Information Sciences 560 (2021) 92–106

Contents lists available at ScienceDirect

Information Sciences

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

Semi-supervised classification by graph *p*-Laplacian convolutional networks

Sichao Fu [a,b], Weifeng Liu [b,*], Kai Zhang [c], Yicong Zhou [d], Dapeng Tao [e]

[a] School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China
[b] College of Control Science and Engineering, China University of Petroleum (East China), Qingdao 266580, China
[c] School of Petroleum Engineering, China University of Petroleum (East China), Qingdao 266580, China
[d] Faculty of Science and Technology, University of Macau, Macau 999078, China
[e] School of Information Science and Engineering, Yunnan University, Kunming 650091, China

ARTICLE INFO

Article history:
Received 25 May 2020
Received in revised form 20 January 2021
Accepted 25 January 2021
Available online 5 February 2021

Keywords:
Graph convolutional networks
Manifold learning
p-Laplacian
Semi-supervised classification

ABSTRACT

The graph convolutional networks (GCN) generalizes convolution neural networks into the graph with an arbitrary topology structure. Since the geodesic function in the null space of the graph Laplacian matrix is constant, graph Laplacian fails to preserve the local topology structure information between samples properly. GCN thus cannot learn better representative sample features by the convolution operation of the graph Laplacian based structure information and input sample information. To address this issue, this paper exploits the manifold structure information of data by the graph *p*-Laplacian matrix and proposes the graph *p*-Laplacian convolutional networks (GpLCN). As the graph *p*-Laplacian matrix is a generalization of the graph Laplacian matrix, GpLCN can extract more abundant sample features and improves the classification performance utilizing graph *p*-Laplacian to preserve the rich intrinsic data manifold structure information. Moreover, after simplifying and deducing the formula of the one-order spectral graph p-Laplacian convolution, we introduce a new layer-wise propagation rule based on the one-order approximation. Extensive experiment results on the Citeseer, Cora and Pubmed database demonstrate that our GpLCN outperforms GCN.

1. Introduction

With the advent of the information age and the growth of unstructured data, higher data dimensions and faster data updates are more prominent. How to extract effective and reasonable data information from massive datasets is an urgent problem to be solved. Therefore, the data information representation methods [1], especially the manifold structure information of data, has become an important research topic for machine learning. The goal of manifold learning [2] is to discover low-dimensional manifold structure from high-dimensional sampled data, i.e. it learns low-dimensional manifold in high-dimensional space, and then finds corresponding embedding mapping relationships to achieve data visualization. The related algorithms of manifold learning (ML) have an important research significance in theory and applications including machine learning [3], data mining [4], and computer vision [5].

* Corresponding author.
E-mail address: liuwf@upc.edu.cn (W. Liu).

https://doi.org/10.1016/j.ins.2021.01.075
0020-0255/

In recent years, many prominent ML algorithms have been proposed and achieved great performance in the research hotspots of dimensionality reduction [6], clustering [7] and semi-supervised learning [8].

In the dimensionality reduction methods of ML, it maps the samples globally into a low-dimensional space by constructing the local neighborhood structure of the sample on the manifold. Tenenbaum et al. [9] utilized the geodesic distance in differential geometry to measure the pairwise distance between data. Roweis et al. [10] assumed that the low-dimensional manifold of the data sets and their mappings in the high dimensional observation space is locally linear. Then the dimensionality reduction is achieved by maintaining the fixed local linear relationships. Belkin et al. [11] described a manifold with an undirected weight graph, then it found a low-dimensional representation by using graph embedding.

The manifold clustering methods are mainly to find the potential manifold structure from the global or local correlation between data. The data will be clustered according to the data similarity for the manifold structure. Souvenir et al. [12] used the Isometric Feature Mapping (Isomap) method to calculate the distance between data, and solved the clustering problem through the expectation–maximization (EM) iterative method. In [13], the similarity of the data is measured by the distance how each sample moves to other samples on the manifold. Ye et al. [14] projected the data into a low-dimensional manifold that maximizes data separability, and then used the Mahalanobis distance as the distance measure in low-dimensional space.

The semi-supervised learning algorithms [15,16] incorporate label information into the data manifold hypothesis, i.e. all samples are distributed on a low-dimensional manifold structure, and the geodesic distance for the sample with the same label is small. Liu et al. [17] used Hessian to preserve the manifold structure of data, which aimed to solve the poor extrapolating power of Laplacian. Cai et al. [18] took full advantage of the information of labeled and unlabeled samples to build a graph, which aimed to express a discrete approximation of the data manifold structure. The goal of this model is to learn a smooth discriminant function using the data manifold information. Kokiopoulou et al. [19] exploited the sparse geometry to express transformed manifold information of data.

Recently, deep learning has also received widespread concerns in the graph data field. Kipf et al. [20] proposed a generalization method of the convolutional neural network (CNN) on graph data, called graph convolutional networks (GCN). GCN can deal with arbitrary spatial structure data. It uses the graph Laplacian method to obtain the structure information of the graph data. GCN can extract representative data feature via automatically learning feature information and structure information of the graph data simultaneously. It is currently the best choice for the graph data learning task. Besides, the learning and computation complexity of the model has linear relationships. Different from GCN, two-order GCN [21] simultaneously utilized the direct and indirect neighbors relationships between nodes by using the two-order approximation of spectral graph Laplacian convolution. HyperGCN [22] generalized the simple graph convolution operation of GCN to the hypergraph domain by using hypergraph Laplacian to capture more complex or beyond pairwise relationships between nodes. Compared with the first-order derivatives of Laplacian, HesGCN [23] can acquire more accurate data features by the existence of the Hessian matrix's two-order derivatives to get richer structure information in the original data with the complex structures. Graph attention network (GAT) [24] encoded the hidden representations of each node in the graph by introducing the self-attention mechanism to attend over its neighbors. Liu et al. [25] proposed a novel weakly supervised multi-label image classification framework based on GCN with learning the semantic label co-occurrence in an image.

However, the null space of the graph Laplacian is not rich. Thus, the extracted manifold structure relationship fails to have the representative capability of the graph data. Due to the lack of rich structure information, GCN cannot learn sufficient features from input samples (The learning process of GCN can be regarded as the fusion process of the input sample information and the structure information based on graph Laplacian). This reduces the classification performance of GCN as well. To better show the difference of graph Laplacian, graph p-Laplacian (p = 2) and graph p-Laplacian ($p \neq 2$) in preserving local geometry of data, Fig. 1 plots the experiment results of graph Laplacian, graph p-Laplacian (p = 2) and graph p-Laplacian ($p \neq 2$) based semi-supervised regression methods. The graph Laplacian has a bias towards the constant function and the extrapolation function remains unchanged along the spiral for unseen data, which cannot fit the data properly. When parameter $p = 2$, graph p-Laplacian has a similar performance with graph Laplacian. Due to the variation smoothness of graph p-Laplacian's ($p \neq 2$) extrapolation function with the geodesic distance, it can fit the data properly and extrapolates perfectly to unseen data.

In this paper, we exploit the graph p-Laplacian method to express the manifold structure of the data samples. The graph p-Laplacian [26,27] is a nonlinear generalization of the graph Laplacian. It provides the most compelling theoretical evidence to better express the local structure information. In other words, graph p-Laplacian obtains tighter isoperimetric inequality, thus the upper and lower bounds on the second eigenvalue approximate the optimal Cheeger cut value well [28]. Compared with graph Laplacian, graph p-Laplacian can fit the data properly and extrapolates smoothly to unseen data with the geodesic distance, i.e. graph p-Laplacian has great superiority in the local manifold structure of data preserving. Besides, we apply the graph p-Laplacian matrix to spectral convolutions in the graph field and obtain a novel form of the spectral graph Laplacian convolutions, i.e. spectral graph p-Laplacian convolutions. Finally, we introduce a layer-wise rule representation form by optimizing the one-order approximation of the spectral graph p-Laplacian convolutions. Specially, we build a model by this layer-wise rule formula, i.e. graph p-Laplacian convolutional networks (GpLCN). It can learn rich sample features by integrating graph p-Laplacian into feature information. We test our model on Citation network datasets for node classification. Extensive experimental results prove that GpLCN obtains a higher recognition accuracy in comparison to GCN.

