Neurocomputing 362 (2019) 166–174

Contents lists available at ScienceDirect

Neurocomputing

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

HpLapGCN: Hypergraph *p*-Laplacian graph convolutional networks

Sichao Fu[a], Weifeng Liu[a,*], Yicong Zhou[b], Liqiang Nie[c]

[a] College of Information and Control Engineering, China University of Petroleum (East China), Qingdao 266580, China
[b] Faculty of Science and Technology, University of Macau, Macau 999078, China
[c] School of Computer Science, Shandong University, Qingdao 266237, China

ARTICLE INFO

Article history:
Received 5 March 2019
Revised 12 June 2019
Accepted 23 June 2019
Available online 19 July 2019

Communicated by Jun Yu

Keywords:
Graph convolutional networks
Hypergraph
p-Laplacian

ABSTRACT

Currently, the representation learning of a graph has been proved to be a significant technique to extract graph structured data features. In recent years, many graph representation learning (GRL) algorithms, such as Laplacian Eigenmaps (LE), Node2vec and graph convolutional networks (GCN), have been reported and have achieved great success on node classification tasks. The most representative GCN fuses the feature information and structure information of data, which aims to generalize convolutional neural networks (CNN) to learn data features with arbitrary structure. However, how to exactly express the structure information of data is still an enormous challenge. In this paper, we utilize hypergraph *p*-Laplacian to preserve the local geometry of samples and then propose an effective variant of GCN, i.e. hypergraph *p*-Laplacian graph convolutional networks (HpLapGCN). Since hypergraph *p*-Laplacian is a generalization of the graph Laplacian, HpLapGCN model shows great potential to learn more representative data features. In particular, we simplify and deduce a one-order approximation of spectral hypergraph *p*-Laplacian convolutions. Thus, we can get a more efficient layer-wise aggregate rule. Extensive experiment results on the Citeseer and Cora datasets prove that our proposed model achieves better performance compare with GCN and *p*-Laplacian GCN (pLapGCN).

1. Introduction

With the richness and expansion of the current social life, the structure presented by the data [1,2] has become more and more complicated. How to effectively manage and apply these complex structured data [3,4] has become a serious challenge for researchers. Graphs are the frequently used type of data structure that can describe the intricate connections between things. Many tasks [5] in the real world can be described as graph problems, so more and more emerging technology fields [6,7] begin to represent complex data by means of graph models. For example, in the field of biotechnology, graph models are used to describe the internal structure of protein [8]. In the field of social networking, graph structures are used to describe the relationships between a large number of people or groups [9]. At present, plenty of methods for GRL have been widely used to extract the data features with the complex structure [10,11], such as natural language processing [12–14], image processing [15–17] and so on [18,19].

In the past few years, many traditional graph embedding (network embedding) models including LE [20], Node2vec [21], CANE [22], SDNE [23] have made a great development. Recently, the GRL models based on spectral theory have drawn a widely public attention [24–26], i.e. GCN. More optimized GCN models have been proposed and are categorized as spectral convolution approaches [27–29] and spatial convolution approaches [30–32].

The spectral convolution models [27–29] make the convolution transformation in the Fourier domain. Kipf and Welling [27] used a convolutional network to fuse the graph Laplacian-based structure information and feature information of data, which aimed to effectively generalize convolutional neural networks [33,34] to learn the data features that have arbitrary graph structures. Yadati et al. [28] dealt with the problem of extending graph convolutional networks to a hypergraph, not to a standard graph by considering the hypergraph Laplacian-based structure information. Zhuang and Ma [29] proposed a dual graph convolutional network to consider both the local consistency and global consistency of a graph.

The spatial convolution models [30–32] directly make a convolution operation on the graphs. Atwood and Towsley [30] captured the diffusion-based representations from graph structured data by introducing the diffusion convolution, which aimed to extend convolutional neural networks [35] to graph structured data. Veličković et al. [31] distributed a different weight for the neighborhoods of each node by using the attention mechanism, which can get the hidden representations of each node from the graphs.

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

https://doi.org/10.1016/j.neucom.2019.06.068
0925-2312/

Niepert [32] proposed a PATCHY-SAN (Select-Assemble-Normalize) frame to make a preprocessing process from graph to vector mapping, and then made the corresponding convolution operations.

However, the geodesic function of the graph Laplacian null space is a constant, which results in incorrectly predicting samples beyond the scope of the training datasets [36,37]. Thus, GCN cannot get the detailed manifold structure information of data properly.

In this paper, we propose a hypergraph p-Laplacian graph convolutional network (HpLapGCN) for semi-supervised classification. First of all, the hypergraph p-Laplacian [38–40] has provided the convincing theoretical evidence to better preserve the local geometric structure of data. Thus, the combinant of the Hypergraph and p-Laplacian is used, i.e. Hypergraph p-Laplacian matrix. In the next place, we apply the hypergraph p-Laplacian to the spectral graph convolutions, and then we introduce a different definition, i.e. spectral hypergraph p-Laplacian convolutions. Finally, we get a more efficient layer-wise aggregation rule by the derivation and simplification of the one-order approximation for spectral graph convolutions. A deep HpLapGCN can be built through this layer-wise aggregation rule. We conduct substantial experiments on public datasets including Citeseer and Cora to demonstrate the performance of the HpLapGCN. From the experimental data, we can see that, the HpLapGCN outperforms the GCN and pLapGCN.

