can be used for fast convergence.
 14: Compute the predicting label of unlabeled sample y_j by formula (4), $j = l+1, l+2, \dots, n$.

3.6 Computational complexity analysis

In this section, we provide the analysis on the computational complexity of the Algorithm 1, which consists of three phases, preparation phase (line 1–3), learning phase (line 4–13) and predicting phase (line 14).

In the preparation phase (line 1–3), assume that the time complexity on the clustering algorithm is $O(T_{\text{cluster}})$, then the time complexity on generating m clustering results is $O(mT_{\text{cluster}})^2$. The time complexity on calculating \mathbf{v}^{ml} (see formula (37)) and \mathbf{S} (see formula (39)) are $O(l^2m)$ and $O(l^2m^2n)$ respectively. And the time complexity on calculating the smallest eigenvalue μ_{min} of \mathbf{S} is $O(m^3)$. At last, the time complexity on initializing the \mathbf{W} is $O(mn^2)$. To sum up, the time complexity of the preparation phase is $O(mT_{\text{cluster}} + l^2m + l^2m^2n + m^3 + mn^2)$. In semi-supervised learning, usually l is a small number and $l \ll n$, so it can be simplified to be $O(mT_{\text{cluster}} + m^2n + m^3 + mn^2)$.

In the learning phase (line 4–13), the loop body contains four main steps: updating the predicting label matrix \mathbf{Z} , updating the \mathbf{v} , updating the weights of m clustering results and updating the edge weight matrix \mathbf{W} . The time complexity of these four steps is as follows.

1. The time complexity on updating \mathbf{Z} is depending on the label inference algorithm $A_{\text{label inference}}$. Assume that it's time complexity is $O(T_{\text{label inference}})$.
2. When \mathbf{Z} is given, the time complexity on updating \mathbf{v} (see formula (38)) is $O(mn^2)$.
3. When \mathbf{v} is given, the time complexity on updating λ is depending on whether the \mathbf{S} is a positive semi-definite matrix.
 - (a) If the \mathbf{S} is a positive semi-definite matrix, then updating process of λ is a convex quadratic programming and it can be solved by ellipsoid method in polynomial time. So the time complexity is $O(P(m))$, where $P(\cdot)$ is a polynomial function.

² In practice, we can use different clustering methods to generate different clustering results. In this case, the time complexity of the this stage can be written as $O(m_1T_{\text{cluster}1} + m_2T_{\text{cluster}2} + \dots)$.

Table 2 Basic information of three artificial data sets

Data set	# Samples	# Features	# Classes
Xclara	3000	2	3
Chainlink	1000	3	2
Spiralsquare	1500	2	6

- (b) If the \mathbf{S} is not a positive semi-definite matrix, then λ is updated through a series of convex quadratic programming, so the time complexity is $O(n_{\text{qp}}P(m))$, where n_{qp} is the number of quadratic programming problems involved in the process. Usually n_{qp} can be treated as a constant, so we have $O(n_{\text{qp}}P(m)) = O(P(m))$.

As a result, the time complexity on updating λ is $O(P(m))$.

4. When λ is given, the time complexity on updating \mathbf{W} is $O(mn^2)$.

Assuming that the number of iteration is n_{ite} , the time complexity of learning phase is $O(n_{\text{ite}}(T_{\text{label inference}} + mn^2 + P(m) + mn^2))$. In practice, we can specify a maximum number of iterations as the stopping condition of the algorithm, so n_{ite} can be treated as a constant, and the time complexity of this phase can be simplified to be $O(T_{\text{label inference}} + mn^2 + P(m))$.

In the predicting phase (line 14), the time complexity on calculating the predicting labels of u unlabeled samples is $O(uc)$. Notice that $u < n$ and $c \ll n$, we have $O(uc) = O(n)$.

To sum up, the time complexity of Algorithm 1 is

$$\begin{aligned} &O(mT_{\text{cluster}} + m^2n + m^3 + mn^2) + O(T_{\text{label inference}} + mn^2 + P(m)) + O(n) \\ &= O(mT_{\text{cluster}} + T_{\text{label inference}} + P(m) + m^2n + mn^2) \end{aligned}$$

4 Experiments

In this section, systematic experiments are conducted to illustrate the working mechanism and the effectiveness of the proposed GSSL-IQGD framework.

4.1 Experiments on artificial data sets

To illustrate the working mechanism of the proposed GSSL-IQGD framework, as a specific algorithm under the framework, the SLLI-IQGD algorithm described in formula (30) is selected to perform experiments on the 3 artificial data sets.

4.1.1 Artificial data sets

There are 3 artificial data sets used in this experiment. These data sets can be downloaded from <https://github.com/deric/clustering-benchmark>. The basic information of the three artificial data sets is given in Table 2.

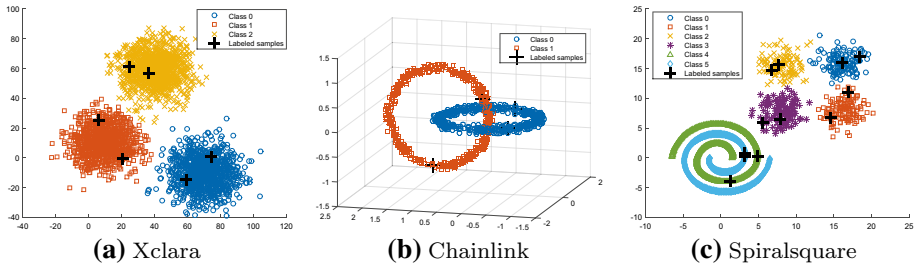


Fig. 1 Three artificial data sets and the selected labeled samples in each class (marked in a dark cross)

These three data sets are all 2-dimensional or 3-dimensional data. Thus, we can observe their distribution via the visualization. Figure 1 shows the distribution of the three artificial data sets.

As seen from Fig. 1a, the Xclara data set is a typical mixed Gaussian distribution containing three components, and each of them corresponds to a class. From Fig. 1b, we can observe that the Chainlink data set contains two typical manifold structures, and each of them corresponds to a class. In Fig. 1c, there are two kinds of data distribution on the Spiralsquare data set. Including four mixed Gaussian clusters and two manifold structures. Therefore, there are in total six classes on the Spiralsquare data set. We randomly selected two samples in each class as labeled samples for each data set, which are marked with the dark cross in Fig. 1.

For the first two data sets shown in Fig. 1a and b, we can construct high-quality graphs that are consistent with the true data distribution by using the appropriate distance metric and parameters. For the Xclara data set, the Euclidean distance should be the best choice, while for the Chainlink data set, the geodesic distance would be better. However, in practical applications, the true data distribution is usually unknown and cannot be visualized since the data's dimension is much larger than three; therefore, the correct distance metric is unknown. Even if we fortuitously choose the correct distance metric, there is still a lack of theoretical guidance on how to choose the graph construction parameters. If the parameters are set improperly, the quality of the graph will still be poor.

For the third Spiralsquare data set shown in Fig. 1c, the situation will be worse. If we use the Euclidean distance, the similarities between samples on the four Gaussian clusters can be calculated correctly, while the similarity on the manifolds will be calculated incorrectly. Moreover, if we switch to the geodesic distance, the situation will be reversed. In this case, it is difficult to construct a high-quality graph by using only one particular distance metric.

From the above analysis, the data distribution in practice is usually unknown, complex and varies from data to data. Most traditional graph construction methods make specific assumptions about the data distribution, so it is difficult to adaptively measure the similarities between samples and build a high-quality graph. The next section elaborates how the SLLI-IQGD algorithm proposed in this paper can adaptively discover the data distribution via fusing multiple clustering results, and then coping with the complexity and variety of the unknown data distribution.

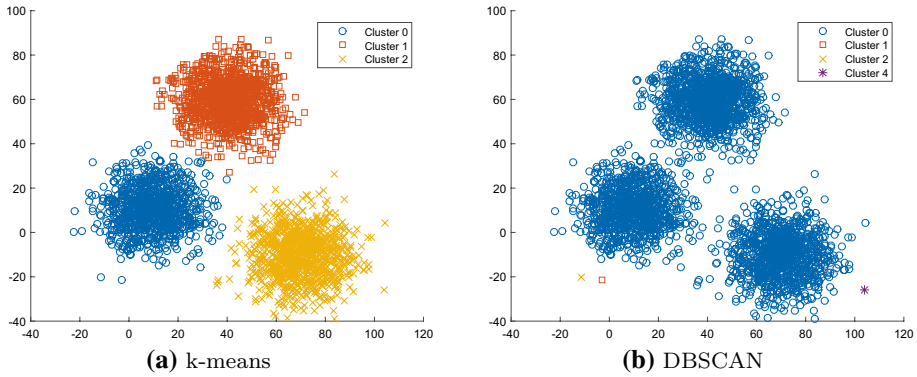


Fig. 2 k-means and DBSCAN clustering results on Xclara data set

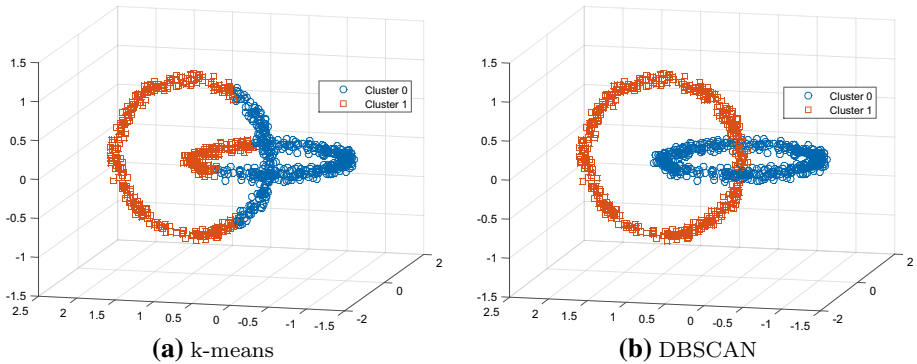


Fig. 3 k-means and DBSCAN clustering results on Chainlink data set

4.1.2 Experimental setting

For each data set, the k-means (Jain 2010) and DBSCAN (Ester et al. 1996) algorithms are used to obtain different clustering results. For the k-means algorithm, the number of clusters is fixed to the number of classes c , and the initial class centers are selected randomly. For each data set, the algorithm repeats five times to obtain five clustering results. For the DBSCAN algorithm, the parameter $MinPts$ is fixed to be 3 and the parameter Eps is determined by the given different noise ratios of samples. By setting different noise ratios ($\{0\%, 0.01\%, 0.03\%, 0.05\%, 0.07\%\}$ for Xclara and $\{0\%, 0.1\%, 0.3\%, 0.5\%, 0.7\%\}$ for Chainlink and Spiralsquare), five different clustering results are obtained for each data set. According to the above settings, a total of $5 + 5 = 10$ different clustering results are obtained for each data set.

In these three toy examples, the hyper-parameters of the SLLI-IQGD algorithm are set to be $\gamma_{fit} = \gamma_{smooth} = \gamma_{ml} = \gamma_{cl} = 0.5$.