The main contributions of this paper can be summarized as follows:

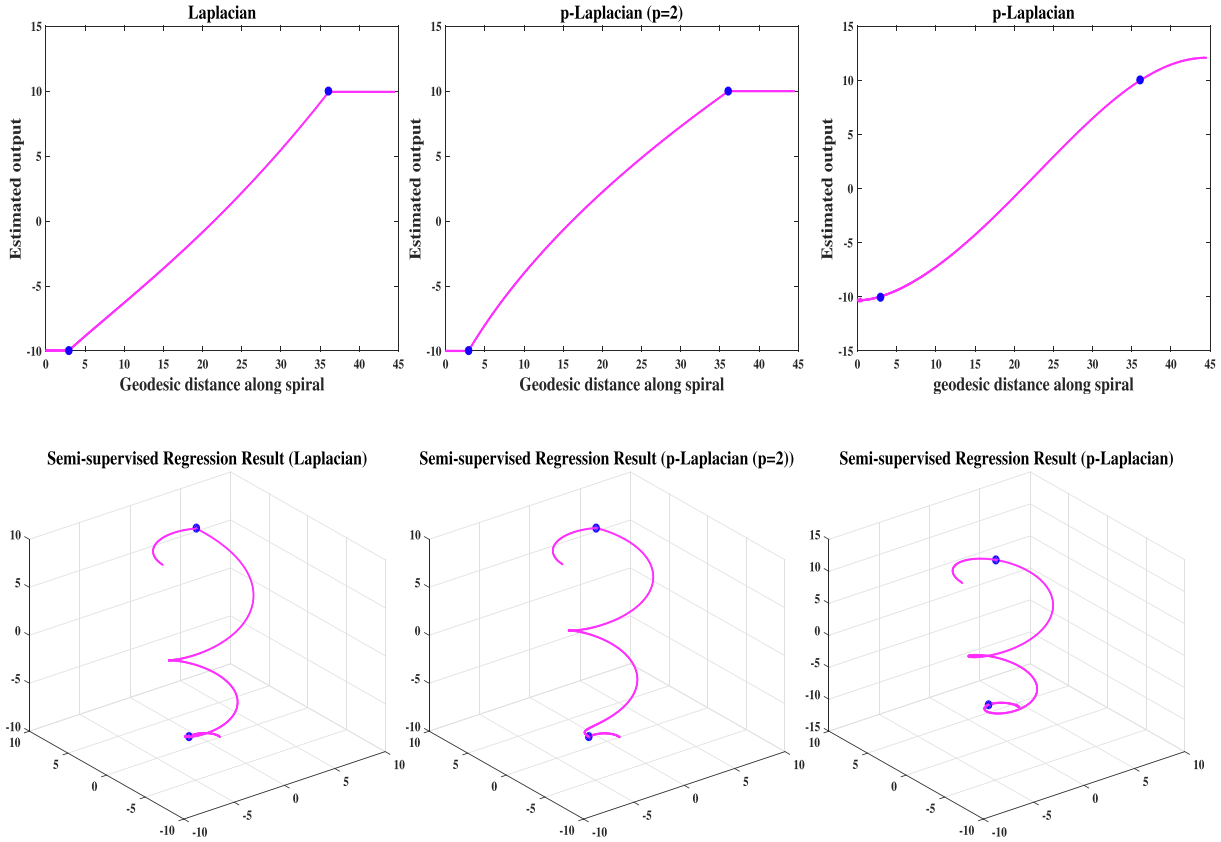

Fig. 1. The difference of semi-supervised regression methods for fitting two points on the one-dimensional spiral by separately utilizing graph Laplacian, graph p-Laplacian (p = 2) and graph p-Laplacian ($p \neq 2$) to preserve the local geometry structures of the data manifold.

1) We generalize the spectral graph Laplacian convolutions into the spectral convolutions on graph p-Laplacian. It can utilize the graph p-Laplacian to better preserve the manifold structure information of data.
2) Simplifying and deducing the one-order approximation of the spectral graph *p*-Laplacian convolutions, we propose an effective convolution layer rule.
3) Based on the proposed convolution layer rule, we design a two-layer graph *p*-Laplacian convolutional networks (GpLCN).
4) Extensive experiment results on public datasets show that the proposed GpLCN outperforms several semi-supervised classification methods such as GCN, HyperGCN.

The rest of this paper is organized as follows. Section 2 briefly describes related works on graph *p*-Laplacian theory and graph convolutional network. Section 3 presents the definition of spectral graph p-Laplacian convolutions and GpLCN in detail. Experimental results and analysis of GpLCN are shown in Section 4. Finally, Section 5 concludes this paper.

2. Related work

Our GpLCN is inspired by graph *p*-Laplacian theory and GCN [20]. We give a brief description of the related works in this section.

2.1. Graph p-Laplacian theory

Tan et al. [29] proposed the definition of the standard graph Laplacian operator Δ_2 through the form of inner product, i.e.

$$\langle f, \Delta_2 f \rangle_i = \frac{1}{2} \sum_{i,j=1}^{n} M_{ij} (f_i - f_j)^2 \tag{1}$$

First, we suppose an undirected and weighted graph $G = \{V, E, M\}$, V is the vertex set of the graph, E is a set of edges, M is the edge weight set of the graph. M_{ij} is the weight of any edge. f is the eigenvector of the graph Laplace operator. Then, we set a parameter $p \geqslant 1$ and obtain the inner product form of the graph p-Laplacian operator Δ_p, i.e.

$$\langle f, \Delta_p f \rangle_i = \frac{1}{2}\sum_{i,j=1}^{n} M_{ij}(f_i - f_j)^p \tag{2}$$

Finally, we can get the unnormalized and normalized matrix forms of the graph p-Laplacian operator Δ_p, i.e.

$$\left(\Delta_p^{(u)}\right)_i = \sum_{j\in V} M_{ij}\varphi_p(f_i - f_j) \tag{3}$$

$$\left(\Delta_p^{(n)}\right)_i = \frac{1}{d_i}\sum_{j\in V} M_{ij}\varphi_p(f_i - f_j) \tag{4}$$

Here, $d_i = \sum_{j=1}^{n} M_{ij}$ is the degree function of the graph. $\varphi_p(x) = |x|^{p-1}sign(x)$. When the parameter $p = 2$, $\varphi_p(x) = x$, and the graph p-Laplacian reverts to the standard graph Laplacian. In addition, Liu et al. [30] has been proved the superiority of graph p-Laplacian in manifold structure exploiting compared to graph Laplacian. The computational process of graph p-Laplacian is divided into four parts [30]: First, we compute the adjacency matrix A and normalized graph Laplacian L. Second, we get the eigenvector decomposition of L. Third, we get an approximation for full eigenvectors of graph p-Laplacian by the gradient descend optimization method [31]. Finally, we obtain the graph p-Laplacian matrix through above steps.

Due to the superiority of the graph p-Laplacian in a local structure preserving compared to the graph Laplacian, many authors have done some attempts. Liu et al. [30] used p-Laplacian to preserve the manifold structure information of data. They combined pLapR with support vector machines (SVM) and kernel least squares (KLS) for scene recognition. Liu et al. [32] popularized Laplacian regularized sparse coding to p-Laplacian regularized sparse coding. The method can get the rich manifold structure relationships of samples. It combined SVM with p-Laplacian regularized sparse coding (pLSC) for human activity recognition. Luo et al. [31] proposed a method to get a natural global p-Laplacian embedding by full eigenvector analysis for multi-class clustering problems. The embedding space of p-Laplacian is achieved through a valid gradient descend optimization method. To avoid the staircase effect of the smooth regions for image denoising and better preserving edges' structure information, Zhang et al. [33] introduced an adaptive second-order partial differential equation based on the p-Laplacian equation to solve this problem. To acquire a fitting graph p-Laplacian, the selection of the parameter p is critical for preserving the manifold structure. Thus, Ma et al. [34] gave the ensemble p-Laplacian regularization KLS and SVM methods with an optimal fused graph for scene image recognition to better approximate the local geometry of the data. Bougleux et al. [35] utilized the p-Laplacian matrix on the directed graphs to construct a structure-preserving filter for editing of 3D colored meshes. It can better preserve the intrinsic structures of the true data distribution.

2.2. Graph Convolutional Networks

The original definition of spectral graph convolution in Fourier domain is the multiplication of a signal X and a filter g_θ, i.e.

$$\begin{aligned} g_\theta(L)\bigstar X &= U\left(\left(U^T g_\theta\right)\odot\left(U^T X\right)\right) \\ &= U g_\theta(\wedge)U^T X \end{aligned} \tag{5}$$

Here, in this paper, it used the normalized graph Laplacian, i.e. $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}, D_{ii} = \sum_j A_{ij}$. A denotes the adjacency relationship matrix of graph data. U is the eigenvector matrix of L. $\wedge$ is the eigenvalue matrix of L. However, it failed to reflect non-localized in vertex domain and is not applicable for the large graph.

To solve the above-mentioned problems, Defferrard et al. [36] proposed the optimized definition of spectral convolutions on graph data, i.e.

$$g_\theta(L)\bigstar X = \sum_{k=0}^{k}\theta_k T_k\left(\widetilde{L}\right)X \tag{6}$$

Here, $\widetilde{L} = \frac{2}{\lambda_{max}}L - I_N$. λ_{max} represents the maximum eigenvalue of graph Laplacian L. The definition of Chebyshev is calculated through the recursive form, i.e. $T_0(X) = 1, T_1(X) = X, T_k(X) = 2XT_{k-1}(X) - T_{k-2}(X)$. Defferrard et al. [36] built a model through K-order localized spectral graph convolutions.