The main contributions of this work are two-fold:

1) This paper proposes the hypergraph p-Laplacian to preserve the geometry of the probability distribution, and then gets a more efficient convolution formulation by the derivation and simplification of the one-order spectral hypergraph p-Laplacian convolutions.
2) This paper proposes a hypergraph p-Laplacian graph convolutional network to improve the semi-supervised classification performance. The experimental results validate the effectiveness of their method.

The rest of this paper is organized as follows. Section 2 briefly describes the related works on the hypergraph p-Laplacian theory and the basic GCN model. Section 3 and Section 4 detail the proposed HpLapGCN framework and motivation. Section 5 describes the experimental results of HpLapGCN. And finally Section 6 is the conclusion.

2. Related works

Our proposed model is motivated by the hypergraph p-Laplacian theory and the basic GCN model. In this section, we briefly describes the related works of the HpLapGCN.

2.1. Hypergraph p-Laplacian theory

In the traditional graph theory, we often assume that there are only pairwise relationships between the different objects. These pairwise relationships can be represented through a simple graph. In a simple graph, a vertex represents an object and the relationships between two objects are represented by an edge of the simple graph. However, in the real world, it is difficult to describe the complex relationships between the different objects by only using a simple graph [41]. The main difference between a hypergraph and a simple graph is the different numbers of vertex on the edge. The edges of a simple graph have only two vertices. In a hypergraph, each edge can connect N vertices ($N \geq 2$). In order to distinguish from the edges of the simple graph, the edges of the hypergraph are called hyperedges. Fig. 1 shows the difference between a hypergraph and a simple graph.

In a hypergraph $G = (V, E, W)$, V is a finite set of vertices and E, i.e. a family of subsets of V, are the sets of hyperedges. Each hyperedge $e \in E$ is given a nonnegative weight $w(e)$. The structure of the hypergraph can be described by a $|V| \times |E|$ dimensional correlation matrix H. In addition, if a vertex v is located in a hyperedge e, the $h(v, e)$ is assigned a value of one, else the $h(v, e)$ is given a value of zero. For each vertex $v \in V$, its degree matrix is defined as the following expression:

$$d(v) = \sum_{e \in E} w(e) h(v, e) \tag{1}$$

For each hyperedge $e \in E$, its degree matrix are denoted as the numbers of total vertices on the hyperedges, i.e.

$$\delta(e) = \sum_{v \in V} h(v, e) \tag{2}$$

In addition, the D_V is the diagonal degree matrix of each vertex. The D_e denotes the diagonal matrices including the degree matrix of each hyperedge. W are the diagonal matrix of hyperedge weights. In the construction process of hypergraph, we first calculate the weight W^s of simple graph by the k-nearest neighbor with the Euclidean distance. And then we regard each node and its k-nearest neighbor nodes as the nodes of a hyperedge. Finally, we can get the weight W of hypergraph.

Thus, we can get the definition of the adjacency matrix of hypergraph, i.e.

$$W^h = HWH^T - D_V \tag{3}$$

At present, the learning methods of hypergraph can be mainly divided into three categories. Specially, a hypergraph is built from data. The first methods, including clique expansion [42], star expansion [42], Rodriguez's Laplacian [43] and clique averaging [44], are to construct a simple graph based on the hypergraph, and then use the spectral clustering methods based on the simple graph to segment the hypergraph. The second type of methods are a tensor-based hypergraph learning method [45,46]. In this type of methods, the hypergraph structure is described by a tensor. Then the hypergraph is segmented by using the joint clustering methods. The third type of methods are to generalize a simple graph Laplacian to a hypergraph Laplacian, such as Zhou's normalized Laplacian [47] and Bolla Laplacian [48]. In [47], the normalized hypergraph Laplacian is proposed and is denoted as:

$$L^{nh} = I_N - D_V^{-\frac{1}{2}} HWD_e^{-1}H^T D_V^{-\frac{1}{2}} \tag{4}$$

In addition, the D_e (edge matrix) is equal to $2I_N$ in a simple graph. Then, the standard graph Laplacian is defined as the following form:

$$\bar{L} = \frac{1}{2}\left(I_N - D_V^{-\frac{1}{2}} W^h D_V^{-\frac{1}{2}}\right) \tag{5}$$

p-Laplacian L_p (high-order) is a generation of Laplacian (one-order). Graph p-Laplacian gets tighter isoperimetric inequality, thus the upper and lower bounds on the second eigenvalue approximates the optimal Cheeger cut value well [49]. In addition, Liu et al. [36] have demonstrated the differences of Laplacian and p-Laplacian. That is to say, it has been proved that the L_p can extrapolate smoothly to unseen data that have the geodesic distance [36], i.e. it has richer extrapolation capability. The detailed math theoretical analysis also can found in [36]. In addition, the p is a fundamental parameter of L_p. Currently, Luo et al. [50] utilized the p-Laplacian to solve the clustering problem. Zhou and Scholkopf [51] built a p-Laplacian based discrete regularization framework for the classification problem.