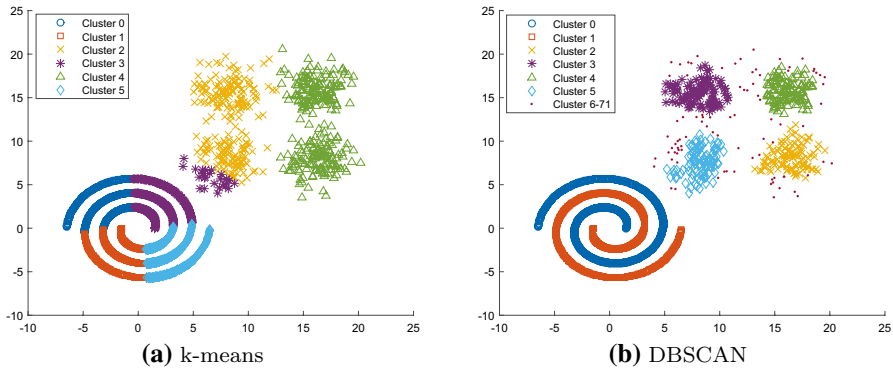


Fig. 4 k-means and DBSCAN clustering results on Spiralsquare data set. Note that only the top 6 biggest clusters of the 72 clusters obtained by DBSCAN algorithm are given in sub-figure **b**

Table 3 The classification error rate and the weight vector λ^* on three artificial data sets

Data set	Error rate	λ^*
Xclara	2.34×10^{-3}	(0.20, 0.20, 0.20, 0.20, 0.20, $1.45 \times 10^{-24}, 1.45 \times 10^{-24}, 1.24 \times 10^{-24}, 1.65 \times 10^{-24}, 1.65 \times 10^{-24})^T$
Chainlink	0	$(1.29 \times 10^{-26}, 1.29 \times 10^{-26}, 1.29 \times 10^{-26}, 1.29 \times 10^{-26}, 1.29 \times 10^{-26},$ 0.50, 0.50, 2.91 $\times 10^{-26}, 2.58 \times 10^{-26}, 2.58 \times 10^{-26})^T$
Spiralsquare	2.02×10^{-3}	$(1.21 \times 10^{-27}, 1.21 \times 10^{-27}, 1.62 \times 10^{-27}, 1.62 \times 10^{-27}, 1.62 \times 10^{-27},$ $6.06 \times 10^{-28}, 1.41 \times 10^{-27}, 1.92 \times 10^{-27}, 1.92 \times 10^{-27}, \mathbf{1})^T$

4.1.3 Experimental results and analysis

The representative clustering results on the three data sets are given in Figs. 2–4.

As seen from Fig. 2, for the Xclara data set, the k-means algorithm can discover the data distribution structure correctly, while the DBSCAN algorithm does not work effectively. For the Chainlink data set, the situation is reversed (see Fig. 3), i.e., the DBSCAN algorithm can discover the data distribution structure correctly, while the k-means algorithm does not work well. For the Spiralsquare data set, neither of the two clustering algorithms can discover the true data distribution structure (see Fig. 4). However, each clustering algorithm can discover a part of the data distribution structure.

It can be seen from the above results that different clustering algorithms are good at mining different data distribution. Therefore, different data distribution can be discovered by using different clustering algorithms. By fusing multiple clustering results reasonably, the similarities between samples can be measured adaptively, and then the constructed graph will be of high quality.

The classification error rate and the learned weight vector λ^* of the clustering results on three artificial data sets are given in Table 3. In the last column of Table 3, the former five numerical values are the weights of the clustering results obtained by the k-means algorithm, and the latter five numerical values are the weights of the clustering results

Table 4 Classification error rate and change of λ after each iteration on three artificial data sets

Number of iterations: i		0	1	2	3	4
Xclara	Error rate	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}	2.34×10^{-3}
	$\ \lambda^{(i)} - \lambda^{(i-1)}\ _2$	N/A	3.16×10^{-1}	6.21×10^{-17}	4.14×10^{-25}	0
Chainlink	Error rate	3.47×10^{-1}	0	0	0	0
	$\ \lambda^{(i)} - \lambda^{(i-1)}\ _2$	N/A	6.32×10^{-1}	1.99×10^{-14}	3.98×10^{-14}	7.96×10^{-15}
Spiralsquare	Error rate	4.70×10^{-3}	2.69×10^{-3}	2.02×10^{-3}	2.02×10^{-3}	
	$\ \lambda^{(i)} - \lambda^{(i-1)}\ _2$	N/A	9.49×10^{-1}	2.29×10^{-26}	0	

obtained by the DBSCAN algorithm. The largest element in the weight vector is marked in bold.

It can be seen from Table 3 that these clustering results that contain the true data distribution are given the largest weight on all data sets (marked in bold) in the learned weight vector λ^* . Specifically, on the Xclara data set, the five clustering results obtained by k-means are given the largest weight 0.20, while the other five clustering results obtained by DBSCAN are given the weight very close to 0. On the Chainlink data set, the first two clustering results obtained by DBSCAN are given the largest weight 0.50, while the five clustering results obtained by k-means and the other three clustering results obtained by DBSCAN are given weight close to 0. On the Spiralsquare data set, the largest weight 1 is given to the last clustering result obtained by DBSCAN, while the other nine clustering results are given the weight close to 0. This result is consistent with the clustering results shown in Figs. 2–4. As a result, the graph constructed by the weighted fusion of multiple clustering results can measure the similarities between samples correctly, and the classification error rate of the SLLI-IQGD algorithm is very low on all three data sets.

To further illustrate that the quality of the graph is gradually improved during the iterative learning process, Table 4 records the classification error rate of the unlabeled samples after each iteration of the SLLI-IQGD algorithm and the ℓ_2 norm of the difference between the weight vectors λ obtained by two successive iterations on three artificial data sets.

It can be seen from Table 4 that on artificial data sets, after a few iterations, the ℓ_2 norm of the difference between the weight vectors obtained by successive iterations is rapidly reduced to 0 or very close to 0, i.e., the weight vector converges to the final result λ^* .

As seen from the first row of Table 4, on the Xclara data set, when the weight of each cluster is initialized to be equal, the classification error rate of the result of label inference on the corresponding graph is 2.34×10^{-3} , that is very close to 0. With the increases of the number of iteration, the weights of the former five clustering results generated by the k-means algorithm are continuously increased, the weights of the latter five clustering results generated by the DBSCAN algorithm are continuously reduced. In the process of iterative learning, the classification error rate does not decrease significantly. This result is related to the clustering results. The former five clustering results generated by the k-means algorithm on the Xclara data set can correctly capture the data distribution, while the latter five clustering results generated by the DBSCAN algorithm assign almost all the samples into one cluster, which is trivial and cannot capture any data distribution information. Therefore, the classification error rate does not change significantly during the iteration. It can be seen from the second row and the third row of Table 4 that as the number of

Table 5 Basic information of 10 benchmark data sets

ID	Data set	# Samples	# Features	# Classes	Source
D1	Iris	150	4	3	UCI ^a
D2	Wine	178	13	3	UCI ^a
D3	Glass	214	9	7	UCI ^a
D4	Mammographic Mass	961	5	2	UCI ^a
D5	Image Segmentation	2310	19	7	UCI ^a
D6	Waveform	2746	21	3	UCI ^a
D7	Set = 5(g241c)	1500	241	2	SSL-book benchmarks ^b
D8	Set = 7(g241n)	1500	241	2	SSL-book benchmarks ^b
D9	Pubmed	19717	500	3	linqs-data ^c
D10	MNIST	35000 ^d	784	10	Yann LeCun's home page ^e

^a <http://archive.ics.uci.edu/ml/index.php>

^b <http://olivier.chapelle.cc/ssl-book/benchmarks.html>

^c <https://linqs-data.soe.ucsc.edu/public/>

^d There are 70,000 samples in the original data set. In this experiments, 35,000 samples are randomly selected so as to complete the experiments at affordable time and space costs.

^e <http://yann.lecun.com/exdb/mnist/>

iterations increases, the weights of the clustering results is continuously adjusted, and the classification error gradually decreases to 0 or very close to 0.

Recall that the error rate of the result of the label inference is the most credible criterion to evaluate the quality of the graph. Therefore, we can conclude that in the three toy examples, the quality of the graph is gradually improved during the iterative process.

These three toy examples illustrate the effectiveness of the proposed framework and explain why it works well. In the next section, we will evaluate three algorithms under the proposed framework through a large number of comparative experiments on benchmark data sets.

4.2 Compared with GSSL based on static graph construction methods

In this section, a large amount of comparison experiments are conducted to verify the effectiveness of the proposal from two perspectives. Specifically, the experiments in this section are designed based on the following two questions. Compared with the commonly used static graph construction methods, can the proposed IQGD method construct better graphs so as to obtain better GSSL results? Whether the proposed GSSL-IQGD framework is a general framework, i.e. can different GSSL methods be embedded into it to improve their performances?

4.2.1 Data sets

A total of 10 data sets are used in this experiment. The basic information of them is shown in Table 5. Among these 10 data sets, the number of samples ranges from several hundred to thirty-five thousand. The number of features ranges from several to more than seven hundred. Both binary and multi-class classification tasks are included.

For the first 8 data sets, 10 different ratios $\{1\%, 2\%, \dots, 10\%\}$ of samples are selected as labeled samples. And for the last 2 data sets, 5 different number $\{1, 5, 10, 15, 20\}$ of samples are selected randomly from each class as labeled samples. The rest of the samples are used as the unlabeled samples. It should be pointed out that the number of samples on D9 and D10 is relatively large (an order of magnitude larger than the first 8 data sets). If the ratio of labeled samples is still set to $\{1\%, 2\%, \dots, 10\%\}$, then the total number of labeled samples will be relatively large, which is contradictory to the idea of semi-supervised learning. For every ratio (or number) of labeled samples, the labeled samples are randomly selected whose class proportion is equal (or approximately equal) to the whole data set. These experiments are repeated 10 times.

4.2.2 Comparison methods and experimental setting

A. Label inference methods for comparison

The focus of this paper is how to construct a high-quality graph to improve the performance of the GSSL method. For this goal, this paper proposes a framework for dynamically improving the quality of the graph for GSSL. In order to verify the effectiveness of the proposed IQGD methods. In this experiment, three of the most representative label inference methods in GSSL, the Harmonic (Zhu et al. 2003), LLGC (Zhou et al. 2003) and SSC-GCN (Kipf and Welling 2017) are selected as the comparison methods. These GSSL methods are selected for two purposes. First, they are combined with various representative graph construction methods to serve as comparison methods. Second, they are embedded into the proposed framework to verify whether the proposed framework can construct higher quality graphs to get better label inference results.

The first two are traditional methods, and the third method is based on graph neural networks. The Harmonic method can only deal with the binary classification problem. In this experiment, the “One vs Rest” strategy is employed to extend the Harmonic method to multi-class classification tasks. For the LLGC method, the hyper-parameter α is set to be 0.99 throughout this experiment, which is recommended in the literature (Zhou et al. 2003). For the SSC-GCN method, a two-layer neural network is used in the experiment which is the same as literature (Kipf and Welling 2017). Unlike the literature (Kipf and Welling 2017), no validation set is used to assist training, because the number of labeled samples is limited in semi-supervised learning.

B. Graph construction methods for Comparison

As mentioned above, the focus of this paper is how to construct a high-quality graph. In order to verify the effectiveness of the proposed IQGD method. In this experiment 4 different common used graph construction methods. Including the kNN graph, see formula (8), the b -matching graph, see formula (10), the ℓ_1 graph, see formula (12, 14) and the LRR graph, see formula (15, 16) are selected as the comparison methods.