Kipf et al. [20] provided a method for processing graph-structure data and applied the convolutional neural networks for images in deep learning to graph data. The filter of the convolution networks is moved to the Fourier domain with the graph signal for processing. This model is based on the one-order polynomial of graph Laplacian on graph data ($K = 1$), i.e. graph convolutional networks (GCN). Finally, it uses an optimized layer-wise formula of GCN, i.e.

$$H^{(L+1)} = \sigma\left(\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}H^{(L)}W^{(L)}\right) \tag{7}$$

Here, $\widetilde{A} = A + I_N$is the adjacency relationship matrix including self-connections on graph data. A is the adjacency relationship matrix. $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$ is the degree matrix about $\widetilde{A}$. $W^{(L)}$ is the weight parameter of each-layer networks. $H^{(L)}$ is the sample feature information extracted by the last layer, $H^{(0)} = X$. $\sigma$is the activation function. The GCN model was built by the multiple convolution layers in the layer-wise form.

3. Graph *p*-Laplacian convolutional networks

Inspired by the definition of spectral convolution on graph data and the GCN model, we introduce a novel definition on the graph, i.e. spectral graph p-Laplacian convolution. Moreover, we introduce a new layer-wise propagation rule form via the one-order approximation of spectral graph p-Laplacian convolutions. Specifically, we build a new model by using the multiple layer-wise propagation formula, named graph p-Laplacian convolutional networks (GpLCN).

The GpLCN algorithm can be divided into three parts. The first part is the definition of the layer-wise propagation rule. In this part, we demonstrate that the concept of spectral graph p-Laplacian convolution can be derived from the definition of spectral convolution on graph data. Then we limit the order in the graph p-Laplacian polynomial and obtain a scalable layer-wise model. The second part is the convolution layer networks of GpLCN. We show the definition of the first and second convolution layers of GpLCN, and explain the amalgamation process of the structure information into the feature information. Finally, the most representative data features are used for classification. In this paper, we use the Softmax classifier. Fig. 2 shows the basic framework of the multi-layer GpLCN model, which can be built by stacking the multi-layer layer-wise convolution layer rule.

3.1. Spectral graph p-Laplacian convolution

GCN model uses a one-order approximation of the graph Laplacian to express the manifold structure information of data. In each convolution layer, it embeds the structure information to feature information, which aims to obtain the representative data features. But the geodesic function in the graph Laplacian null space is a constant. This results in poor extrapolating power for the GCN model and cannot obtain the manifold structure relationships that fit the data properly.

To extract more structure information of graph data manifold, we utilize graph p-Laplacian to acquire the data manifold structure. Furthermore, we combine the spectral convolution with graph p-Laplacian and reach a new definition for processing signals, i.e. spectral graph p-Laplacian convolution.

$$g_\theta(L_p) \star X = \sum_{k=0}^{k} \theta_k T_k\left(\widetilde{L}_p\right) X \tag{8}$$

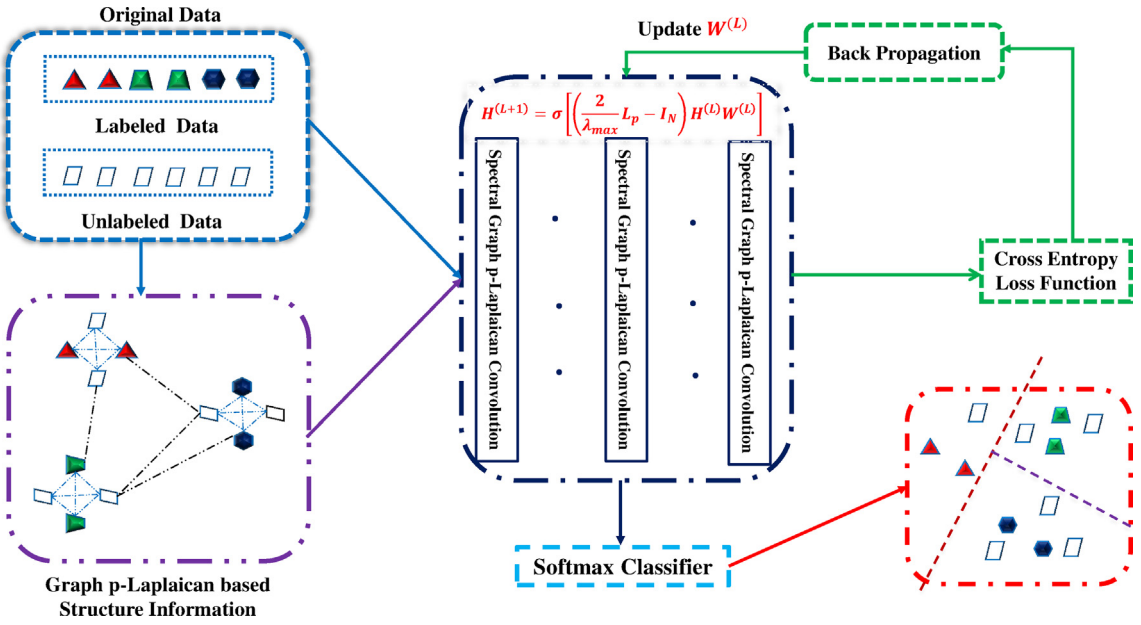

Fig. 2. The basic framework of the multi-layer GpLCN model.

Here, $\widetilde{L}_p = \frac{2}{\lambda_{max}} L_p - I_N$. L_p represents the graph p-Laplacian. I_N is an identity matrix. λ_{max} is the largest eigenvalue in the graph p-Laplacian. θ is the coefficient of the filter.

This definition uses the K-order approximation of the graph p-Laplacian. Each model can be built through multiple operation units defined by Eq. (8). In this paper, we limit the layer-wise convolution operation, i.e. $K = 1$. The goal is to obtain a linear function on the graph p-Laplacian spectrum. It is used to improve the expressive power of the convolution filter function. Thus, we can acquire the simplified formula of a spectral graph p-Laplacian convolution, which is named GpLCN-1. i.e.

$$\begin{aligned} g_\theta(L_p) \star X &= \sum_{k=0}^{k=1} \theta_k T_k\left(\widetilde{L}_p\right) X \\ &= \theta_0 X + \theta_1 \left(\tfrac{2}{\lambda_{max}} L_p - I_N\right) X \end{aligned} \tag{9}$$

This formula has two filter parameters, i.e. θ_0 and θ_1. The filter parameters are shared by each convolution layer. However, in practice, it will increase the computation complexity of each convolution layer. To avoid overfitting and cut down model computation complexity, we further optimize Eq. (9) (In this paper, to distinguish the adjacency relationships between self-connections and avoid the existence of abundant negative numbers in sample structure information, we let $\theta_0 = 0$ and $\theta_1 = \theta$) and reach a final definition of one-order spectral graph p-Laplacian convolution, which is named GpLCN, i.e. layer-wise propagation rule of GpLCN.

$$Z = g_\theta(L_p) \star X = \theta\left(\frac{2}{\lambda_{max}} L_p - I_N\right) X \tag{10}$$

Here, θ is the weight matrix of each convolution layer. $\frac{2}{\lambda_{max}} L_p - I_N$ is the sample manifold structure information. X is the output feature information of last convolution layer. Z is the output of each convolution layer, which is the amalgamation process of sample feature information and manifold structure information.

3.2. GpLCN based on spectral graph p-Laplacian convolution

Thus, we can obtain the following layer-wise propagation rule, i.e.

$$H^{(L+1)} = \sigma\left[\left(\frac{2}{\lambda_{max}} L_p - I_N\right) H^{(L)} W^{(L)}\right] \tag{11}$$

where, graph p-Laplacian $L_p \in R^{N\times N}$ and graph Laplacian $L \in R^{N\times N}$ are positive semi-definite matrices. Graph Laplacian based structure information $\widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}} \in R^{N\times N}$ and graph p-Laplacian based structure information $\left(\frac{2}{\lambda_{max}} L_p - I_N\right) \in R^{N\times N}$ are pre-processing steps and have been calculated before model training. Thus graph p-Laplacian L_p does not increase the computation complexity of GpLCN, i.e. GpLCN model has the same computation complexity $O(|\ \varepsilon\ |)$ as GCN. Kipf et al. [20] detailedly analyzed the complexity of the local structure information. We briefly analyse the network structure of our proposed graph p-Laplacian convolutional networks in Algorithm 1.