Recently, Ma et al. [38] generalized the hypergraph Laplacian to hypergraph p-Laplacian HL_p, and then utilized the hypergraph p-Laplacian regularization to express the local geometry of data for the remote sensing image classification. The computational process of HL_p are divided into two parts, i.e. firstly, it constructs

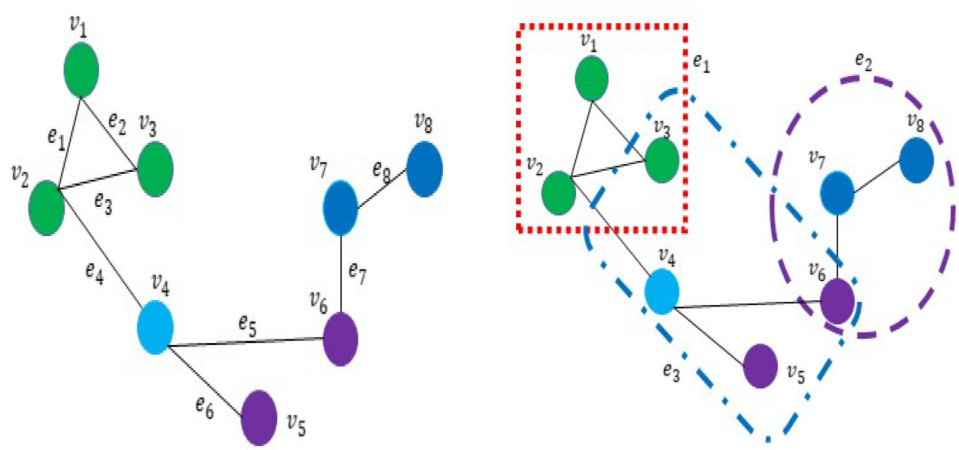

Fig. 1. The difference between a hypergraph and a simple graph. Left: a simple graph: each edge of a simple graph have only two vertices. Right: a hypergraph. In a hypergraph, each edge can connect N vertices ($N \geq 2$).

the hypergraph Laplacian HL by means of W^h (i.e. $HL = D^h - W^h$, $D^h_{ii} = \sum_j W^h_{ij}$), then it generalizes the HL to HL_p. The detailed process can be found in [38].

2.2. Graph convolutional networks

Defferrard et al. [24] proposed an optimized spectral graph convolution formula, which is defined as the convolution of signal $X \in \mathbb{R}^{\mathbb{N}\times\mathbb{C}}$ (C are the numbers of the dimensional for each sample) with a filter g_θ that has F features maps, i.e.

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

Here, θ is a filter coefficient matrix. $T_k(X)$ is a Chebyshev polynomial sequence with recursive form, i.e. $T_0(X) = 1, T_1(X) = X, T_k(X) = 2XT_{k-1}(X) - T_{k-2}(X)$. $\widetilde{L}$ is rescaled through $\widetilde{L} = \frac{2}{\lambda^L_{max}}L - I_N$. λ^L_{max} is the largest eigenvalue of the graph Laplacian L. In this definition, Defferrard et al. [24] used the normalized Laplacian matrix, i.e. $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, $D_{ii} = \sum_j A_{ij}$. The adjacency matrix A is a matrix that represent the adjacent relationships between the samples. I_N represents the identity matrix. Defferrard et al. [24] used the K-order network neighborhood on each convolution layer.

To build a linear model, Kipf et al. [27] only used the node's direct neighborhood on each iteration, i.e. $K = 1$. Finally, they proposed an efficient layer-wise definition, i.e.

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

Here, $\bar{H}^{(L+1)}$ is the sample feature matrix of each layer, $\bar{H}^{(0)} = X$. $\bar{W}^{(L)}$ is trainable parameters matrix of each layer. Kipf et al. [27] used the nonlinear activation function $RELU$, i.e. $f(x) = max(0, x)$. $\widetilde{A} = A + I_N$ represents the sample's adjacency relationships including self-connections. $\widetilde{D}_{ii} = \sum_j \widetilde{A}_{ij}$. On each aggregation iteration process, each sample can get the sample structure information with the node's direct neighborhood. Each convolution is the fusion process of the structure information and feature information.