Among the 4 methods, the former 2 methods are both graph construction methods based on the distance metric. The kNN graph is the simplest but most frequently used graph construction method in GSSL. In addition, the b -matching graph is a regularized neighbor graph such that every node has the same degree equal to b , which can overcome the adverse effects of the due to the large differences between node degrees. The latter 2 methods are based on data representation. They can simultaneously learn the adjacency structure and

Table 6 The setting of four graph construction methods

Method	Hyper-parameter	Distance metric	Abbreviation
kNN graph	$k = 3$	{ <i>Euclidean, Manhattan, Cosine</i> }	{ <i>E3NN, M3NN, C3NN</i> }
$b - matching$ graph	$b = 3$	{ <i>Euclidean, Manhattan, Cosine</i> }	{ <i>E3M, M3M, C3M</i> }
ℓ_1 graph	N/A	N/A	$\ell_1 - G$
LRR graph	$\lambda = 4^a$	N/A	$LRR - G$

^a It was recommended in the literature (Liu et al. 2013)

Table 7 The setting of k-means and DBSCAN

Algorithm	# Clustering results	Setting
k-means	$5 \times 10 = 50$	$k: \{c, 2c, 3c, 4c, 5c\}^a$ initial cluster centers: randomly selected and repeat 10 times
DBSCAN	$2 \times 6 = 12$	<i>MinPts</i> : {3, 5} <i>Eps</i> : determined by 6 different noise ratios {0%, 1%, ..., 5%}

^a Where c is the number of categories. In unsupervised clustering, it is hard to determine the correct number of clusters. However, in semi-supervised learning the c is known in advance

the edge weight matrix of the graph. The ℓ_1 graph learns the linear representation coefficients for every sample individually with a sparse regularization term, while the LRR graph learns the linear representation coefficients for all samples at the same time with a low-rank regularization term.

The parameter settings for the 4 kinds of graph construction methods are given in Table 6. As seen from the table, there are a total of $3 + 3 + 1 + 1 = 8$ graph construction methods. Combined with the 3 label inference methods, we obtain $3 \times 8 = 24$ comparison methods in the experiment. For the last two data sets, only 3 kinds of kNN graphs are used for comparison, because the construction of the remaining 5 kinds of graphs is very time-consuming, each graph cannot be constructed within 15 days.³

C. The proposed methods

The setting of the proposed method in this paper includes three aspects, i.e., the selection of the label inference algorithm, the way to generate clustering results and the setting of the hyper-parameters.

First, for the selection of the label inference algorithm $A_{\text{label inference}}$, 3 label inference algorithms, including the Harmonic (Zhu et al. 2003), LLGC (Zhou et al. 2003), and SSC-GCN (Kipf and Welling 2017) are selected and embedded in the GSSL-IQGD framework.

³ We run these experiments on a PC with Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.1 GHz and 512 GB RAM.

As a result, 3 corresponding methods, Harmonic-IQGD, LLGC-IQGD, SSC-GCN-IQGD and SLLI-IQGD are obtained. By comparing the Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD with the Harmonic, LLGC and SSC-GCN respectively, the effectiveness of IQGD method can be directly observed.

Second, k-means (Jain 2010) and DBSCAN (Ester et al. 1996) are employed for generating clustering results. And the setting of them is shown in the Table 7. According to the setting, there are total $50 + 12 = 62$ clustering results that are obtained for each data set, i.e. in Algorithm 1, $m = 62$. These clustering results are fused for graph construction, and their fusion weights are dynamically adjusted to improve the quality of the graph.

In practice, if there is more prior knowledge about the data distribution, choosing the appropriate clustering method can achieve better results. If there is little prior knowledge of the data distribution, different kinds of clustering algorithms can be used to cover as many kinds of data distribution as possible. Regardless of how the clustering results are generated, the proposed method can adaptively fuse them to construct a high-quality graph for GSSL. In this experiment, only two kinds of classical clustering methods are selected to discover the potential data distribution, and they are enough to verify the effectiveness of the proposed GSSL-IQGD framework.

Third, the simplest settings are used for the hyper-parameters throughout this experiment⁴. In the Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD methods, $\gamma_{\text{smooth}} = \gamma_{\text{ml}} = \gamma_{\text{cl}} = 0.5$ is employed for updating the weight vector λ . The relative magnitudes between γ_{fit} and γ_{smooth} in Harmonic-IQGD and LLGC-IQGD methods are set to be the same as their counterparts, i.e. the Harmonic and LLGC, respectively.

4.2.3 Experimental results and analysis

Tables 8–15 show the classification error rate for the proposed methods and comparison methods on the first 8 data sets.

In Tables 8–15, there are 10 columns in every table, and each column corresponds to a labeled samples ratio. Each row in the table corresponds to a method and there are a total of 27 methods that are used for comparison.

These 27 methods can be divided into 3 groups. In the first group, the 1st–9th rows in the table, show the methods obtained by combining the Harmonic with different graph construction methods. In the second group, the 10th–18th rows in the table, show the methods obtained by combining the LLGC with different graph construction methods. In the third group, the 19th–27th rows in the table, show the method obtained by combining the SSC-GCN with different graph construction methods.

In the three groups, the Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD are obtained by embedding the corresponding GSSL methods into the proposed GSSL-IQGD framework. In each group, the method of graph label inference is the same, and the only difference is in the method of graph construction. Therefore, it is straightforward to demonstrate the effectiveness of the proposal by comparing the Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD with their counterparts, respectively.

⁴ Generally, some strategies such as cross-validation can be used to tune hyper-parameters. However it is impractical in semi-supervised learning because the number of labeled samples is limited.

Table 8 Data set D1 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	50.27	52.04	45.97	46.74	40.57	36.67	34.86	35.07	31.93	29.11
<i>M3NN</i>	47.42	48.44	45.00	40.28	35.39	36.03	35.65	34.86	30.96	32.59
<i>C3NN</i>	59.52	60.14	53.68	53.33	49.65	46.31	42.61	43.33	39.48	42.59
<i>E3M</i>	32.18	32.24	26.94	27.71	20.21	18.51	18.26	17.54	17.63	18.44
<i>M3M</i>	28.23	23.33	19.03	20.00	17.45	14.75	11.52	12.90	10.59	11.70
<i>C3M</i>	19.18	18.91	12.08 ³	15.97	10.78	11.13	10.00	11.67	7.78	7.41 ²
$\ell_1 - G$	52.99	54.01	57.01	52.36	54.47	45.82	43.91	43.12	38.00	34.52
<i>LRR - G</i>	60.75	64.22	65.07	61.11	60.43	53.76	50.36	46.09	48.00	47.33
IQGD	22.86 ⁵	15.44 ³	30.14 ¹⁰	16.04 ⁵	15.67 ⁵	10.21 ³	5.00 ¹	10.00 ³	7.33 ³	11.19 ⁵
<i>LLGC</i>										
<i>E3NN</i>	32.99	35.92	28.47	26.46	27.02	24.89	21.59	18.91	14.00	17.41
<i>M3NN</i>	33.54	35.10	34.03	30.83	20.35	23.48	20.65	23.04	14.52	19.63
<i>C3NN</i>	42.18	48.44	36.04	29.17	34.61	26.24	23.19	20.94	24.74	23.33
<i>E3M</i>	31.16	31.70	25.63	25.56	23.62	19.15	18.19	18.62	18.15	17.93
<i>M3M</i>	28.84	24.76	20.07	20.63	19.93	17.31	13.19	16.88	12.89	13.85
<i>C3M</i>	15.99 ³	19.80	14.10	14.44 ³	10.07 ²	10.64	9.06	11.38	10.22	10.67
$\ell_1 - G$	49.25	55.51	60.35	54.93	60.07	61.70	62.97	58.33	55.04	55.48
<i>LRR - G</i>	61.22	64.22	64.65	60.97	66.45	64.40	58.70	57.83	64.89	63.04
IQGD	12.24 ²	10.61 ²	10.56 ²	13.47 ²	10.07 ²	8.30 ²	5.51 ³	9.57 ²	5.85 ²	8.07 ³
<i>SSC-GCN</i>										
<i>E3NN</i>	46.33	41.84	44.24	43.26	41.84	34.40	47.75	44.49	35.41	38.44
<i>M3NN</i>	42.93	41.97	38.47	48.06	41.14	38.30	35.36	39.20	34.15	40.00
<i>C3NN</i>	31.63	53.67	38.13	45.14	35.60	51.06	48.70	44.57	44.22	54.44
<i>E3M</i>	44.76	45.99	46.25	43.33	42.55	40.50	50.15	45.22	50.74	48.82
<i>M3M</i>	59.93	37.76	46.18	43.75	30.57	49.22	41.45	34.86	35.19	31.26
<i>C3M</i>	46.87	33.61	52.64	34.51	32.13	47.87	47.10	45.36	28.44	39.11
$\ell_1 - G$	55.31	51.09	50.97	33.40	59.22	52.77	59.93	48.33	53.70	46.30
<i>LRR - G</i>	54.90	58.98	52.99	51.67	48.30	58.16	55.80	50.73	54.44	53.93
IQGD	7.82 ¹	6.87 ¹	7.15 ¹	5.07 ¹	7.80 ¹	5.82 ¹	5.00 ¹	6.09 ¹	4.22 ¹	3.48 ¹

In each table, the ranks of the 3 methods under the proposed GSSL-IQGD framework and the top 3 methods are marked with digital superscripts. The rank of each method is used to compare the performance of each method directly.

Table 8 records the classification error rate of the 27 methods on D1 data set. The following results can be drawn from this table directly.

1. At labeled ratios of 2%, 6%, 7%, 8% and 9%, the Harmonic-IQGD method defeats its counterparts with 8 different graph construction methods; At labeled ratios of 1%, 4%, 5% and 10%, the Harmonic-IQGD defeats its 7 counterparts and is defeated by the Harmonic method with the *C3M* graph; At the labeled ratio of 3%, the Harmonic-IQGD

defeats its 5 counterparts and is defeated by the Harmonic method with the $b - matching$ graph using 3 different distance metrics. To sum up, when the label inference method is fixed to Harmonic, the Harmonic-IQGD method obtained by embedding the Harmonic into the proposed GSSL-IQGD framework can achieve better performance compared with the Harmonic method using 8 different graphs in most cases. These results show that the proposed IQGD method can indeed construct a higher-quality graph compared with 8 different graph construction methods.

2. The LLGC-IQGD method defeats its counterparts with 8 different graph construction methods at 10 different labeled ratios. These results indicate that when the label inference method is fixed to LLGC, compared with the 8 different graph construction methods, the IQGD method can construct a better graph and then the better label inference performance can be achieved.
3. At 10 different labeled ratio, the SSC-GCN-IQGD defeats its 8 counterparts with significant advantage. These results indicate that when the label inference method is fixed to SSC-GCN, compared with the 8 different graph construction methods, the proposed IQGD method can also construct a better graph, and then the better label inference performance can be achieved. It can also be seen from these results that as one of the most advanced methods at present, the performance of SSC-GCN also heavily depends on the quality of the graph. And by embedding it into the proposed framework, its performance can also be improved significantly.
4. The Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD methods take the top 3 places among the 27 methods at labeled ratios of 2%, 6%, 7%, 8% and 9%, and in the remaining 5 labeled ratios, two of them ranked in the top 3 places. These results show that by embedding the Harmonic, LLGC and SSC-GCN into the proposed GSSL-IQGD framework respectively, the performance of all three methods can be improved significantly. That is to say, the proposed method is a general framework.

The comparison results on D2, D3, D4 and D5 data sets, as shown in Tables 9, 10, 11 and 12, are similar to the result on D1 data set and are summarized as follows. The results on each data set are no longer described separately.

1. When the label inference method is fixed to Harmonic, the Harmonic-IQGD method obtained by embedding the Harmonic into the proposed GSSL-IQGD framework can achieve better performance compared with the Harmonic method using 8 different graphs in most cases. The results on LLGC and SSC-GCN methods are virtually the same as on the Harmonic. These results show that, compared with the 8 representative graph construction methods, the proposed IQGD method is an effective graph construction method.
2. The Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD methods take the top 3 places or two of them ranked in the top 3 places among the 27 methods in almost all cases. These results show the proposed method is a general framework since different GSSL methods can improve their performance by embedding them into it.