Algorithm 1. Multi-layer Graph p-Laplacian Convolutional Networks

Input: Data X
Parameters: Dropout rate, learning rate etc.
Output: Mean classification accuracy
1: Construct p-Laplacian matrix L_p of data.
2: Compute structure information $\widetilde{G} = \frac{2}{\lambda_{max}} L_p - I_N$.
3: Initialize the hyperparameters.
4: **for** $j = 0 \rightarrow k - 1$
5: $H^{(1)} = RELU\left(\widetilde{G} X W^{(0)}\right)$.
6: $H^{(2)} = RELU\left(\widetilde{G} H^{(1)} W^{(1)}\right)$.
7: $\cdots$
8: $H^{(L+1)} = \widetilde{G} H^{(L)} W^{(L)}$.
9: **until convergence**
10: Get the optimal $W^{(0)}$, $W^{(1)}$, $\cdots$ and $W^{(L)}$.
11: Return the mean classification accuracy of data.

In this paper, we use a two-layer GpLCN as an example to demonstrate the effectiveness of our proposed method (In fact, we can also build a deeper GpLCN model. When the number of the model layer increases, the computation efficiency of the model will become worse significantly. However, the classification accuracy of the model may not be effectively improved.). After performing a pre-processing, i.e. $\widetilde{G} = \frac{2}{\lambda_{max}} L_p - I_N$, we have the following convolution layer formula.

$$H^{(2)} = \widetilde{G}\left(RELU\left(\widetilde{G}H^{(0)}W^{(0)}\right)\right)W^{(1)} \tag{12}$$

Fig. 3 illustrates the implementation process of a two-layer GpLCN (the experimental model of this paper) in detail. $W^{(0)}$ is the weight parameter of the first layer. $H^{(0)}$ is feature information of the original input data. We use k-NearestNeighbor with Euclidean distance to construct the adjacency matrix of data, i.e. A. The first convolution layer is the embedding process of sample structure relationships and features information of the original input data. This aims to extract the most representative sample features, i.e. $H^{(1)} = RELU\left(\widetilde{G}H^{(0)}W^{(0)}\right)$. In this paper, we use the *RELU* activation function. The convergence speed will be much faster than the sigmoid and tanh functions. It needs only a threshold to obtain the activation value.

The process of the second convolution layer is similar to that of the first convolution layer. Each layer constructs the manifold structure information of the data samples. $H^{(1)}$ is the sample output feature information of the first layer. $W^{(1)}$ is the weight matrix of the second convolution layer. In the GpLCN model, we use two-layer neural networks to demonstrate our ideal. $H^{(2)} = \widetilde{G}H^{(1)}W^{(1)}$ is an $M \times N$ matrix (M is the number of samples, N is the number of sample classes). It is the final output of the data feature.

Following, we take the extracted sample features as the input of the classifier. In the GpLCN model, we use the *Softmax* classifier. The number of dataset categories determines the number of units in the output layer of *Softmax*. Under the actions of the Softmax classifier, each neural unit calculates the probability that the current sample belongs to this class.

In the training process of the GpLCN model, we need to optimize parameters continuously. Thus, we use the cross-entropy loss function to minimize the error. This aims to let the model fit our training data, i.e.

$$C = -\sum_k y_k \log Z_k \tag{13}$$

y_k is the probability of the true distribution, i.e. the class label. Z_k is the probability estimation that calculates truth data using the *Softmax* classifier. Besides, we treat the accuracy as the evaluation criterion of the GpLCN model.

4. Experiments

In this section, to demonstrate the performance of the proposed GpLCN model, we conduct substantial experiments for node classification on citation network databases, such as Citeseer, Cora and Pubmed [37]. The description of the citation

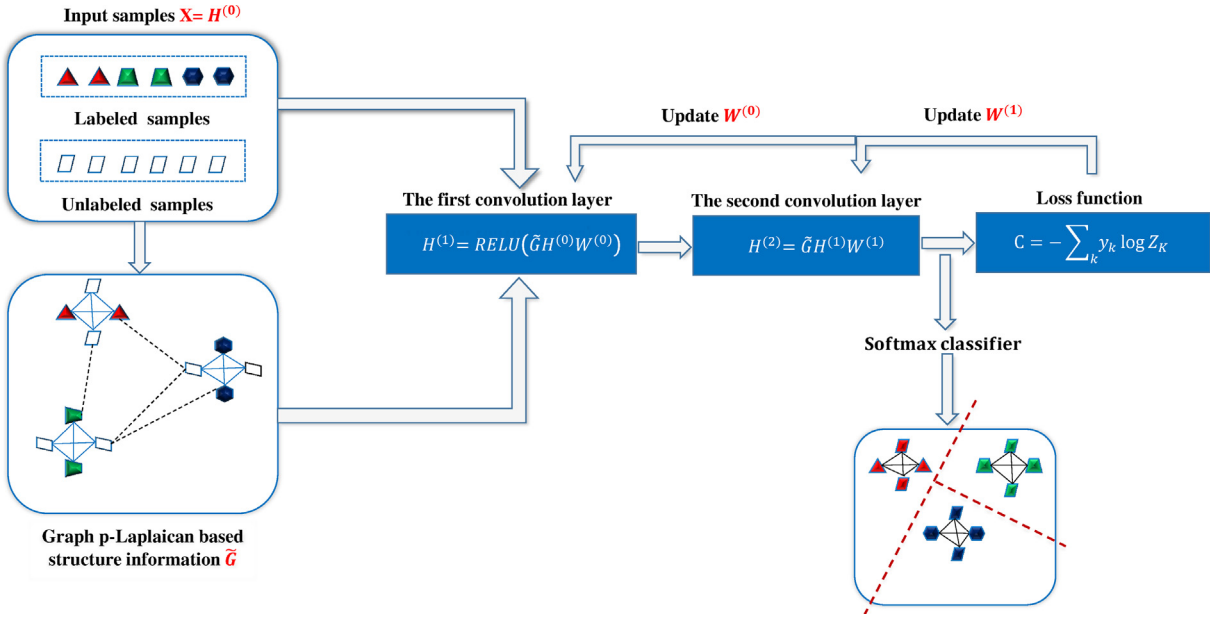

Fig. 3. The detailed implementation process of a two-layer GpLCN (the experimental model of this paper).

network datasets is shown in detail. The parameter settings of the GpLCN model are then given. We finally provide the experiment results.

4.1. Experiment database

The Citeseer dataset [38] contains 3327 machine learning publications. All publications are divided into six classes, such as AI, IR, HCI, Agents, DB and ML. Each publication has 3703 different words. This database contains 4732 document link relationships.

The Cora dataset [39] is composed of 2708 machine learning papers. It can be categorized into seven categories including neural-networks, reinforcement-learning, case-based, probabilistic-methods, theory, genetic-algorithms and rule-learning. It uses 1433 distinct words for each paper and has 5429 link relationships between all papers.

The Pubmed database [40] consists of 19717 science books, which is used for diabetes mellitus patients. This database is selected into three classes, i.e. diabetes mellitus type 2, diabetes mellitus type 1 and diabetes mellitus experimental. Each book is represented by 500 word vectors. The number of all scientific publication links is 44338 (see Table 1).

4.2. Parameter settings

In our experiment, we use all samples of the Citeseer and Cora, and select 5000 data of the Pubmed. For Citeseer, Cora and Pubmed, 500 samples are selected as the validation set, 1000 samples are testing set and the remaining samples are training set. In our semi-supervised experiment of the citation network classification, all samples of the validation set and testing set are labeled data.

For Citeseer and Cora, we select 20%, 30%, 40%, 50% and 60% of the training set as labeled data, respectively, and the remainings are used for unlabeled data. For the Pubmed dataset, we use a certain percentage of training data as labeled data including 10%, 15%, 20%, 25% and 30%, the rests of the samples are unlabeled data.

The training iteration times of the GpLCN model are 200 epochs. We use Adam [41] to optimize model parameters and the learning rate are 0.01 (Citeseer, Cora and Pubmed). If the cross-entropy loss in the training process of the model validation set remains unchanged for 10 consecutive times, our model will stop updating parameters. Besides, we use the L2 regularization method with the coefficient of 5×10^{-4} (Citeseer, Cora and Pubmed) to prevent overtraining. Other training parameters are set as follows. When $p = 2$, Citeseer: 0.6 (dropout rate) and 16 (hidden units), Cora: 0.5 (dropout rate) and 32 (hidden neurons), Pubmed: 0.5 (dropout rate) and 16 (hidden neurons). When $p \neq 2$, Citeseer (p = 1.8): 0.5 (dropout rate) and 32 (hidden units), Cora (p = 1.3): 0.5 (dropout rate) and 32 (hidden units), Pubmed (p = 1.7): 0.5 (dropout rate) and 16 (hidden neurons).