3. Motivation

Inspired by the spectral graph convolutions, we introduce a novel definition by combining the hypergraph p-Laplacian and the spectral convolutions on graphs, i.e. spectral hypergraph p-Laplacian convolutions. Moreover, we get a different layer-wise aggregate rule by optimizing the one-order polynomial of spectral hypergraph p-Laplacian convolutions. Specifically, the definition of spectral hypergraph p-Laplacian convolutions is showed first. The one-order approximation of hypergraph p-Laplacian convolutions is given then. In this part, we explain the motivation of HpLapGCN model and introduce the derivation and optimization process of layer-wise aggregate rule.

3.1. Spectral hypergraph p-Laplacian convolutions

Because of the poor null space of the Laplacian matrix [36,52], i.e. the manifold structure information of Laplacian is not rich, and then the extracted sample features of GCN are not representative. To get the richer sample features, we use the hypergraph p-Laplacian matrix to preserve the manifold structure of the data. Then, we apply the hypergraph p-Laplacian to spectral graph convolutions. Finally, we can get different spectral convolutions on the graph, i.e. spectral hypergraph p-Laplacian convolutions.

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

Here, $\widetilde{HL_p} = \frac{2}{\lambda_{max}}HL_p - I_N$. λ_{max} denotes the largest eigenvalue in the hypergraph p-Laplacian. In addition, $T_0(\widetilde{HL_p}) = I_N, T_1(\widetilde{HL_p}) = \widetilde{HL_p}, T_k(\widetilde{HL_p}) = 2\widetilde{HL_p}T_{k-1}(\widetilde{HL_p}) - T_{k-2}(\widetilde{HL_p})$. In this paper, we construct the HL_p using the method described in [38].

3.2. One-order approximation of hypergraph p-Laplacian convolutions

To build a linear model and reduce the calculation of the model, we also use the one-order approximation of spectral hypergraph p-Laplacian convolutions, i.e. $K = 1$. Thus, the above formula can be simplified, which is named HpLapGCN-1, i.e.

$$g_\theta(HL_p) \star X = \theta_0 X + \theta_1\left(\frac{2}{\lambda_{max}}HL_p - I_N\right)X \tag{9}$$

It has two filter parameters θ_0 and θ_1. The two different filter parameters can be used on each convolution layer. Specially, we constraint the filter parameters to further reduce model's convolution operations and avoid the overfitting problem. Thus we can get an optimal formula, which is named HpLapGCN, i.e. layer-wise aggregate rule.

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

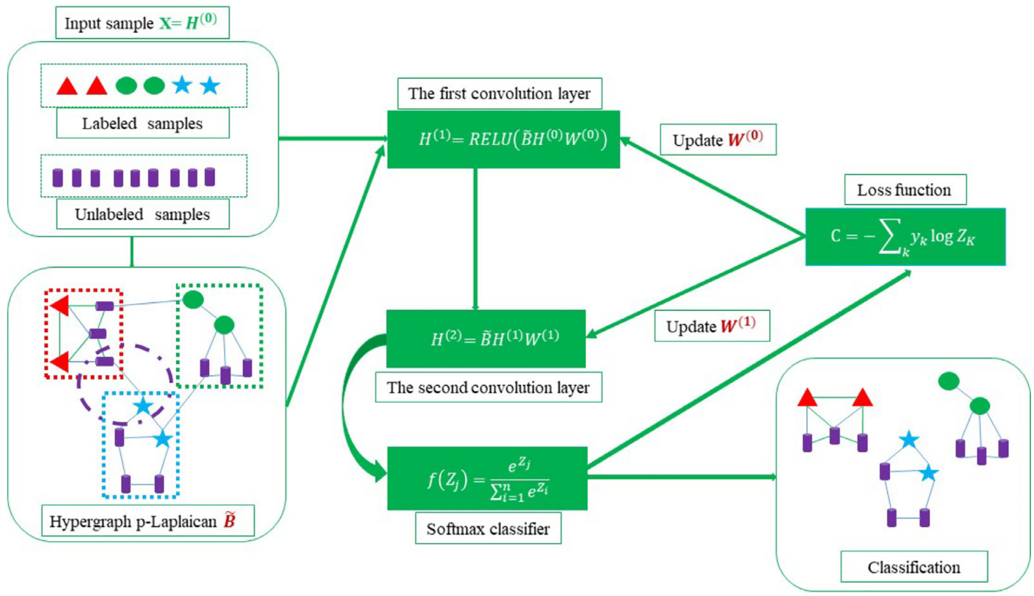

Fig. 2. Framework of the HpLapGCN model for semi-supervised classification.

Here, X is a matrix that represents the original data and $\theta \in \mathbb{R}^{\mathbb{C}\times\mathbb{F}}$. $\frac{2}{\lambda_{max}}HL_p - I_N$ is a symmetric matrix, which is the structure information of samples. Z is the output of each convolution layer, which represents the node's transformed sample features.

The above formula can also be expressed as the following expression, i.e.

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

4. Hypergraph *p*-Laplacian graph convolutional networks

In this part, we propose the hypergraph *p*-Laplacian graph convolutional networks (HpLapGCN) based on the new layer-wise aggregate rule. Specifically, we detailed describe the framework of a two-layer HpLapGCN. Fig. 2 shows the framework of a two-layer HpLapGCN for semi-supervised classification.