For the D6, D7 and D8 data sets, as shown in Tables 13, 14 and 15, the comparison results are summarized as follows.

Table 9 Data set D2 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	61.77	60.81	56.08	56.47	53.39	53.13	52.32	47.24	48.50	47.30
<i>M3NN</i>	62.74	61.50	53.74	55.47	49.46	51.81	50.98	47.61	46.69	47.67
<i>C3NN</i>	61.77	58.84	52.75	55.12	50.83	51.69	47.99	45.89	46.63	44.84
<i>E3M</i>	47.31	47.34	37.54	42.35	40.24	38.19	36.16	32.82	35.13	37.36
<i>M3M</i>	50.69	47.28	38.30	38.65	37.68	37.83	34.82	32.76 ³	33.81 ³	36.10
<i>C3M</i>	44.46	40.40 ³	37.78	37.24 ²	36.01	36.08	34.33 ²	34.85	35.13	34.28
$\ell_1 - G$	64.34	60.06	53.04	48.65	48.87	49.64	42.62	41.35	38.19	46.42
<i>LRR - G</i>	63.71	66.36	58.95	67.06	62.02	58.43	67.07	66.87	65.31	50.88
IQGD	48.97 ¹¹	56.07 ¹⁶	36.02 ²	42.59 ¹⁰	35.89 ³	44.94 ¹¹	34.39 ³	34.72 ⁸	35.56 ⁸	35.79 ⁷
<i>LLGC</i>										
<i>E3NN</i>	59.37	58.32	50.23	52.18	48.93	48.86	46.95	42.76	46.75	42.08
<i>M3NN</i>	58.91	58.27	49.36	52.00	45.18	49.76	47.99	43.93	44.63	41.51
<i>C3NN</i>	55.43	49.36	36.78	41.76	40.95	42.59	40.79	36.56	37.81	38.05
<i>E3M</i>	47.37	46.47	38.19	43.18	39.35	36.81	36.59	32.82	35.31	35.66
<i>M3M</i>	48.97	44.80	36.37 ³	39.12	36.96	35.60 ³	35.18	33.44	34.44	35.03
<i>C3M</i>	46.06	41.85	38.25	38.18	38.51	39.64	36.40	33.01	36.94	33.52 ³
$\ell_1 - G$	61.94	61.04	59.82	56.88	59.40	59.04	57.50	56.44	59.63	60.38
<i>LRR - G</i>	63.54	67.05	60.23	67.06	61.43	55.96	67.07	62.88	60.31	57.86
IQGD	46.40 ⁸	41.10 ⁴	36.78 ⁴	34.24 ¹	34.40 ²	36.39 ⁵	36.22 ⁸	31.90 ²	32.44 ²	33.21 ²
<i>SSC-GCN</i>										
<i>E3NN</i>	52.38	48.44	46.11	50.56	52.13	50.14	49.20	63.12	57.63	44.44
<i>M3NN</i>	50.88	58.37	56.25	57.08	48.09	52.91	55.44	51.45	50.44	49.33
<i>C3NN</i>	44.08	54.01	42.36	63.19	49.01	48.58	56.23	47.32	45.11	59.56
<i>E3M</i>	43.95 ³	43.20	49.72	51.46	52.84	55.75	54.93	53.12	57.48	50.44
<i>M3M</i>	49.12	52.11	44.31	57.57	65.53	47.31	66.96	62.83	56.37	72.89
<i>C3M</i>	56.40	64.76	45.76	48.96	53.48	73.33	63.55	49.71	63.70	48.22
$\ell_1 - G$	37.28 ¹	38.78 ²	37.36	39.79	38.01	33.19 ¹	36.09	42.75	44.59	40.07
<i>LRR - G</i>	42.18 ²	44.76	51.88	46.53	47.73	51.28	50.65	54.28	49.93	43.78
IQGD	44.34 ⁵	37.69 ¹	35.73 ¹	38.00 ³	34.35 ¹	33.92 ²	34.33 ¹	30.98 ¹	31.06 ¹	32.45 ¹

1. The Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD methods defeat their counterparts the Harmonic, LLGC and SSC-GCN with 8 different graphs at 10 different ratios of labeled samples, respectively. These results show that compared with the 8 representative graph construction methods, the proposed IQGD method can construct a better graph, as a result, the performance of the corresponding GSSL is improved.
2. The 3 methods, Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD under the GSSL-IQGD framework take the top 3 places among the 27 methods at 10 different ratios of labeled samples. These results show that the proposed framework can be applied to both the three GSSL methods and the expected performance gains are achieved.

Table 10 Data set D3 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	66.92	64.03	65.29	59.36	58.46	62.53	59.49	53.78	56.41	55.97
<i>M3NN</i>	61.68 ²	60.68	59.46	55.69	55.07	59.19	54.23	51.04	52.03	53.51
<i>C3NN</i>	64.81	62.33	63.43	57.38	54.83	60.15	55.51	52.28	54.01	53.25
<i>E3M</i>	73.27	61.99	59.07	55.64	52.59	57.02	53.01	48.19	49.38	47.02
<i>M3M</i>	67.16	60.10	60.10	55.20	54.38	57.12	50.41	50.41	49.06	48.95
<i>C3M</i>	67.69	58.88	56.42	55.25	49.80 ³	53.64	48.32 ³	49.38	46.09	45.24
$\ell_1 - G$	72.07	64.95	59.46	53.47	52.44	53.38	49.03	49.74	49.69	48.27
<i>LRR - G</i>	80.00	63.93	63.77	64.36	63.48	63.23	64.29	63.32	61.98	64.29
IQGD	62.69 ⁴	53.50 ³	52.35 ²	49.51 ²	48.41 ¹	50.15 ²	45.31 ²	43.73 ²	45.89 ³	42.09 ²
<i>LLGC</i>										
<i>E3NN</i>	70.58	63.93	68.43	56.78	55.77	63.28	56.12	54.30	54.22	53.87
<i>M3NN</i>	66.59	58.11	61.27	51.24 ³	52.34	58.74	49.39	50.57	48.39	50.89
<i>C3NN</i>	70.72	60.05	62.70	54.95	53.73	59.60	53.21	50.36	50.63	50.21
<i>E3M</i>	72.07	64.81	59.90	57.77	55.77	56.57	52.45	52.90	50.00	49.95
<i>M3M</i>	68.41	62.28	60.00	54.21	56.57	57.88	51.48	54.40	50.42	52.20
<i>C3M</i>	65.77	56.21	55.78	53.86	50.40	52.73	48.78	47.20 ³	44.58 ²	44.19 ³
$\ell_1 - G$	78.89	63.88	62.99	60.74	61.54	62.27	60.87	60.47	62.08	63.14
<i>LRR - G</i>	84.09	64.22	64.90	64.36	63.73	64.65	64.29	64.25	63.54	64.40
IQGD	58.89 ¹	53.06 ²	53.63 ³	45.50 ¹	48.56 ²	48.94 ¹	44.64 ¹	42.80 ¹	42.14 ¹	39.63 ¹
<i>SSC-GCN</i>										
<i>E3NN</i>	68.85	70.29	69.17	66.44	60.95	69.90	65.36	63.06	61.46	57.70
<i>M3NN</i>	67.50	66.70	65.83	59.55	62.19	66.67	64.90	66.32	61.35	61.36
<i>C3NN</i>	70.34	66.17	63.87	64.80	61.69	66.26	61.53	61.71	63.07	62.72
<i>E3M</i>	73.41	68.69	66.67	60.84	59.65	67.98	60.61	55.13	65.42	60.84
<i>M3M</i>	72.55	64.85	64.76	59.90	57.31	60.61	63.88	64.25	60.26	60.58
<i>C3M</i>	72.21	69.56	58.09	62.57	58.76	60.00	59.44	59.69	61.51	57.33
$\ell_1 - G$	73.75	65.49	63.38	59.80	63.23	60.56	64.90	59.90	63.13	57.75
<i>LRR - G</i>	79.81	69.85	66.72	67.92	63.38	64.85	64.69	60.21	62.29	63.35
IQGD	62.12 ³	52.77 ¹	51.76 ¹	53.61 ⁵	51.74 ⁵	50.81 ³	51.73 ⁹	53.06 ¹³	52.60 ¹³	50.42 ¹⁰

Tables 16 and 17 show the classification error rate for the proposed methods and comparison methods on the last 2 data sets. Each table contains 5 columns and 12 rows. Each column corresponds to a labeled number and each row corresponds to a method. Similar to Tables 8–15, these 12 methods can also be divided into 3 groups.

Table 16 shows the classification error rate of the 12 methods on D9 data set. The following conclusions can be drawn from the table.

Table 11 Data set D4 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	52.32	50.98	50.10	49.20	47.96	47.46	46.67	46.01	45.51	44.61
<i>M3NN</i>	52.34	51.01	50.12	49.26	48.09	47.60	46.82	46.18	45.66	44.85
<i>C3NN</i>	52.40	51.35	50.41	49.71	48.73	48.06	47.30	46.57	45.85	45.31
<i>E3M</i>	48.71	46.88	44.86	44.13	42.07	42.18	40.49	39.21	37.90	37.77
<i>M3M</i>	50.88	48.34	45.89	44.65	43.53	43.10	42.20	41.60	40.79	39.79
<i>C3M</i>	46.64	46.14	44.74	43.56	42.24	41.86	41.50	40.60	39.48	38.96
$\ell_1 - G$	43.08	47.43	40.02	39.37	39.07	36.15	33.37 ³	30.09 ¹	27.93 ¹	27.13 ¹
<i>LRR - G</i>	46.32	47.07	44.81	42.04	38.87	38.79	39.75	40.52	38.60	38.94
IQGD	43.84 ¹²	42.93 ¹²	36.68 ³	38.15 ⁷	36.77 ⁴	34.84 ³	34.93 ⁵	35.47 ⁶	33.79 ⁴	33.61 ⁴
<i>LLGC</i>										
<i>E3NN</i>	43.96	43.71	41.68	40.33	38.39	37.64	37.20	36.81	35.25	34.70
<i>M3NN</i>	43.88	43.74	41.64	40.22	38.26	37.60	37.24	36.78	34.91	34.31
<i>C3NN</i>	44.16	43.66	42.50	41.50	39.78	39.15	38.53	37.94	37.07	36.93
<i>E3M</i>	39.98	40.63	38.05	37.39	36.03 ³	34.91	34.41	35.32	34.35	34.33
<i>M3M</i>	41.86	40.49	38.17	37.98	37.02	36.58	37.00	37.61	36.22	36.41
<i>C3M</i>	46.58	46.66	43.51	42.67	41.11	40.76	39.99	39.64	38.57	38.31
$\ell_1 - G$	47.69	46.81	43.99	40.30	43.26	38.15	39.07	38.75	36.98	41.22
<i>LRR - G</i>	46.32	46.33	46.29	46.31	46.27	46.29	46.30	46.32	46.28	46.30
IQGD	37.73 ³	36.11 ³	34.14 ¹	34.03 ²	33.50 ²	33.48 ²	33.33 ²	33.18 ²	33.53 ³	33.31 ³
<i>SSC-GCN</i>										
<i>E3NN</i>	44.08	42.25	48.50	40.02	40.53	42.75	36.83	38.69	39.70	41.00
<i>M3NN</i>	34.73 ¹	33.05 ¹	42.65	37.40	47.17	46.86	47.71	41.88	41.72	44.34
<i>C3NN</i>	45.85	41.52	46.85	40.42	38.67	45.71	40.01	44.04	49.31	41.97
<i>E3M</i>	41.77	43.57	41.43	34.15 ³	44.25	40.16	48.53	35.39	46.90	35.34
<i>M3M</i>	39.56	37.75	40.46	43.74	40.04	47.14	41.22	39.97	39.53	40.80
<i>C3M</i>	43.53	39.00	39.40	44.91	37.94	37.02	40.09	41.29	42.58	41.42
$\ell_1 - G$	40.92	38.43	40.63	40.54	40.67	46.96	38.25	39.75	42.98	40.67
<i>LRR - G</i>	40.90	41.31	44.52	42.05	38.49	40.18	43.92	43.18	37.33	40.68
IQGD	36.08 ²	34.65 ²	34.50 ²	33.31 ¹	33.22 ¹	33.24 ¹	33.23 ¹	33.26 ³	33.26 ²	33.24 ²

1. At 5 different numbers of labeled samples, the Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD under the GSSL-IQGD framework take the top 3 places among the 12 methods.
2. As a direct result, the Harmonic-IQGD, LLGC-IQGD and SSC-GCN-IQGD methods defeat the Harmonic, LLGC and SSC-GCN using the other three graph construction methods, respectively. These results show that compared with the three k NN graphs, the proposed IQGD method can construct a better graph so as to obtain a better GSSL result.