4.3. Experiment results

In this section, we compare the different variants of our GpLCN with many state-of-the-art models. In Tables 2–4 and Figs. 4, 7 and 8, we show the mean recognition rates with five times experiments independently of all models. The mean recognition rates of all categories in the citation network datasets are summarized in Tables 2–4. Figs. 4, 7 and 8 show the mean recognition rates of each class for the citation network datasets. In each figure, the x-axis is the label rate of samples. The y-axis shows the mean recognition rate of each class for the different datasets.

From Table 2, we can see that the recognition rate of the GpLCN and GpLCN-1 model with $p = 2$ is similar to the GCN model on the Citeseer dataset. It demonstrates that the graph p-Laplacian can revert to the standard graph Laplacian when the parameter $p = 2$. In other works, graph Laplacian and p-Laplacian with $p = 2$ can capture similar structure relationships. Our GpLCN outperforms others model at any label rate of samples when $p = 1.8$. It also reveals the superiority of optimization process, i.e. from GpLCN-1 (Eq. (9)) to GpLCN (Eq. (10)). As show in Fig. 7, in most cases GpLCN with $p = 1.8$ performs better than others model at the recognition rate of each class.

Table 3 shows the recognition accuracy of GCN and the diffident variants of GpLCN model on Cora database. From the experiment results, in most cases, we can find that the best method is GpLCN with p = 1.3. Moreover, the GCN, GpLCN-1 and GpLCN models achieve the same performance within the error when p = 2. Fig. 8 demonstrates that the best recognition accuracy of each class is obtained by GpLCN (p = 1.3) under most conditions.

Table 1
The experiment datasets.

Datasets	Nodes	Classes	Links	Features
Citeseer	3327	6	4732	3703
Cora	2708	7	5429	1433
Pubmed	19717	3	44338	500

Table 2
Mean recognition accuracy of all classes in the Citeseer database.

Methods	20%	30%	40%	50%	60%
GCN	66.8	68.66	69.94	71.94	72.4
GpLCN-1 (p = 2)	65.52	68.06	70.24	71.04	72.1
GpLCN (p = 2)	67.18	68.78	70.66	71.6	72.2
GpLCN-1 (p = 1.8)	67.18	69.12	70.86	71.96	72.48
GpLCN (p = 1.8)	67.58	69.38	71.14	72.14	72.64

Table 3
Mean recognition accuracy of all classes in the Cora database.

Methods	20%	30%	40%	50%	60%
GCN	61.34	64.22	66.16	68.14	70.58
GpLCN-1 (p = 2)	61.14	64.38	66.32	68.16	70.7
GpLCN (p = 2)	61.46	64.88	66.62	68.88	70.78
GpLCN-1 (p = 1.3)	61.88	65.64	68	70.84	72.1
GpLCN (p = 1.3)	62.68	66.38	69.18	71.04	72.42

Table 4
Mean recognition accuracy of all classes in the Pubmed database.

Methods	10%	15%	20%	25%	30%
GCN	77.04	78.92	80.3	80.72	81.92
GpLCN-1 (p = 2)	77.14	79	80.3	81.06	82.02
GpLCN (p = 2)	77.42	79.34	80.42	81.16	82.14
GpLCN-1 (p = 1.7)	77.76	80	81.02	82.06	83.12
GpLCN (p = 1.7)	79.38	80.88	81.64	82.66	83.48

As seen from Table 4, as the sample label rate increases, the GpLCN model consistently obtains the best recognition rate. We can observe that the performance of the GCN, GpLCN-1 and GpLCN model is almost unanimous when p is equal to 2. Fig. 4 reveals that GpLCN also has the highest performance under most categories.

From Tables 2–4 and Figs. 4, 7 and 8, we can see that, GCN and GpLCN-1 (p = 2) or GpLCN (p = 2) have the same classification performance in the range that the error allows. The main difference between GCN and GpLCN is the representation of structure information or local geometry structure for graph data. Extensive experiment results also demonstrate that graph Laplacian is the same as the graph p-Laplacian when parameter p = 2, i.e. graph Laplacian and graph p-Laplacian with p = 2 can capture the same manifold structure relationships. Besides, our proposed GpLCN ($p \neq 2$) performs better than other methods. It reveals that the graph p-Laplacian with an appropriate p value can better preserve the local similarity information of the data manifold. (Please note that the parameter $p \geqslant 1$ needs to manually adjust according to different datasets.).

To further demonstrate the performance of our proposed model, we compare with many algorithms based on semi-supervised learning or supervised learning on the Citeseer and Pubmed datasets, such as manifold regularization (ManiReg) [42], semi-supervised embedding (SemiEmb) [43], label propagation (LP) [44], skip-gram based graph embedding (DeepWalk) [45], multi-layer perception (MLP), Chebyshev [36], HyperGCN [22], GAT [24], HesGCN [23], LINE [46], Property features [47], Naive combination [47], PPNE_{ineq} [47], PPNE_{num} [47], TADW [48], Struc2vec [49]. We follow the experimental setting in [36] for Table 6. On the Citeseer and Pubmed datasets, report numbers are the average recognition accuracy with 100 repeated runs in the semi-supervised classification. For Table 5, we give the mean accuracy in the supervised classification and adopt the same experimental setting [47] under the 20% and 30% labeled training data (Notice: for our proposed model, we add the validation set). From Tables 5 and 6, we can observe that our proposed model performs better than the state-of-art solutions. In addition, the entire experimental results of this paper demonstrate the benefit of the graph p-Laplacian to preserve the local geometry of the data. In other words, GpLCN can learn more representative sample features by fusing the graph p-Laplacian-based structure information.

Moreover, we also analyze the run time of our proposed algorithm and give the wall-clock training time until the model convergences in seconds [20]. We can find that GpLCN ($p \neq 2$) performs better than Chebyshev and HyperGCN in the semi-supervised classification and the operating efficiency. With the richer sample structure information, the run times of the model also increases. Thus the proposed GpLCN has a huge benefit compared with GCN.

To further demonstrate the superiority of the proposed method that GpLCN can better preserve the local structure, we introduce two standard numerical analysis criteria for GpLCN, such as Macro-F1 and Micro-F1. In Tables 7–10, we give the Micro-F1 and Macro-F1 with five times experiments of GCN, GpLCN ($p = 2$) and GpLCN ($p \neq 2$) models in percentage. From these data, we can see that how to construct the p-Laplacian matrix with an optimal parameter p is vital. When