4.1. The first convolution layer of HpLapGCN

In this paper, we use a two-layer HpLapGCN to demonstrate the performance of the model. Firstly, we need to construct the structure information of the samples, i.e. $\widetilde{B} = \frac{2}{\lambda_{max}}HL_p - I_N$. Then, we can get the first convolution layer of HpLapGCN, i.e.

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

$\bar{W}^{(0)}$ are the weight parameters in the first layer. $\bar{H}^{(1)}$ is the output matrix in the first convolution layer. In the first layer, it can extract the first layer's sample features by fusing the hypergraph *p*-Laplacian structure information and original feature information.

4.2. The second convolution layer of HpLapGCN

The second convolution layer has a similar process. We take the output of last layer as the second layer's input. Then, the expression of the second layer's network is as follows, i.e.

$$\bar{H}^{(2)} = \widetilde{B}\bar{H}^{(1)}\bar{W}^{(1)} \tag{13}$$

$\bar{W}^{(1)}$ are the second layer's weight parameters. $\bar{H}^{(2)}$ is the output of the final layer. In this layer, the structure information is embedded in the first layer's sample features.

4.3. The classification layer of HpLapGCN

After two convolutional layers, we put the final sample features into the classifier. Currently, Softmax is a commonly used classifier in deep learning. The Softmax classifier generalizes Logistic regression to multi-class classification. The Softmax function can be defined as

$$f(Z_j) = \frac{e^{Z_j}}{\sum_{i=1}^{n} e^{Z_i}} \tag{14}$$

It can convert the sample features of each class to the probability that belong to each class by the Softmax function. Z is the final extracted sample features, i.e. $H^{(2)}$. Moreover, we use cross entropy loss function to update training parameters.

$$C = -\sum_k y_k \log Z_k \tag{15}$$

Here, y_k represents the true label information. Z_K is the output of the Softmax function. During the training process of HpLapGCN, our proposed model will stop training model until the cross entropy loss value C of validation set are stable. In addition, we only use a part of true label information y_k during this process. The model can get the best parameters by reducing the value of the loss function.

5. Experiments

In this section, we utilize extensive public datasets including Citeseer [53] and Cora [54] to test our proposed HpLapGCN for semi-supervised classification. In the first place, we give a brief description of the Citeseer and Cora datasets. In the next place, we introduce the parameters setting in the experiment. Finally, we show the experiment results of HpLapGCN.

5.1. Experiment datasets

The Citeseer database [53] contains 3327 scientific books. All scientific publications are composed of 3703 different words, which express the existence or absence of a corresponding word through one or zero. The total dataset is divided into six categories, such as Agents, AI, DB, IR, ML and HCI. All books contain 4732 citation relationships.

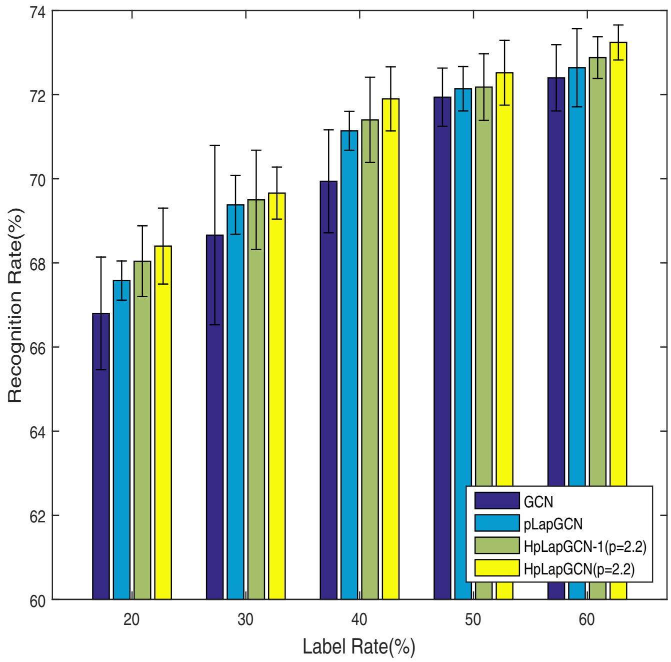

Fig. 3. Classification accuracy for all class of Citeseer database.