Table 12 Data set D5 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	77.50	73.37	68.02	63.88	61.26	57.91	55.29	52.99	51.20	49.34
<i>M3NN</i>	78.97	74.03	68.77	64.59	60.43	58.82	55.58	54.03	51.68	48.22
<i>C3NN</i>	77.76	71.63	67.23	62.03	59.07	56.12	53.81	50.97	48.73	47.44
<i>E3M</i>	44.70	39.92	33.28	28.63	27.62	26.90	22.05	21.90	20.74	20.13
<i>M3M</i>	42.48	34.92	27.09 ³	25.85 ³	23.07 ³	22.06 ³	19.14 ³	18.24 ²	17.33 ¹	16.33 ¹
<i>C3M</i>	42.39	36.41	30.83	26.45	24.82	23.10	19.48	19.34	17.37 ²	17.73
$\ell_1 - G$	81.67	68.39	58.17	46.75	49.00	35.71	37.19	30.56	29.28	26.96
<i>LRR - G</i>	84.48	85.36	84.95	83.58	82.42	84.72	81.31	79.07	76.03	78.90
IQGD	46.03 ⁹	31.86 ²	23.23 ²	19.79 ²	21.02 ²	19.76 ²	18.94 ²	18.41 ³	18.64 ⁴	17.23 ²
<i>LLGC</i>										
<i>E3NN</i>	64.53	54.18	45.58	42.45	37.23	33.63	30.88	29.87	28.06	27.30
<i>M3NN</i>	65.34	56.29	48.40	41.61	36.34	36.26	31.21	30.14	27.31	25.47
<i>C3NN</i>	66.03	55.71	49.33	42.33	38.06	35.34	30.94	29.66	27.85	26.73
<i>E3M</i>	44.32	40.35	32.30	29.71	28.50	28.08	25.35	23.57	23.88	24.16
<i>M3M</i>	39.91 ³	34.64	29.75	26.17	26.08	24.25	23.31	23.32	22.57	22.04
<i>C3M</i>	40.92	37.18	30.06	26.19	25.52	25.87	20.29	21.03	19.17	20.43
$\ell_1 - G$	75.84	64.20	64.33	64.31	66.97	58.78	61.35	58.75	55.21	54.02
<i>LRR - G</i>	83.64	84.77	85.71	85.66	84.92	85.71	85.70	85.69	85.69	85.71
IQGD	29.38 ¹	23.38 ¹	20.24 ¹	19.67 ¹	19.52 ¹	19.29 ¹	18.37 ¹	18.03 ¹	17.58 ³	17.28 ³
<i>SSC-GCN</i>										
<i>E3NN</i>	79.36	76.13	78.00	81.76	76.32	76.97	77.37	79.32	78.01	76.77
<i>M3NN</i>	77.02	75.87	75.51	75.57	76.75	76.53	80.69	78.58	76.73	75.50
<i>C3NN</i>	75.94	79.21	79.20	78.74	78.28	78.60	77.25	79.45	76.53	75.66
<i>E3M</i>	78.95	79.49	78.73	75.69	79.52	78.48	78.53	82.23	75.20	73.03
<i>M3M</i>	77.85	78.87	77.69	80.42	76.62	76.95	77.35	75.51	74.04	75.44
<i>C3M</i>	76.00	76.41	77.15	77.88	75.50	78.42	69.83	85.39	74.66	77.41
$\ell_1 - G$	82.55	77.09	80.34	83.52	74.08	80.02	82.55	78.99	76.13	80.20
<i>LRR - G</i>	80.62	77.48	79.24	80.47	79.74	79.65	80.05	78.13	83.68	82.59
IQGD	34.46 ²	33.12 ³	31.39 ⁷	30.19 ⁹	27.04 ⁷	28.77 ⁹	28.09 ⁹	29.45 ⁹	27.59 ¹⁰	28.16 ¹³

Table 17 shows the classification error rate of the 12 methods on D10 data set. The following conclusions can be drawn from the table.

1. For the Harmonic-IQGD, although it ranks in the top 3 only once among the 12 methods, it ranks first in its group and is significantly better than other methods in the same group. These results show that when the label inference method is fixed to Harmonic, the proposed IQGD method can construct a better graph compared with the three k NN graph construction methods.
2. The methods in the second group have a good performance on this data set. When the number of labeled samples is 10 and 50, the LLGC-IQGD method ranks first in

Table 13 Data set D6 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	56.18	49.11	49.24	47.78	46.58	45.99	44.11	44.29	44.07	44.72
<i>M3NN</i>	54.79	47.77	48.43	44.90	44.67	43.21	44.01	42.57	42.71	41.87
<i>C3NN</i>	50.51	46.80	47.12	45.25	45.02	44.23	43.40	43.78	43.17	43.23
<i>E3M</i>	27.79	25.65	25.24	23.63	23.33	23.35	22.71	22.63	22.74	22.40
<i>M3M</i>	29.76	27.21	25.06	24.19	23.99	23.53	23.73	23.03	23.22	23.13
<i>C3M</i>	28.33	26.24	25.14	22.94	23.09	23.14	22.70	22.83	22.89	22.42
$\ell_1 - G$	65.54	64.86	62.31	60.88	59.89	56.19	54.94	55.72	54.19	51.80
<i>LRR - G</i>	64.85	65.17	64.81	64.14	64.92	65.10	65.12	64.99	65.16	65.03
IQGD	24.37 ²	21.48 ²	20.03 ¹	19.62 ²	19.10 ²	18.83 ²	18.45 ²	18.30 ²	18.60 ¹	18.32 ²
<i>LLGC</i>										
<i>E3NN</i>	42.15	37.84	40.21	33.28	35.24	32.79	30.91	33.68	30.22	31.29
<i>M3NN</i>	43.54	36.21	37.75	35.34	35.45	34.41	34.29	35.52	32.82	33.34
<i>C3NN</i>	42.67	40.74	47.24	42.84	39.94	40.26	38.88	39.78	39.06	40.83
<i>E3M</i>	26.74	25.14	24.93	23.38	23.53	23.12	22.42	22.50	22.94	22.41
<i>M3M</i>	28.08	26.44	25.10	23.69	23.76	23.38	23.10	23.26	22.60	22.40
<i>C3M</i>	26.66	25.87	24.59	22.47	22.81	22.57	21.69	22.36	22.35	21.85
$\ell_1 - G$	65.37	63.27	63.60	64.27	63.41	63.85	63.16	62.63	64.97	62.80
<i>LRR - G</i>	65.16	65.17	65.15	65.16	65.15	65.16	65.14	65.15	65.16	65.14
IQGD	22.19 ¹	20.10 ¹	20.10 ²	19.26 ¹	18.36 ¹	18.70 ¹	18.34 ¹	18.11 ¹	18.73 ²	18.30 ¹
<i>SSC-GCN</i>										
<i>E3NN</i>	30.18	26.43	25.23	25.96	25.18	23.30	25.26	23.72	23.75	23.15
<i>M3NN</i>	31.93	29.18	26.79	24.57	25.49	24.55	23.78	24.21	23.55	24.33
<i>C3NN</i>	29.46	30.35	26.56	27.35	25.31	23.78	24.42	25.05	24.08	24.96
<i>E3M</i>	32.61	26.52	25.73	25.14	24.81	24.64	22.41	21.99	22.66	23.03
<i>M3M</i>	32.09	27.52	26.44	25.54	24.61	23.55	22.82	22.78	23.06	22.90
<i>C3M</i>	32.68	27.20	25.87	25.31	24.85	23.80	23.81	22.88	23.54	23.10
$\ell_1 - G$	37.67	33.32	36.42	29.21	30.71	28.42	28.49	27.73	25.74	29.23
<i>LRR - G</i>	42.01	41.09	44.95	38.66	38.76	41.68	41.17	35.99	37.49	36.30
IQGD	25.61 ³	23.35 ³	24.36 ³	20.74 ³	22.06 ³	20.43 ³	19.92 ³	20.24 ³	19.78 ³	19.66 ³

this group. And for the remaining number of labeled samples, the LLGC method with three different kNN graphs performs better. These results show that when the number of labeled samples is smaller, the proposed IQGD method outperforms the other three graph construction methods for GSSL.

- For the SSC-GCN-IQGD method, it ranks first in its group at first 4 different numbers of labeled samples and ranks second in the last case. These results show that when the label inference method is fixed to SSC-GCN, the proposed IQGD method can construct a better graph so as to improve the performance of the SSC-GCN method.