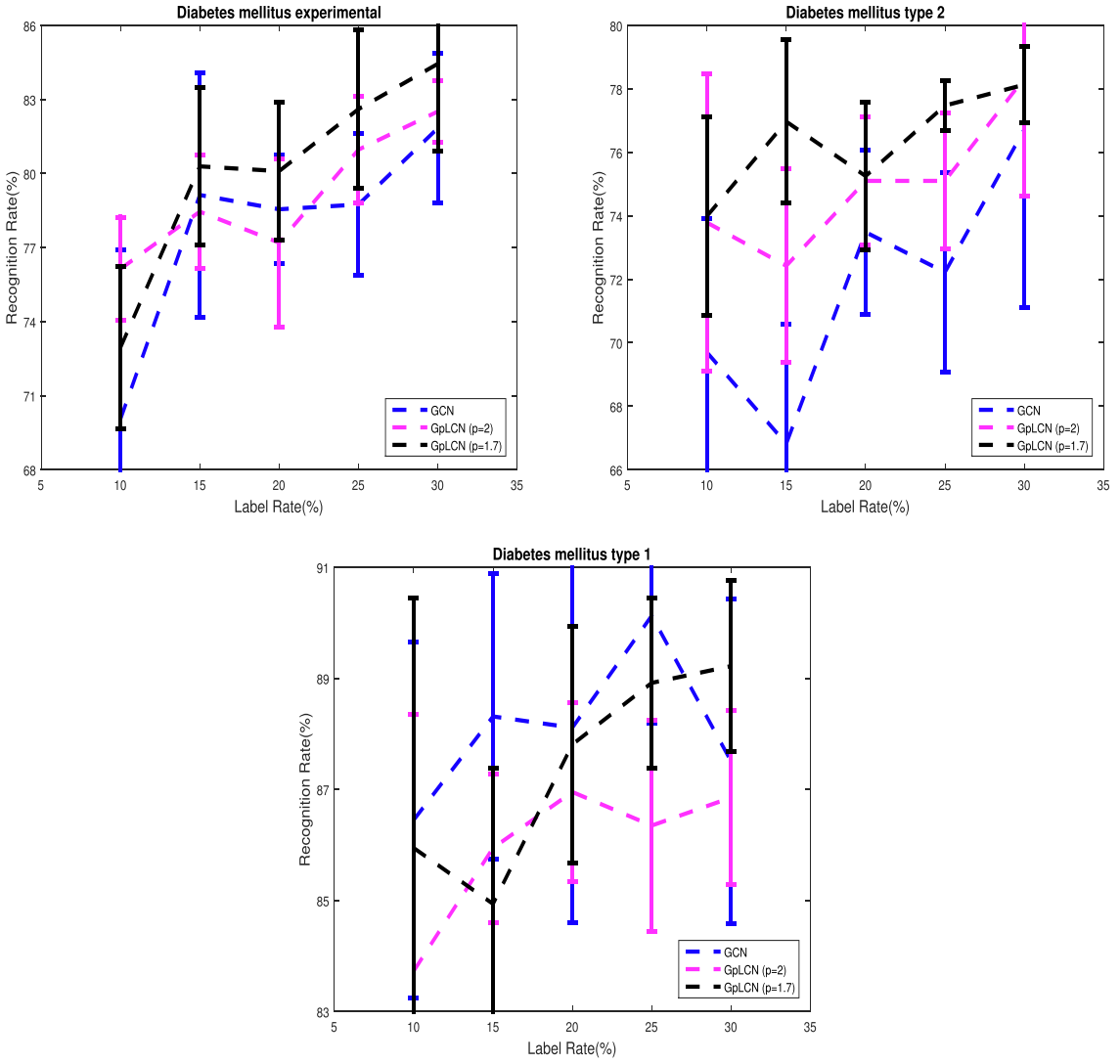

Fig. 4. Mean recognition for each class of Pubmed database, including Diabetes mellitus experimental, Diabetes mellitus type 1, Diabetes mellitus type 2. Each subfigure corresponds on single class.

$p \neq 2$, whether it is Micro-F1 and Macro-F1 or Pubmed database and Cora database, GpLCN all have the best performance because it integrates graph p-Laplacian learning.

For our proposed GpLCN, it exists a large number of parameters to train. We note that the most important parameter in our method is the selection of the parameter p. Thus, we conduct extensive experiments for analyzing parameter p to validate the effectiveness of the proposed method. In Figs. 5 and 6, it shows the mean average recognition accuracy of the GpLCN with different p values under the 20% and 30% labeled samples of the Citeseer and Cora database. From those data, we can find that the GpLCN shows its best superiority when $p = 1.8$ in the Citeseer database, while the best classification performance of the GpLCN is acquired in the Cora database when $p = 1.3$.

5. Conclusion

We have proposed a p-Laplacian method to preserve the local structure information of data. The null space of p-Laplacian is rich, and it can use the geodesic distance to extrapolate smoothly for unseen data. We applied the graph p-Laplacian matrix to spectral graph convolution, i.e. spectral graph p-Laplacian convolution. Finally, we have proposed a new model graph p-Laplacian convolutional networks (GpLCN) for node classification. The GpLCN is based on the one-order approximation for spectral graph p-Laplacian convolution. We have provided the experiment results on the Citeseer, Cora and Pubmed datasets to demonstrate the effectiveness of our GpLCN model in comparison to the GCN model.

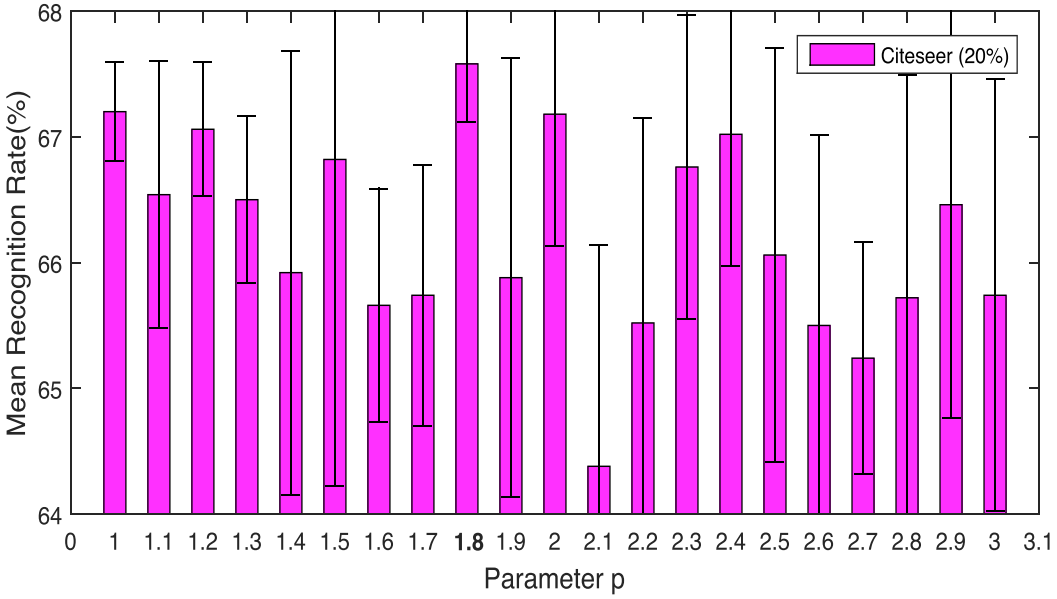

Fig. 5. Mean recognition accuracy with different p values in the Citeseer database.

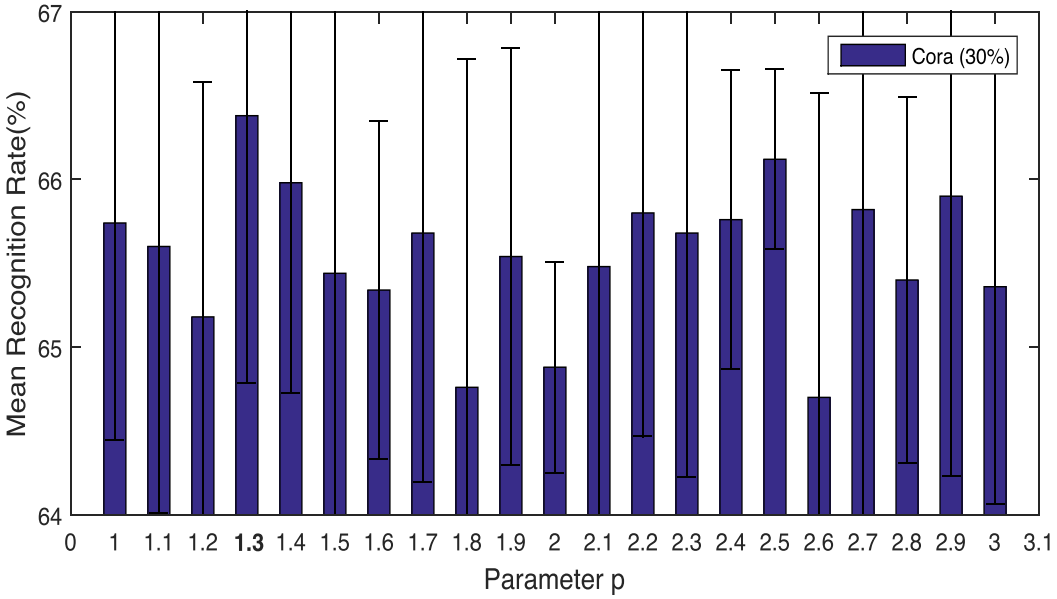

Fig. 6. Mean recognition accuracy with different p values in the Cora database.

CRediT authorship contribution statement

Sichao Fu: Conceptualization, Methodology, Formal analysis, Writing - original draft. **Weifeng Liu:** Supervision, Funding acquisition, Writing - review & editing. **Kai Zhang:** Supervision, Funding acquisition, Supervision. **Yicong Zhou:** Supervision, Funding acquisition, Writing - review & editing, Writing - review & editing. **Dapeng Tao:** Supervision, Funding acquisition.