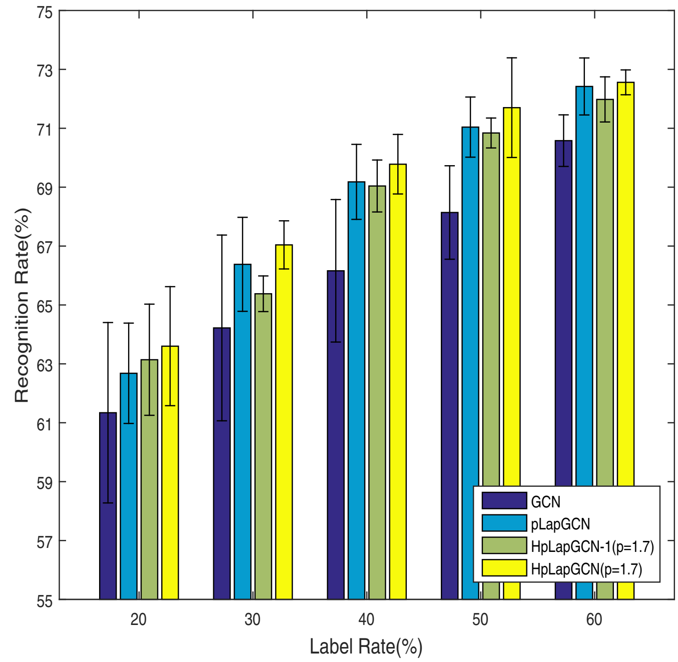

Fig. 4. Classification accuracy for all class of Cora database.

The Cora dataset [54] consists of totally 2708 publications, which are collected from seven classes including case-based, genetic-algorithms, neural-networks, probabilistic-methods, reinforcement-learning, rule-learning and theory. Each publication has 1433 different words. The dataset is divided into 5429 citation links.

5.2. Parameters setting

In the process of the experiment, we select 500 samples to form validation set, 1000 samples as testing set and the remaining samples are training set. Moreover, the validation set and testing set are all labeled data. In the training set, we random choose a certain percentage samples including 20%, 30%, 40%, 50%, 60% as labeled data, the rest training set as unlabeled data.

The maximum model training iteration numbers of HpLapGCN up to 200 by using the Adam [55] optimization method with a learning ration of 0.01. We will stop updating model parameters if the cross entropy loss value of validation set remain unchanged for consecutive ten times. To increase the generalization ability of our proposed model, we use the L2 regularization with the coefficient of 5×10^{-4} and Xavier [56] weight initialization method. Other parameters are as follows: Citeseer and Cora: 0.5 (dropout rate) and 32 (hidden neurons).

5.3. Experiment results

In this section, we compare the proposed HpLapGCN model with the GCN and pLapGCN, In addition, the pLapGCN is our previous works. In this work, we proposed a p-Laplacian graph convolutional networks (pLapGCN) for citation network classification by only utilizing the p-Laplacian to express the local structure information of simple graph structured data, not a hypergraph structured data. In each figure, the x-axis represents the label rate of training data. The y-axis is the total classification accuracy of each database in Figs. 3 and 4. Fig. 5 shows the classification accuracy of Citeseer dataset on each category including Agents, AI, DB, IR, ML and HCI. For Cora dataset, the classification accuracy results of three models on each class (case-based, genetic-algorithms, neural-networks, probabilistic-methods, reinforcement-learning, rule-learning and theory) are shown in Fig. 6.

Table 1
Comparison of the different algorithms.

Method	Citeseer (120)	Cora(140)
ManiReg	60.1	59.5
SemiEmb	59.6	59
Chebyshev($K = 2$)	53.6	49.8
Chebyshev($K = 3$)	53.7	50.5
MLP	46.5	55.1
HpLapGCN	62.5	59.8

As shown in Fig. 3, we can see that, the HpLapGCN with p=2.2 achieves the best performance compared to other models. In addition, with the label rate growing up, the classification accuracy of models also are on the increase. Fig. 5 reveals that under most circumstances, our proposed model outperforms other methods in the performance of each class.

Fig. 4 illustrates the classification accuracy of all categories in Cora database. From Fig. 4, we can observe that the HpLapGCN performs better than GCN and pLapGCN when p=1.7. For the most part, we can find that our proposed method also obtains higher recognition results from Fig. 6.

From the experiment results of Figs. 3 and 4, it suggests that HpLapGCN can extract the richer data features to increase the classification accuracy because it fuses the hypergraph p-Laplacian-based structure information, i,e, hypergraph p-Laplacian has the superiority to express the manifold structure of data compare with p-Laplacian and Laplacian. In addition, it also proves that the effectiveness of optimization method that our proposed, i.e. from HpLapGCN-1 to HpLapGCN.

To show the effectiveness of our proposed HpLapGCN model, we further compare HpLapGCN with other semi-supervised learning algorithms including manifold regularization (ManiReg) [57], semi-supervised embedding (SemiEmb) [58], Chebyshev ($K = 2$) [24], Chebyshev ($K = 3$) [24], multi-layer perceptron (MLP) [27]. Report numbers represent the mean classification accuracy of one hundred runs randomly in percent. From Table 1, we can see that