Table 14 Data set D7 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	48.34	48.64	48.06	48.46	47.96	48.17	47.83	47.67	47.40	47.10
<i>M3NN</i>	50.05	49.83	49.46	49.38	49.45	49.16	49.00	49.24	48.92	48.21
<i>C3NN</i>	48.58	48.12	47.42	46.41	46.14	44.46	44.71	43.30	43.06	42.69
<i>E3M</i>	49.82	47.77	46.34	44.94	44.63	42.60	43.24	42.75	41.67	42.21
<i>M3M</i>	49.25	47.12	46.37	45.73	46.29	44.74	44.58	44.30	43.78	43.04
<i>C3M</i>	49.41	46.90	44.64	44.40	44.28	42.95	43.13	42.30	41.37	42.01
$\ell_1 - G$	49.93	50.30	50.13	50.12	50.04	50.30	49.55	49.94	50.04	50.90
<i>LRR - G</i>	50.05	50.25	50.20	50.43	50.22	50.43	49.91	50.36	50.68	51.01
IQGD	27.58 ³	24.12 ³	25.88 ³	24.06 ²	24.70 ²	23.39 ³	23.69 ²	23.86 ²	24.02 ²	23.64 ³
<i>LLGC</i>										
<i>E3NN</i>	49.99	49.99	49.17	48.76	49.12	49.48	50.00	49.65	48.95	49.09
<i>M3NN</i>	50.07	49.90	49.65	49.79	49.82	49.18	49.69	49.62	48.92	47.70
<i>C3NN</i>	48.88	48.68	47.39	46.78	45.86	44.55	44.79	46.17	44.76	44.35
<i>E3M</i>	49.95	46.99	45.93	44.42	44.48	42.52	43.22	42.88	41.44	42.04
<i>M3M</i>	49.32	46.66	46.42	45.49	46.30	45.26	44.39	44.44	43.94	43.30
<i>C3M</i>	49.16	46.64	44.37	44.29	44.22	42.81	43.45	42.09	41.22	41.44
$\ell_1 - G$	50.26	50.29	50.17	50.34	49.97	50.18	49.25	50.16	50.04	50.29
<i>LRR - G</i>	50.46	50.30	50.39	50.13	50.12	49.99	49.76	50.38	50.34	50.27
IQGD	23.57 ²	23.15 ²	25.01 ²	24.40 ³	24.71 ³	23.26 ²	23.69 ²	23.86 ²	24.02 ²	23.34 ²
<i>SSC-GCN</i>										
<i>E3NN</i>	49.35	47.06	45.74	45.43	42.56	42.60	44.27	42.22	41.95	40.64
<i>M3NN</i>	50.04	47.37	45.25	44.24	44.87	41.50	42.20	40.36	40.50	39.70
<i>C3NN</i>	48.74	48.12	45.47	43.12	43.01	41.84	42.27	39.97	40.67	40.76
<i>E3M</i>	48.58	45.06	45.72	44.68	43.62	44.85	41.21	41.96	41.44	40.50
<i>M3M</i>	45.90	45.38	44.72	44.77	43.20	42.19	42.48	39.95	41.75	40.10
<i>C3M</i>	48.92	45.39	45.32	44.99	44.14	43.26	41.84	42.86	40.42	40.50
$\ell_1 - G$	46.89	44.83	43.55	43.22	41.16	41.45	40.33	40.32	40.73	39.18
<i>LRR - G</i>	45.11	40.85	37.26	35.68	33.37	32.91	31.89	30.43	30.29	29.82
IQGD	15.53 ¹	13.95 ¹	13.65 ¹	13.15 ¹	13.13 ¹	12.91 ¹	13.04 ¹	12.93 ¹	12.92 ¹	13.02 ¹

To more intuitively exhibit the performance of each method, Fig. 5 shows the mean and standard deviation of the rankings of the 27 methods in the whole experiment. The mean and standard deviation of the ranking of each method are computed through $8 \times 10 = 80$ ranking results (the first 8 data sets and 10 different labeled ratios on each data set). In Fig. 5, each method corresponds to a line segment, the midpoint of the line segment represents the mean of this method’s rank, the length of the line segment represents twice the standard deviation. And the methods in the 3 groups are marked with 3 different colors and shapes.

As seen from Fig. 5, among all the 27 methods, the LLGC-IQGD, SSC-GCN-IQGD and Harmonic-IQGD are the top 3 methods, and are significantly better than the rank in the 4th method LLGC-C3M. On the other hand, the Harmonic-IQGD, LLGC-IQGD

Table 15 Data set D8 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>Harmonic</i>										
<i>E3NN</i>	50.44	50.18	49.95	49.73	49.21	49.48	49.34	48.91	49.33	48.84
<i>M3NN</i>	49.18	48.96	48.43	48.12	48.05	47.59	47.53	47.13	47.18	46.50
<i>C3NN</i>	45.79	45.36	43.60	43.43	42.46	41.90	40.76	40.52	40.95	39.56
<i>E3M</i>	47.59	45.94	43.87	42.36	41.49	41.68	40.35	40.46	39.17	38.87
<i>M3M</i>	47.68	45.28	43.87	43.69	43.37	43.26	41.49	41.23	41.24	40.57
<i>C3M</i>	48.79	47.13	44.78	43.79	42.21	43.02	41.44	41.19	39.92	39.62
$\ell_1 - G$	49.63	50.03	49.61	49.34	49.07	49.17	49.12	49.60	49.41	49.18
<i>LRR - G</i>	49.98	49.90	49.88	49.92	49.22	49.59	49.33	49.18	49.84	49.66
IQGD	15.27 ³	11.22 ²	13.03 ³	11.60 ³	11.24 ²	11.49 ²	11.51 ²	11.93 ³	11.07 ²	11.70 ²
<i>LLGC</i>										
<i>E3NN</i>	49.41	49.69	50.03	48.99	49.08	48.82	49.71	49.29	49.85	48.80
<i>M3NN</i>	49.90	49.71	49.41	49.38	49.59	48.90	49.55	49.03	48.47	48.85
<i>C3NN</i>	48.58	49.28	48.93	49.56	49.28	48.96	48.87	49.78	49.52	49.75
<i>E3M</i>	47.54	46.37	44.29	42.90	41.29	41.81	40.64	41.04	39.38	39.56
<i>M3M</i>	46.95	44.91	43.80	43.84	43.27	43.07	41.36	41.30	40.95	40.93
<i>C3M</i>	48.19	47.24	44.66	43.77	42.23	43.13	41.79	41.54	40.06	39.79
$\ell_1 - G$	49.65	49.90	49.68	49.89	49.73	49.81	50.27	49.79	49.48	49.97
<i>LRR - G</i>	49.78	49.90	49.72	49.90	49.69	49.89	49.99	49.89	49.73	49.89
IQGD	10.42 ²	11.99 ³	12.42 ²	11.32 ²	11.25 ³	11.49 ²	11.51 ²	11.65 ²	11.33 ³	11.70 ²
<i>SSC-GCN</i>										
<i>E3NN</i>	47.73	45.96	45.70	44.17	44.80	43.40	42.09	42.38	39.10	39.81
<i>M3NN</i>	48.52	49.16	44.35	44.41	43.76	44.54	42.02	41.97	40.57	40.08
<i>C3NN</i>	48.38	47.60	44.43	44.22	42.99	42.26	39.92	39.94	40.11	38.61
<i>E3M</i>	48.81	45.68	45.46	41.86	43.15	41.36	39.14	40.65	40.41	39.56
<i>M3M</i>	47.39	46.39	45.92	43.36	42.96	41.54	37.86	40.15	39.75	39.30
<i>C3M</i>	47.37	47.02	44.35	42.65	42.61	42.04	39.26	40.45	39.25	37.78
$\ell_1 - G$	47.18	45.45	43.07	41.24	40.77	40.80	39.59	40.21	38.97	37.54
<i>LRR - G</i>	42.57	39.57	38.11	35.53	33.76	32.63	30.86	31.01	29.41	28.62
IQGD	5.12 ¹	5.02 ¹	4.79 ¹	5.07 ¹	4.75 ¹	4.88 ¹	4.41 ¹	4.42 ¹	4.50 ¹	4.70 ¹

and SSC-GCN-IQGD methods are the best methods and are significantly better than the rank in the 2nd methods in the three groups respectively.

Based on the above experimental results, it can be concluded that compared with the common used graph construction methods, the proposed IQGD method can construct a better graph for GSSL. At the same time, the proposed GSSL-IQGD is a general framework and the performance of different GSSL methods can be improved by embedding them into it.

Table 16 Data set D9 classification error rate (%)

Method	# Labeled samples: l				
	1×3	5×3	10×3	15×3	20×3
<i>Harmonic</i>					
<i>E3NN</i>	78.86	77.89	76.88	75.57	74.47
<i>M3NN</i>	74.07	74.36	72.77	72.27	70.37
<i>C3NN</i>	74.37	74.74	74.94	74.20	72.86
IQGD	61.27 ³	50.97 ³	45.73 ³	45.11 ³	43.27 ³
<i>LLGC</i>					
<i>E3NN</i>	77.15	73.66	63.99	62.02	55.68
<i>M3NN</i>	67.87	60.44	55.14	51.28	50.05
<i>C3NN</i>	68.50	62.10	64.09	63.88	56.87
IQGD	52.66 ²	47.50 ²	45.35 ²	44.75 ²	42.82 ²
<i>SSC-GCN</i>					
<i>E3NN</i>	79.20	79.20	79.22	79.23	79.24
<i>M3NN</i>	79.20	79.20	79.22	79.24	79.24
<i>C3NN</i>	79.20	79.18	79.14	79.29	79.17
IQGD	34.12 ¹	25.54 ¹	28.78 ¹	39.99 ¹	22.45 ¹

Table 17 Data set D10 classification error rate (%)

Method	# Labeled samples: l				
	1×10	5×10	10×10	15×10	20×10
<i>Harmonic</i>					
<i>E3NN</i>	81.08	58.87	50.43	45.62	42.17
<i>M3NN</i>	77.53	56.55	50.88	45.90	44.80
<i>C3NN</i>	81.15	60.15	57.11	52.14	49.59
IQGD	71.92 ²	47.92 ⁵	26.74 ⁴	25.24 ⁵	23.48 ⁵
<i>LLGC</i>					
<i>E3NN</i>	74.92	37.89 ³	23.09 ¹	17.88 ²	15.48 ²
<i>M3NN</i>	74.81 ³	41.70	27.38	21.17 ³	22.47
<i>C3NN</i>	79.46	37.84 ²	23.72 ²	16.39 ¹	12.39 ¹
IQGD	50.90 ¹	28.49 ¹	25.84 ³	24.69 ⁴	22.09 ³
<i>SSC-GCN</i>					
<i>E3NN</i>	82.66	61.10	51.10	43.33	38.35
<i>M3NN</i>	82.20	60.43	50.80	43.14	39.69
<i>C3NN</i>	81.03	61.79	47.35	40.73	36.83
IQGD	76.96 ⁵	53.74 ⁶	44.97 ⁶	38.60 ⁶	37.18 ⁷

4.3 Compared with GSSL based on dynamic graph construction methods

4.3.1 Date sets

In this section, the first 8 data sets⁵ shown in Table 5 are used in the experiment. The ratio of the labeled samples and the division of labeled and unlabeled samples are the same as the experiment in Sect. 4.2.

⁵ Because of the high time complexity of the comparison method. In this section, we did not carry out the comparison experiment on the last two data sets.

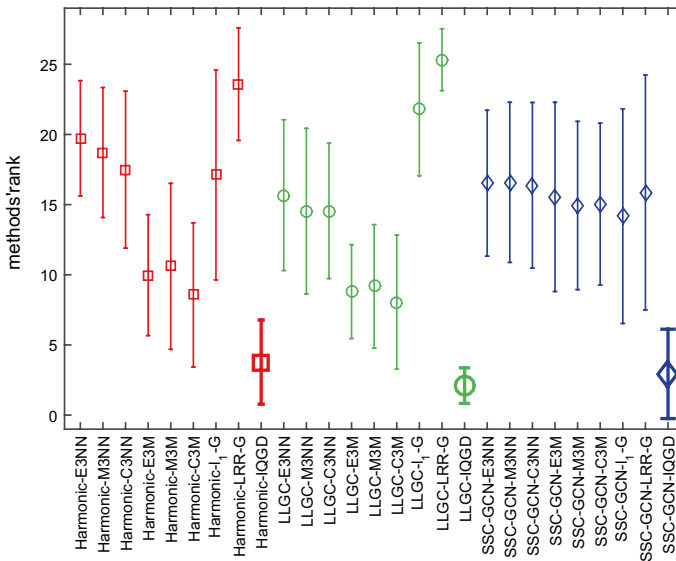


Fig. 5 Mean and standard deviation of rankings of 27 methods on the first 8 data sets

4.3.2 Comparison methods and experimental setting

In this experiment, 3 different GSSL methods based on dynamic graph construction methods are selected for comparison. The 3 methods and their settings are described as follows.