Declaration of Competing Interest

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

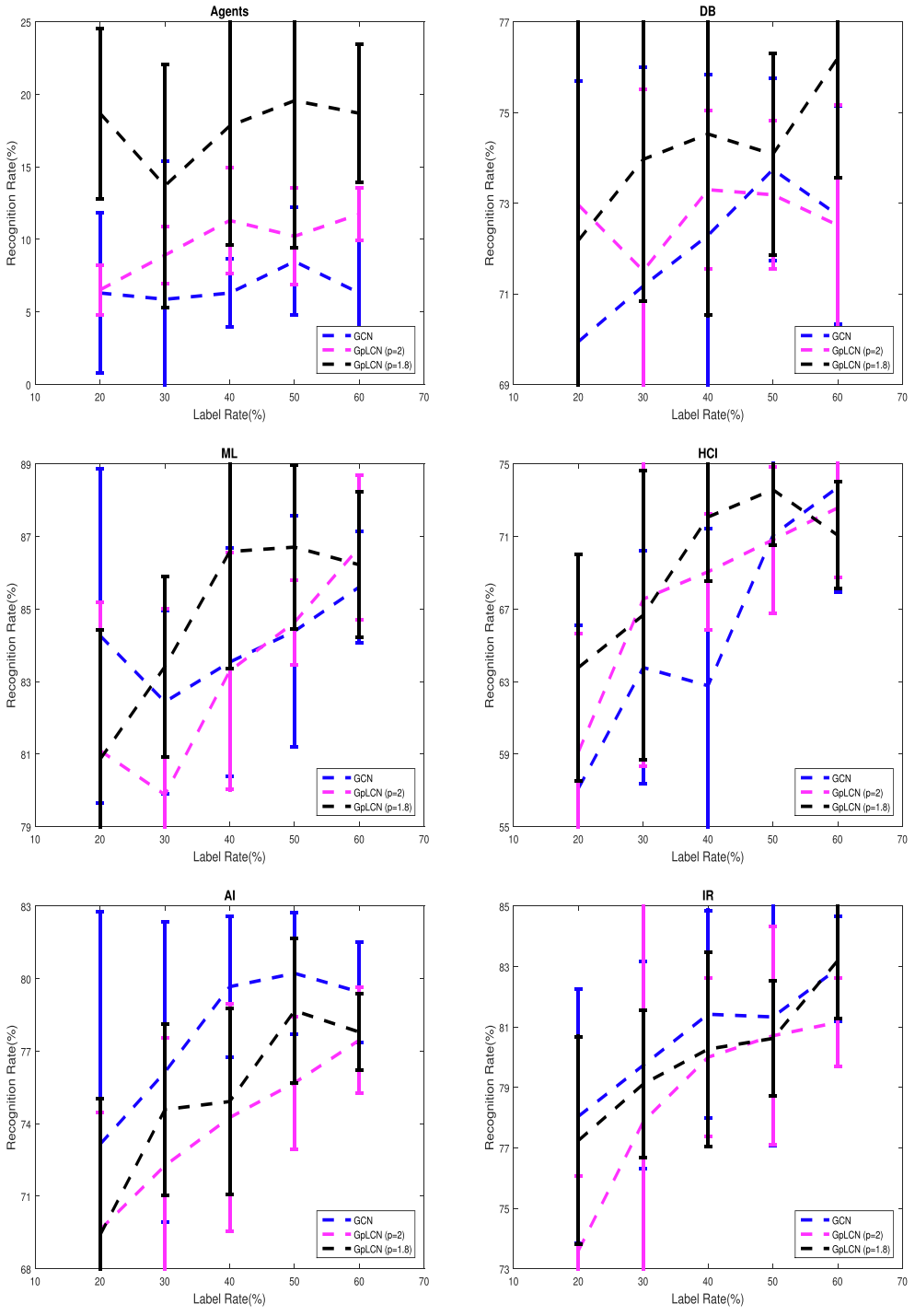

Fig. 7. Mean recognition accuracy for each class of Citeseer database, including Agents, AI, DB, IR, ML, HCI. Each subfigure corresponds on single class.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61671480, in part by the Major Scientific and Technological Projects of CNPC under Grant ZD2019-183-008, in part by the Yunnan Natural Science Funds under Grant 2018FY001(-013) and Grant 2018YDJQ004, and by the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (Grant No. 202000009).

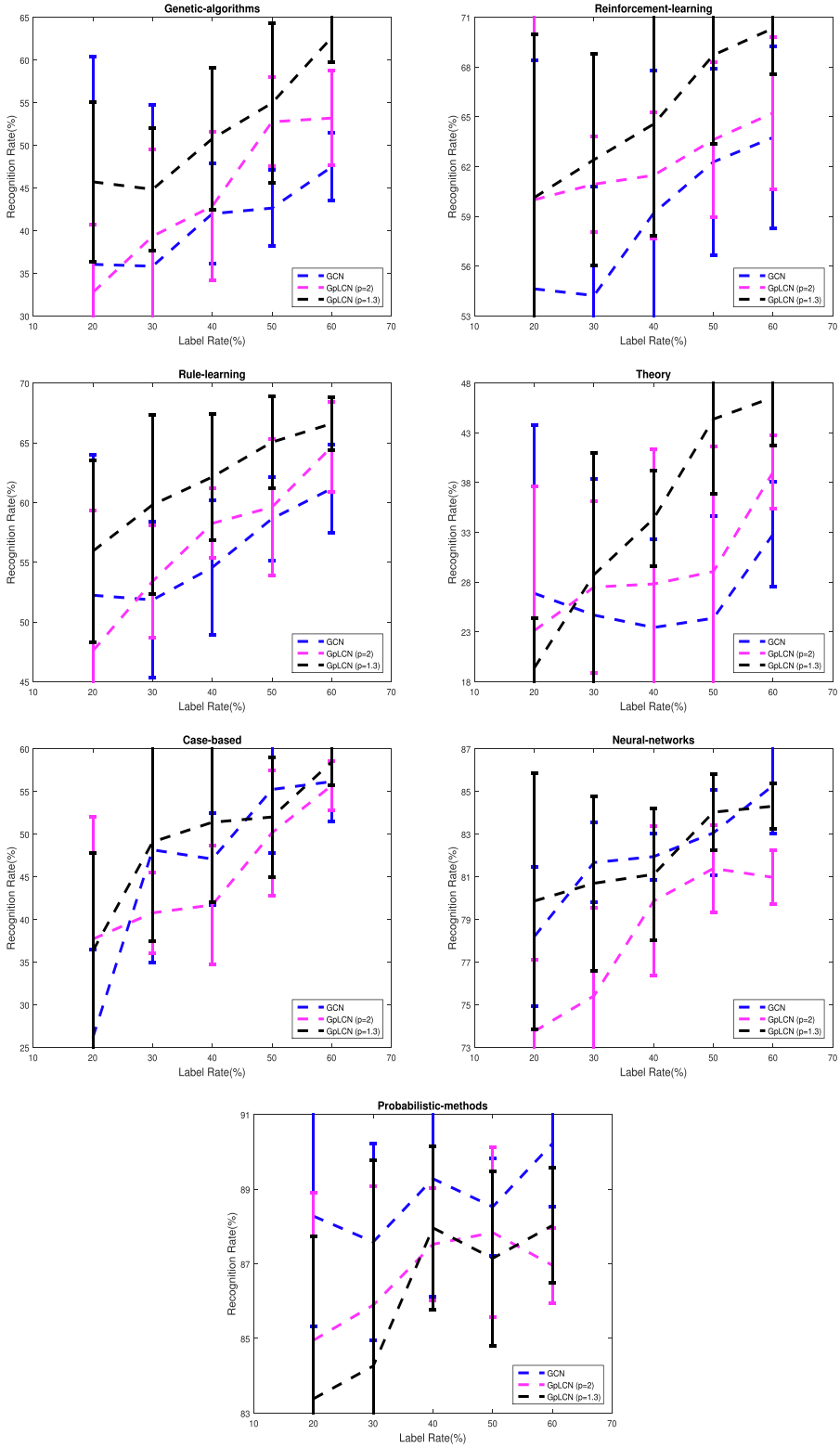

Fig. 8. Mean recognition accuracy for each class of Cora database, including Theory, Case-based, Genetic-algorithms, Neural-networks, Probabilistic-methods, Reinforcement-learning, Rule-learning. Each subfigure corresponds on single class.

Table 5
Mean Accuracy of Nine Models Under the 20% and 30% Labeled Training Data.

Methods	Citeseer (20%)	Citeseer (30%)
DeepWalk	53.5	56.5
LINE	45.8	49.5
Struc2vec	20.9	21.2
Property features	55.8	58.4
Naive combination	56.5	60.5
TADW	58.5	61.8
$PPNE_{ineq}$	63.2	66.1
$PPNE_{num}$	62.7	65.5
GpLCN ($p \neq 2$)	69.8	71.3

Table 6
The Experiment Results on the Citeseer and Pubmed.

Methods	Citeseer (120)	Pubmed (60)
ManiReg	60.1	70.7
SemiEmb	59.6	71.1
LP	45.3	63
DeepWalk	43.2	65.3
MLP	46.5	71.4
GAT	59.8	70.9
HesGCN	60.6	71.5
Chebyshev	54.3 (81s)	68.8 (199s)
GCN	52.8 (15s)	71 (60s)
HyperGCN	55 (50s)	71.3 (118s)
GpLCN ($p \neq 2$)	61.5 (22s)	72.1 (97s)

Table 7
Micro-F1 with five times experiments of all classes in the Cora database.

Methods	20%	30%	40%	50%	60%
GCN	61.3	64.2	66.2	68.1	70.6
GpLCN (p = 2)	61.4	64.9	66.6	68.9	70.8
GpLCN (p = 1.3)	62.7	66.4	69.2	71	72.4

Table 8
Macro-F1 with five times experiments of all classes in the Cora database.

Methods	20%	30%	40%	50%	60%
GCN	53.5	57.4	60.1	62.1	65.8
GpLCN (p = 2)	54.6	57.8	61.4	64.1	66.9
GpLCN (p = 1.3)	57.1	61.2	64.1	67.3	68.5

Table 9
Micro-F1 with five times experiments of all classes in the Pubmed database.