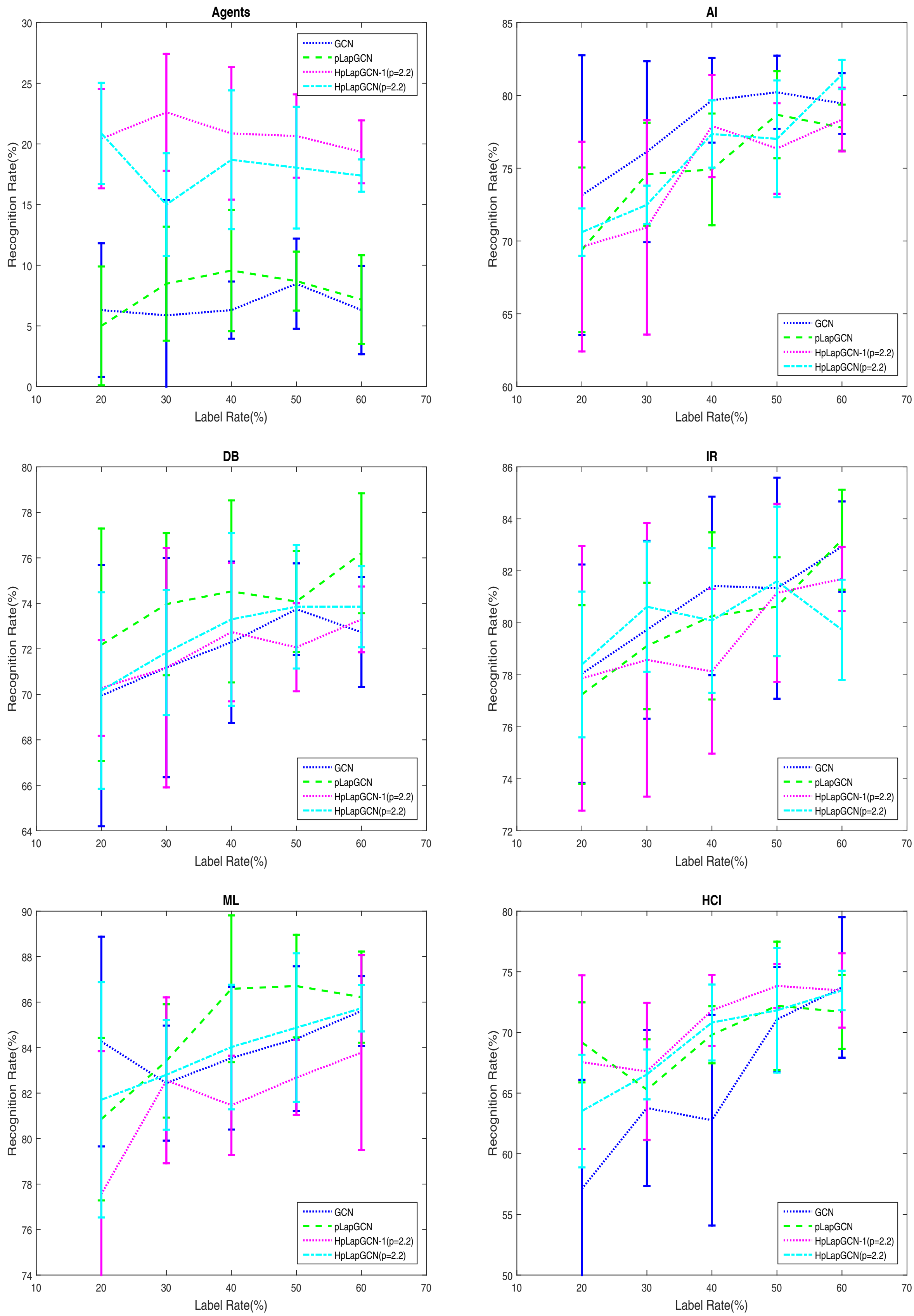

Fig. 5. Classification accuracy for each class of Citeseer database, including Agents, AI, DB, IR, ML, HCI. Each subfigure corresponds on single class.