The STSSL-S³R (Semi-Supervised Sparse Representation) and the STSSL-S²LRR (Semi-Supervised Low Rank Representation) are two specific methods based on the framework proposed in literature (Li et al. 2015). In the STSSL-S³R (STSSL-S²LRR) method, the semi-supervised sparse (low rank) representation based graph construction and the Harmonic likewise label inference are integrated into a unified optimization framework. The alternating minimization algorithm is used to solve the optimization problem. In the process of model solving, the graph constructed by semi-supervised sparse (low rank) representation is updated dynamically. In this experiment, the model hyper-parameters [γ and λ , see formula (13) in literature (Li et al. 2015)] and the optimization hyper-parameters [ϵ and ρ_1 see Algorithm 1 in the literature (Li et al. 2015)] are set to the default values in author's code⁶.

The MGR-GGMC (Multiple Graph Regularized graph transduction via Greedy Gradient Max-Cut) method was proposed in literature (Xiu et al. 2018). In this method, the weighted sum of multiple graph smoothness terms and multiple greedy gradient max-cut based label inference are integrated into a unified optimization framework. In the process of model solving, the weights of multiple graph smoothness terms and the multiple label inference results are updated alternatively. In this method, the graph construction process essentially is a dynamic weighted fusion of multiple base graphs. And these base graphs are constructed by common used methods, for example the k NN graph. In this experiment, the hyper-parameters are set to the recommended values in the article, i.e. $\mu = 0.99$ and $\beta = 0.1$ [see Sect. 3.2 in the literature (Xiu et al. 2018)]. At the same time, the multiple base

⁶ <http://www.pris.net.cn/introduction/teacher/lichunguang>

Table 18 Data sets D1- D4 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>D1</i>										
STSSL-S ³ R	42.72	42.31	41.46	37.01	42.41	37.94	32.32	30.22	28.74	29.85
STSSL-S ² LRR	37.48	36.60	30.83	35.56	31.99	29.50	30.80	28.41	29.41	30.59
MG-GGMC	45.03	40.75	43.13	38.54	40.71	41.13	35.80	36.23	43.85	42.89
SLLI-IQGD	8.98	10.34	9.03	10.14	9.79	8.65	5.00	9.42	5.70	5.26
<i>D2</i>										
STSSL-S ³ R	55.03	49.48	42.22	36.94	42.02	42.53	30.24	23.01	27.56	30.69
STSSL-S ² LRR	58.46	54.39	45.26	48.18	40.89	40.84	48.11	35.71	44.75	37.80
MG-GGMC	62.29	63.93	59.42	59.18	55.48	62.11	56.28	56.07	57.94	59.31
SLLI-IQGD	40.11	38.73	34.44	36.18	32.80	35.42	34.09	31.72	32.19	33.02
<i>D3</i>										
STSSL-S ³ R	66.88	62.67	55.49	49.36	48.61	52.12	49.08	45.96	48.07	44.55
STSSL-S ² LRR	58.56	65.63	64.95	64.36	63.78	62.53	64.29	64.25	64.64	64.40
MG-GGMC	71.73	75.68	73.43	71.63	71.44	71.82	71.84	71.71	73.65	72.57
SLLI-IQGD	60.29	55.19	53.24	47.28	49.01	46.92	43.37	43.11	42.92	39.69
<i>D4</i>										
STSSL-S ³ R	46.32	46.33	46.29	46.31	46.27	46.29	46.30	46.32	46.28	46.30
STSSL-S ² LRR	40.28	44.54	42.58	37.74	37.72	35.26	35.20	34.98	34.81	32.11
MG-GGMC	45.27	47.09	47.83	48.74	47.79	48.01	47.60	50.19	49.27	49.85
SLLI-IQGD	37.73	36.11	34.14	34.03	33.50	33.48	33.33	33.18	33.53	33.31

graphs used in MGR-GGMC method are same as the 8 different graphs used in Sect. 4.2, see Table 6 for details.

At last, the SLLI-IQGD method described in formula (30), one of the simplest methods in the proposed framework, is used for comparison. In this experiment, the hyper-parameters of SLLI-IQGD method are set to $\gamma_{\text{fit}} = \gamma_{\text{smooth}} = \gamma_{\text{ml}} = \gamma_{\text{cl}} = 0.5$, which is the same as the setting in Sect. 4.1. At the same time, the multiple clustering results used in the SLLI-IQGD method are those used in the Harmonic-IQGD method in section 4.2 (see Table 7).

4.3.3 Experimental results and analysis

Tables 18 and 19 show the classification error of the 3 comparison methods and the proposed SLLI-IQGD method on the D1-D4 data sets and the D5-D8 data sets respectively. For each comparison experiment (given a data set and a labeled ratio), the bold marks represent the best results of the four methods.

The following conclusions can be drawn from Tables 18 and 19.

1. On the 6 data sets (D1 and D4–D8), the performance of the proposed SLLI-IQGD method significantly outperforms that of the other 3 comparison methods at 10 different ratios of labeled sample.

Table 19 Data sets D5–D8 classification error rate (%)

Method	Ratio of labeled samples: l/n (%)									
	1	2	3	4	5	6	7	8	9	10
<i>D5</i>										
STSSL-S ³ R	36.35	30.78	27.71	23.51	24.62	22.55	21.49	19.27	18.68	19.39
STSSL-S ² LRR	79.88	69.42	61.15	54.20	53.14	48.29	49.72	43.39	44.24	38.56
MG-GGMC	63.81	50.25	52.47	42.00	48.26	48.56	48.59	49.00	46.64	55.60
SLLI-IQGD	31.04	24.67	21.46	20.29	20.63	19.36	18.59	18.68	18.11	17.53
<i>D6</i>										
STSSL-S ³ R	63.55	62.19	58.96	57.60	56.23	52.97	51.97	52.20	51.23	49.43
STSSL-S ² LRR	62.55	65.17	63.74	63.52	63.18	65.01	65.08	63.73	65.15	64.97
MG-GGMC	47.14	51.59	50.88	54.47	55.33	54.79	54.80	56.06	61.58	54.87
SLLI-IQGD	21.87	19.99	19.81	19.59	18.47	18.84	18.34	18.12	18.70	18.24
<i>D7</i>										
STSSL-S ³ R	50.15	50.37	50.16	49.83	50.18	50.31	50.03	49.85	49.93	50.25
STSSL-S ² LRR	49.96	50.00	49.90	50.06	49.89	50.02	50.02	49.99	50.03	50.22
MG-GGMC	49.89	49.71	48.67	48.43	49.63	49.52	49.51	50.45	50.26	50.77
SLLI-IQGD	26.00	23.55	25.48	25.27	25.02	23.64	23.67	23.75	24.02	23.34
<i>D8</i>										
STSSL-S ³ R	49.57	50.01	49.54	49.25	49.28	49.23	49.27	49.49	49.55	49.08
STSSL-S ² LRR	49.72	49.90	49.91	49.95	50.79	49.86	50.28	49.50	50.07	49.84
MG-GGMC	49.91	49.97	48.19	48.76	47.77	49.00	48.80	50.07	49.50	49.73
SLLI-IQGD	12.20	11.74	12.70	12.15	11.09	11.49	11.51	11.91	11.29	11.71

- On the D2 data set, the performance of the proposed SLLI-IQGD method outperforms that of the other 3 comparison methods at the first 6 labeled ratios and its performance is better than the STSSL-S²LRR and MGR-GGMC methods at the last 4 labeled ratios.
- On the D3 data set, the performance of the proposed SLLI-IQGD method outperforms that of the other 3 comparison methods at the 8 different labeled ratios (except for 1% and 5%). At labeled ratio 1%, the STSSL-S²LRR method obtains a better result than the proposed SLLI-IQGD method. And at labeled ratio 5%, the performance of the STSSL-S³R method is better than the proposed SLLI-IQGD method.
- To sum up, the proposed SLLI-IQGD method is significantly superior to the 3 comparison methods in most cases. The advantages of the proposed SLLI-IQGD method are more significant when the labeled ratio is small.

Although the STSSL-S³R (STSSL-S²LRR) method realizes the mutual guidance between sparse (low rank) representation based graph construction and graph based label inference. But they still make a strong assumption about the data distribution, that is, assuming that the data distribution satisfies the subspace structure. These two methods alternately optimize the label inference result and sparse (low-rank) representation coefficient matrix. When the assumption is not satisfied, it makes the model go further and further in the wrong direction. Therefore. In this experiment, these two methods are difficult to achieve a good result. Different from them, the SLLI-IQGD uses different clustering

methods to mine a variety of possible data distribution structures, which weakens the dependence on data distribution assumptions. At the same time, by designing a reasonable learning scheme of the weights of multiple clustering results, the dynamic improvement of the graph quality is achieved, so a better result can be obtained.

The MGR-GGMC method integrates the smoothing terms on multiple base graphs and multiple greedy gradient max-cut based label inference into a unified optimization problem. Compared with the proposed SLLI-IQGD method, there are three differences. First, the base graphs used in MGR-GGMC are obtained by conventional graph construction methods, while the base graphs in the proposal are obtained by different clustering methods. Compared with the conventional graph construction methods, the clustering method can mine the potential data distribution structure more effectively. Second. In the MGR-GGMC method, label inference is performed on every base graph. In the proposed method, it is carried out on the fused graph. In this way, the graph obtained by fusion of multiple clustering results can guide graph-based label propagation more effectively. Third. In the MGR-GGMC method, the the learning process of weights of multiple base graph is only guild by the intermediate label inference result. In the proposed method, the the learning process of weights of multiple clustering results is jointly guild by three factors: the intermediate label inference result, the “must link” supervision information and the “cannot link” supervision information. So the proposed method can learn better weights so as to construct a better graph. Therefore, compared with the MGR-GGMC method, the method in this paper can achieve a better result.

Through the experiments in this section, the following conclusions can be drawn. Compared with the existing GSSL methods. Including adopting the dynamic graph construction method based on data representation and multiple graphs fusion, the method proposed in this paper can construct a better graph so as to obtain a better performance.

5 Conclusions and future work

This paper proposes a general framework named GSSL-IQGD for improving the quality of the graph in GSSL and the performance of existing GSSL methods. In this framework, the two processes, graph construction based on the weighted fusion of multiple clustering results and label inference are integrated into a unified optimization problem. In the model solving, these two processes are alternately executed and guided by each other, which realizes the dynamic improvement of the quality of the graph and the result of label inference. In the experiment, firstly, three toy examples illustrate the working mechanism of the method. Then, a large number of comparative experiments verify the effectiveness of the proposed IQGD for improving the quality of the graph in GSSL. Meanwhile, these experimental results also indicate that the proposed GSSL-IQGD method is a general method, i.e. it can be used to improve the performance of existing different GSSL methods. Finally, the advantage of the proposed GSSL-IQGD method compared with other existing GSSL methods based on dynamic graph construction is verified through a large number of comparative experiments.

The method proposed in this paper is a general framework. In the experiment, three classic GSSL methods were chosen to be embedded into the framework and the desired performance gains were achieved. In the future, embedding other classic GSSL methods into the framework proposed in this paper to improve the performance of these methods is a worthwhile research work. In addition, similar to most GSSL methods, the method

proposed in this paper belongs to the transductive learning method. How to extend it to inductive learning is also a meaningful research direction.

Acknowledgements This work is supported by the National Key Research and Development Program of China (under Grant 2020AAA0106100) and the National Natural Science Foundation of China (Nos. 61876103, 61772323).