Methods	10%	15%	20%	25%	30%
GCN	77	78.9	80.3	80.7	81.9
GpLCN (p = 2)	77.4	79.3	80.4	81.1	82.1
GpLCN (p = 1.7)	79.4	80.8	81.6	82.6	83.5

Table 10
Macro-F1 with five times experiments of all classes in the Pubmed database.

Methods	10%	15%	20%	25%	30%
GCN	76.8	78.9	80.2	80.6	81.8
GpLCN (p = 2)	76.9	79.4	80.4	81	82.1
GpLCN (p = 1.7)	79.2	80.9	81.6	82.5	83.4

References

[1] Y. Chen, K. Sun, J. Pu, Z. Xiong, X. Zhang, Grapasa: Parametric graph embedding via siamese architecture, Inform. Sci. 512 (2020) 1442–1457.
[2] C. Giladi, A. Sintov, Manifold learning for efficient gravitational search algorithm, Inform. Sci. 517 (2020) 18–36.
[3] S. Hu, X. Yan, Y. Ye, Joint specific and correlated information exploration for multi-view action clustering, Inform. Sci. (2020).
[4] W. Yan, Q. Sun, H. Sun, Y. Li, Joint dimensionality reduction and metric learning for image set classification, Inform. Sci. 516 (2020) 109–124.
[5] J. Jiang, Y. Yu, Z. Wang, X. Liu, J. Ma, Graph-regularized locality-constrained joint dictionary and residual learning for face sketch synthesis, IEEE Trans. Image Process 28 (2019) 628–641.
[6] C. Tang, M. Bian, X. Liu, M. Li, H. Zhou, P. Wang, H. Yin, Unsupervised feature selection via latent representation learning and manifold regularization, Neural Networks 117 (2019) 163–178.
[7] X. Ye, J. Zhao, Multi-manifold clustering: A graph-constrained deep nonparametric method, Pattern Recogn. 93 (2019) 215–227.
[8] Z. Li, J. Tang, L. Zhang, J. Yang, Weakly-supervised semantic guided hashing for social image retrieval, Int. J. Comput. Vis. 128 (2020) 2265–2278.
[9] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.
[10] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[11] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in: Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2002, pp. 585–591 .
[12] R. Souvenir, R. Pless, Manifold clustering, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), volume 1, 2005, pp. 648–653 .
[13] M. Breitenbach, G. Z. Grudic, Clustering through ranking on manifolds, in: Proc. Int. Conf. Mach. Learn. (ICML), 2005, pp. 73–80 .
[14] J. Ye, Z. Zhao, H. Liu, Adaptive distance metric learning for clustering, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2007, pp. 1–7 .
[15] J.E. Van Engelen, H.H. Hoos, A survey on semi-supervised learning, Mach. Learn. 109 (2020) 373–440.
[16] W. Liu, S. Fu, Y. Zhou, Z.-J. Zha, L. Nie, Human activity recognition by manifold regularization based dynamic graph convolutional networks, Neurocomputing (2020), https://doi.org/10.1016/j.neucom.2019.12.150.
[17] W. Liu, D. Tao, Multiview hessian regularization for image annotation, IEEE Trans. Image Process. 22 (2013) 2676–2687.
[18] D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in: Proc. IEEE Int. Conf. Comput. Vis. (ICCV), 2007, pp. 1–7 .
[19] E. Kokiopoulou, D. Kressner, P. Frossard, Optimal image alignment with random projections of manifolds: algorithm and geometric analysis, IEEE Trans. Image Process. 20 (2011) 1543–1557.
[20] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proc. IEEE Int. Conf. Learn. Representations (ICLR), 2017 .
[21] S. Fu, W. Liu, S. Li, Y. Zhou, Two-order graph convolutional networks for semi-supervised classification, IET Image Process. 13 (2019) 2763–2771.
[22] N. Yadati, M. Nimishakavi, P. Yadav, A. Louis, P. Talukdar, Hypergcn: Hypergraph convolutional networks for semi-supervised classification, in: Proc. IEEE Int. Conf. Multimedia and Expo (ICME), 2019 .
[23] S. Fu, W. Liu, D. Tao, Y. Zhou, L. Nie, Hesgcn: Hessian graph convolutional networks for semi-supervised classification, Inform. Sci. 514 (2019).
[24] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proc. IEEE Int. Conf. Learn. Representations (ICLR), 2018 .
[25] Y. Liu, W. Chen, H. Qu, S.H. Mahmud, K. Miao, Weakly supervised image classification and pointwise localization with graph convolutional networks, Pattern Recogn. 109 (2021) , https://doi.org/10.1016/j.patcog.2020.107596 107596.
[26] B. Hua, L. Wang, Dirichlet p-laplacian eigenvalues and cheeger constants on symmetric graphs, Adv. Math. 364 (2020) 106997.
[27] S. Amghibech, Eigenvalues of the discrete p-laplacian for graphs, Ars Combinatoria 67 (2003) 283–302.
[28] D. Slepcev, M. Thorpe, Analysis of p-laplacian regularization in semisupervised learning, SIAM J. Math. Anal. 51 (2019) 2085–2120.
[29] S. Tan, H. Tan, A microcellular communications propagation model based on the uniform theory of diffraction and multiple image theory, IEEE Trans. Antennas Propag. 44 (1996) 1317–1326.
[30] W. Liu, X. Ma, Y. Zhou, D. Tao, J. Cheng, p-laplacian regularization for scene recognition, IEEE Trans. Cybern. (2018) 1–14.
[31] D. Luo, H. Huang, C. Ding, F. Nie, On the eigenvectors of p-laplacian, Mach. Learn. 81 (2010) 37–51.
[32] W. Liu, Z.-J. Zha, Y. Wang, K. Lu, D. Tao, p-laplacian regularized sparse coding for human activity recognition, IEEE Trans. Ind. Electron. 63 (2016) 5120–5129.
[33] X. Zhang, W. Ye, An adaptive second-order partial differential equation based on tv equation and p-laplacian equation for image denoising, Multimed. Tools Appl. 78 (2019) 18095–18112.
[34] X. Ma, W. Liu, D. Tao, Y. Zhou, Ensemble p-laplacian regularization for scene image recognition, Cogn. Comput. 11 (2019) 841–854.
[35] S. Bougleux, O. Lezoray, A. Nouri, 3d colored mesh structure-preserving filtering with adaptive p-laplacian on directed graphs, in: Proc. IEEE Int. Conf. Image Process. (ICIP), 2019, pp. 4380–4384 .
[36] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2016, pp. 3844–3852 .
[37] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI Mag. 29 (2008) 93.
[38] H. Poon, P. Domingos, Joint inference in information extraction, in: Proc. 22nd AAAI Conf. Artif. Intell. (AAAI), vol. 7, 2007, pp. 913–918 .
[39] C. Cabanes, A. Grouazel, K. v. Schuckmann, M. Hamon, V. Turpin, C. Coatanoan, F. Paris, S. Guinehut, C. Boone, N. Ferry, The cora dataset: validation and diagnostics of in-situ ocean temperature and salinity measurements, Ocean Sci. J. 9 (2013) 1–18.
[40] M. Bilgic, L. Licamele, L. Getoor, B. Shneiderman, D-dupe: An interactive tool for entity resolution in social networks, in: Proc. IEEE Symposium on Visual Analytics Science and Technology, 2006, pp. 43–50 .
[41] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proc. IEEE Int. Conf. Learn. Representations (ICLR), 2015 .
[42] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: A geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[43] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural Networks: Tricks of the Trade, 2012, pp. 639–655.
[44] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Proc. Adv. Neural Inf. Process. Syst. (NIPS), 2004, pp. 321–328 .
[45] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: Online learning of social representations, in: Proc. Int. Conf. on Knowledge Discovery and Data Mining (ACM SIGKDD), 2014, pp. 701–710 .
[46] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, Line: Large-scale information network embedding, in: Proc. Int. Conf. World Wide Web (WWW), 2015, pp. 1067–1077 .
[47] C. Li, S. Wang, D. Yang, Z. Li, Y. Yang, X. Zhang, J. Zhou, Ppne: property preserving network embedding, in: Proc. Int. Conf. Database Systems for Advanced Applications, Springer, 2017, pp. 163–179 .
[48] C. Yang, Z. Liu, D. Zhao, M. Sun, E.Y. Chang, Network representation learning with rich text information., in: Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), vol. 2015, 2015, pp. 2111–2117 .
[49] L.F. Ribeiro, P.H. Saverese, D.R. Figueiredo, struc2vec: Learning node representations from structural identity, in: Proc. Int. Conf Knowledge Discovery and Dta Mining (ACM SIGKDD), 2017, pp. 385–394 .