References

- Bai, L., Liang, J. Y., & Guo, Y. (2018). An ensemble clusterer of multiple fuzzy k -means clusterings to recognize arbitrarily shaped clusters. *IEEE Transactions on Fuzzy Systems*, 26(6), 3524–3533.
- Basu, S., Banerjee, A., & Mooney, R. J. (2002). Semi-supervised clustering by seeding. In: *Proceedings of the 19th International Conference on Machine Learning* (pp. 27–34). Sydney, Australia.
- Belkin, M., & Niyogi, P. (2008). Towards a theoretical foundation for laplacian-based manifold methods. *Journal of Computer and System Sciences*, 74(8), 1289–1308.
- Belkin, M., Niyogi, P., & Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7, 2399–2434.
- Berthelot, D., Carlini, N., Goodfellow, I. J., Papernot, N., Oliver, A., & Raffel, C. (2019). Mixmatch: a holistic approach to semi-supervised learning. In: B. C. Vancouver (Ed.), *Advances in neural information processing systems 32. Annual conference on neural information processing systems* (pp. 5050–5060). Canada.
- Berton, L. & de Andrade Lopes, A. (2014). Graph construction based on labeled instances for semi-supervised learning. In: *Proceedings of the 22nd international conference on pattern recognition* (pp. 2477–2482). Stockholm, Sweden.
- Berton, L., de Paulo Faleiros, T., Valejo, A., Valverde-Rebaza, J. C., & de Andrade Lopes, A. (2017). RGCLI: Robust graph that considers labeled instances for semi-supervised learning, *Neurocomputing* 226, 238–248.
- Blum, A. & Mitchell, T. M. (1998). Combining labeled and unlabeled data with co-training. In: *Proceedings of the 11th Annual Conference on Computational Learning Theory* (pp. 92–100). Madison, Wisconsin, USA.
- Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge: Cambridge University Press.
- Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. Cambridge: The MIT Press.
- Chapelle, O., Sindhwani, V., & Keerthi, S. S. (2008). Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9, 203–233.
- Chen, D., Wang, W., Gao, W. & Zhou, Z. H. (2018). Tri-net for semi-supervised deep learning. In: *Proceedings of the 27th international joint conference on artificial intelligence* (pp. 2014–2020). Stockholm, Sweden.
- Cheng, B., Yang, J., Yan, S., Fu, Y., & Huang, T. S. (2010). Learning with ℓ_1 -graph for image analysis. *IEEE Transactions on Image Processing*, 19(4), 858–866.
- Cozman, F. G. & Cohen, I. (2002). Unlabeled data can degrade classification performance of generative classifiers. In: *Proceedings of the 15th international florida artificial intelligence society conference* (pp. 327–331). Pensacola, FL.
- Dai, Z., Yang, Z., Yang, F., Cohen, W. W., & Salakhutdinov, R. (2017). Good semi-supervised learning that requires a bad GAN. *Advances in neural information processing systems 30*. In: *Annual conference on neural information processing systems* (pp. 6510–6520). Long Beach, CA, USA.
- De Sousa, C. A. R., Rezende, S. O. & Batista, G. E. (2013). Influence of graph construction on semi-supervised learning. In: *Proceedings of joint european conference on machine learning and knowledge discovery in databases* (pp. 160–175). Springer.
- Ester, M., Kriegl, H., Sander, J. & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Proceedings of the 2nd international conference on knowledge discovery and data mining* (pp. 226–231). Portland, Oregon, USA.
- Fred, A. L. N., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6), 835–850.
- Ghazvininejad, M., Mahdih, M., Rabiee, H. R., Roshan, P. K. & Rohban, M. H. (2011). Isograph: Neighbourhood graph construction based on geodesic distance for semi-supervised learning. In: *Proceedings of the 11th IEEE international conference on data mining* (pp. 191–200). Vancouver, BC, Canada.

- Huang, B. C. & Jebara, T. (2007). Loopy belief propagation for bipartite maximum weight b-matching. In: *Proceedings of the 11th international conference on artificial intelligence and statistics* (pp. 195–202). San Juan, Puerto Rico.
- Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8), 651–666.
- Jebara, T., Wang, J. & Chang, S. F. (2009). Graph construction and b-matching for semi-supervised learning. In: *Proceedings of the 26th International conference on machine learning* (pp. 441–448). Montreal, Quebec, Canada.
- Jiang, B., Zhang, Z., Lin, D., Tang, J. & Luo, B. (2019). Semi-supervised learning with graph learning-convolutional networks. In: *Proceedings of the 32nd IEEE conference on computer vision and pattern recognition* (pp. 11313–11320). Long Beach, CA, USA.
- Joachims, T. (1999). Transductive inference for text classification using support vector machines. In: *Proceedings of the 16th international conference on machine learning* (pp. 200–209). Bled, Slovenia.
- Karlen, M., Weston, J., Erkan, A. & Collobert, R. (2008). Large scale manifold transduction. In: *Proceedings of the 25th International conference on machine learning* (pp. 448–455). Helsinki, Finland.
- Kingma, D. P., Mohamed, S., Rezende, D. J., & Welling, M. (2014). Semi-supervised learning with deep generative models. *Advances in Neural Information Processing Systems* 27. In: *Annual conference on neural information processing systems* (pp. 3581–3589). Montreal, Quebec, Canada.
- Kipf, T. N. & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In: *The 5th international conference on learning representations*, Toulon, France.
- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: *Proceedings of the 30th international conference on machine learning*, Atlanta, GA, USA.
- Li, Q., Han, Z. & Wu, X. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In: *Proceedings of the 32nd aaai conference on artificial intelligence* (pp.3538–3545). New Orleans, Louisiana, USA.
- Li, C., Lin, Z., Zhang, H. & Guo, J. (2015). Learning semi-supervised representation towards a unified optimization framework for semi-supervised learning. In: *Proceedings of the 15th IEEE international conference on computer vision* (pp. 2767–2775). Santiago, Chile.
- Li, Y. F., Wang, S. B. & Zhou, Z. H. (2016). Graph quality judgement: A large margin expedition. In: *Proceedings of the 25th International joint conference on artificial intelligence* (pp. 1725–1731). New York, NY, USA.
- Li, C., Xu, T., Zhu, J., & Zhang, B. (2017). Triple generative adversarial nets. *Advances in neural information processing systems* 30. In: *Annual conference on neural information processing systems* (pp. 4088–4098). Long Beach, CA, USA.
- Liu, G., Lin, Z., Yan, S., Sun, J., Yu, Y., & Ma, Y. (2013). Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 171–184.
- Li, Y. F., & Zhou, Z. H. (2015). Towards making unlabeled data never hurt. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(1), 175–188.
- Michalski, R. S., & Anderson, J. R. (1984). *Machine learning—an artificial intelligence approach, symbolic computation*. Berlin: Springer.
- Park, S., Park, J., Shin, S. & Moon, I. (2018). Adversarial dropout for supervised and semi-supervised learning. In: *Proceedings of the 32nd AAAI conference on artificial intelligence* (pp. 3917–3924). New Orleans, Louisiana, USA.
- Rasmus, A., Berglund, M., Honkala, M., Valpola, H., & Raiko, T. (2015). Semi-supervised learning with ladder networks. *Advances in neural information processing systems* 28. In: *Annual conference on neural information processing systems* (pp. 3546–3554). Montreal, Quebec, Canada.
- Rustamov, R. M. & Klosowski, J. T. (2018). Interpretable graph-gated semi-supervised learning via flows. In: *Proceedings of the 22nd AAAI conference on artificial intelligence* (pp. 3976–3983). New Orleans, Louisiana, USA.
- Sechidis, K., & Brown, G. (2018). Simple strategies for semi-supervised feature selection. *Machine Learning*, 107(2), 357–395.
- Shahshahani, B. M., & Landgrebe, D. A. (1994). The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hughes phenomenon. *IEEE Transactions on Geoscience and Remote Sensing*, 32(5), 1087–1095.
- Sheikhpour, R., Sarram, M. A., Gharaghani, S., & Chahooki, M. A. Z. (2017). A survey on semi-supervised feature selection methods. *Pattern Recognition*, 64, 141–158.
- Subramanya, A., & Bilmes, J. (2011). Semi-supervised learning with measure propagation. *Journal of Machine Learning Research*, 12, 3311–3370.

- Tenenbaum, J. B., De Silva, V., & Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, *290*(5500), 2319–2323.
- Triguero, I., García, S., & Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: Taxonomy, software and empirical study. *Knowledge and Information Systems*, *42*(2), 245–284.
- Van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Maching Learning*, *109*(2), 373–440.
- Wagstaff, K., Cardie, C., Rogers, S. & Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In: *proceedings of the 18th international conference on machine learning* (pp. 577–584). Williams College, Williamstown, MA, USA..
- Wang, W. & Zhou, Z. H. (2010). A new analysis of co-training. In: *Proceedings of the 27th international conference on machine learning* (pp. 1135–1142). Haifa, Israel.
- Wang, F., & Zhang, C. S. (2008). Label propagation through linear neighborhoods. *IEEE Transactions on Knowledge and Data Engineering*, *20*(1), 55–67.
- Wei, T., Guo, L., Li, Y., & Gao, W. (2018). Learning safe multi-label prediction for weakly labeled data. *Machine Learning*, *107*(4), 703–725.
- Weston, J., Ratle, F. & Collobert, R. (2008). Deep learning via semi-supervised embedding. In: *Proceedings of the 25th international conference on machine learning* (pp. 1168–1175). Helsinki, Finland.
- Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2009). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.
- Xiu, Y., Shen, W., Wang, Z., Liu, S., & Wang, J. (2018). Multiple graph regularized graph transduction via greedy gradient max-cut. *Information Sciences*, *423*, 187–199.
- Yan, S. & Wang, H. (2009). Semi-supervised learning by sparse representation. In: *Proceedings of the SIAM international conference on data mining* (pp. 792–801). Sparks, Nevada, USA.
- Yuille, A. L., & Rangarajan, A. (2003). The concave-convex procedure. *Neural Computation*, *15*(4), 915–936.
- Zeng, H., & Cheung, Y. M. (2012). Semi-supervised maximum margin clustering with pairwise constraints. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 926–939.
- Zhang, D., Zhou, Z. H. & Chen, S. (2007). Semi-supervised dimensionality reduction. In: *Proceedings of the 7th SIAM International conference on data mining* (pp. 629–634). Minneapolis, Minnesota, USA.
- Zhao, X. W., Liang, J. Y., & Dang, C. Y. (2017). Clustering ensemble selection for categorical data based on internal validity indices. *Pattern Recognition*, *69*, 150–168.
- Zhou, Z. H. & Li, M. (2005). Semi-supervised regression with co-training. In: *Proceedings of the 19th international joint conference on artificial intelligence* (pp. 908–916). Edinburgh, Scotland, UK.
- Zhou, D., Bousquet, O., Lal, T. N., Weston, J., & Schölkopf, B. (2003). Learning with local and global consistency. *Advances in neural information processing systems* 16. In: *Annual Conference on neural information processing systems* (pp. 321–328). British Columbia, Canada.
- Zhu, X., Ghahramani, Z. & Lafferty, J. D. (2003). Semi-supervised learning using gaussian fields and harmonic functions. In: *Proceedings of the 20th International conference on machine learning* (pp. 912–919). Washington, DC, USA.
- Zhuang, L., Gao, H., Huang, J. & Yu, N. (2011). Semi-supervised classification via low rank graph. In: *Proceedings of the 6th International conference on image and graphics* (pp. 511–516). Hefei, Anhui, China.
- Zhuang, L., Zhou, Z., Gao, S., Yin, J., Lin, Z., & Ma, Y. (2017). Label information guided graph construction for semi-supervised learning. *IEEE Transactions on Image Processing*, *26*(9), 4182–4192.
- Zhu, X., & Goldberg, A. B. (2009). *Introduction to semi-supervised learning, synthesis lectures on artificial intelligence and machine learning*. San Rafael: Morgan & Claypool Publishers.