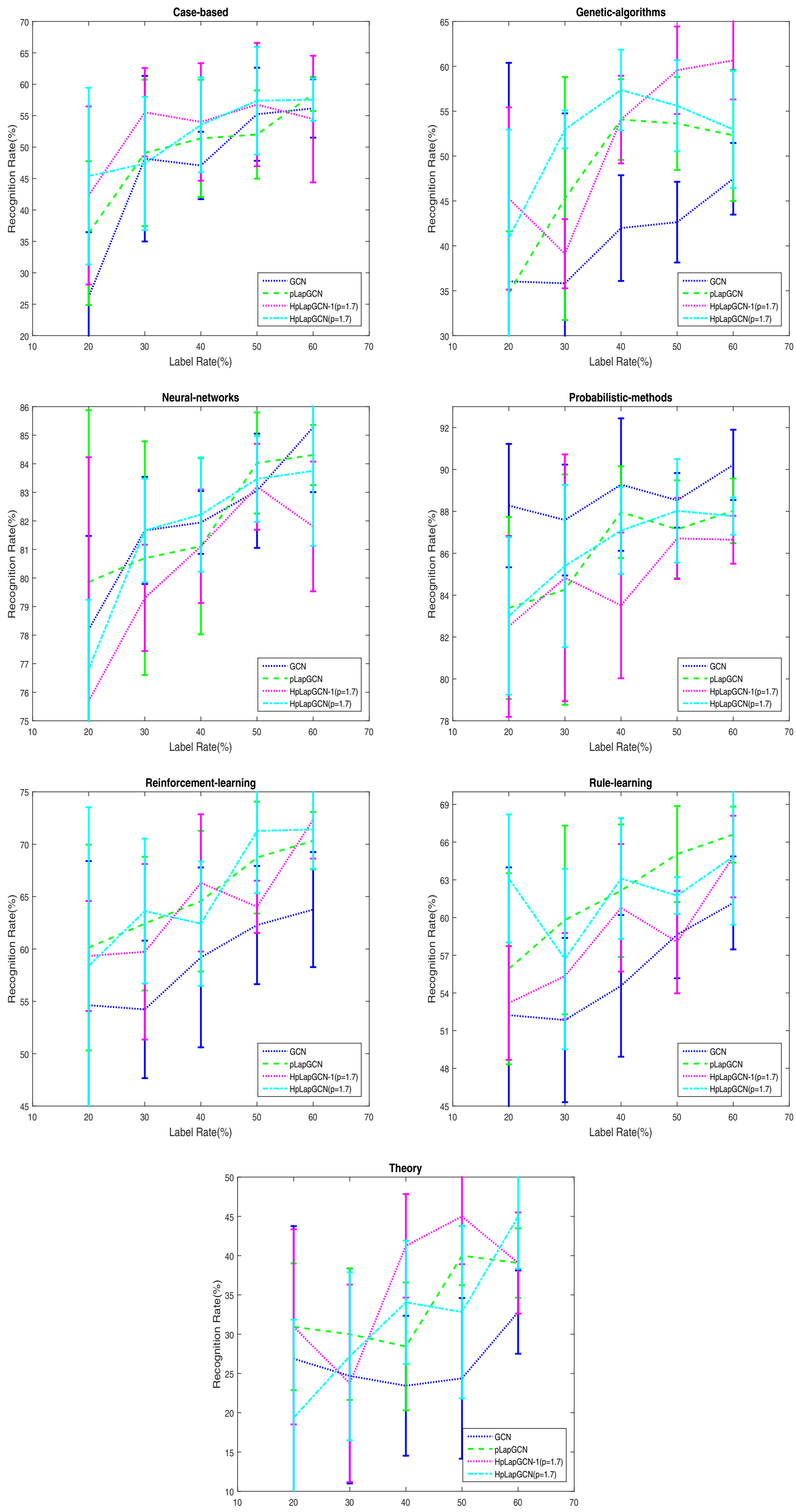

Fig. 6. Classification accuracy for each class of Cora database, including Theory, Case-based, Genetic-algorithms, Neural-networks, Probabilistic-methods, Reinforcement-learning, Rule-learning. Each subfigure corresponds on single class.

HpLapGCN achieves the better classification performance. Those experiment results also demonstrate that hypergraph p-Laplacian can better express the space manifold structure of the data.

6. Conclusion

In the past few years, many ML methods have been successful applied to the feature extraction of the complex space structured data. However, how to preserve the manifold structure of data and then extract the abundant local structure information is a vital problem. To exactly express the geometry structure of data, in this paper, we exploit the hypergraph p-Laplacian matrix to express the space manifold structure of the data. Then we propose a new expression method of structure information by optimizing the hypergraph p-Laplacian-based spectral graph convolutions. Finally, we propose the HpLapGCN model based on the optimal one-order polynomial in the hypergraph p-Laplacian. Hypergraph full considers the many pairwise relationships of data, thus HpLapGCN model can learn richer data features by integrating the feature information with the hypergraph p-Laplacian-based structure information. Extensive experimental results demonstrate the proposed HpLapGCN gets a higher classification performance.

Declaration of conflicting interests

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.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61671480, in part by the Fundamental Research Funds for the Central Universities, China University of Petroleum (East China) under Grant 18CX07011A and YCX2019080, in part by the Macau Science and Technology Development Fund under Grant FDCT/189/2017/A3, and by the Research Committee at University of Macau under Grants MYRG2016-00123-FST and MYRG2018-00136-FST.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.neucom.2019.06.068.

References

[1] C. Hong, J. Yu, J. Wan, D. Tao, M. Wang, Multimodal deep autoencoder for human pose recovery, IEEE Trans. Image Process. 24 (12) (2015) 5659–5670.
[2] J. Yu, Z. Kuang, B. Zhang, W. Zhang, D. Lin, J. Fan, Leveraging content sensitiveness and user trustworthiness to recommend fine-grained privacy settings for social image sharing, IEEE Trans. Inf. Forensics Secur. 13 (5) (2018) 1317–1332.
[3] C. Hong, J. Yu, J. Zhang, X. Jin, K.-H. Lee, Multi-modal face pose estimation with multi-task manifold deep learning, IEEE Trans. Ind. Inform. (2018), doi:10.1109/TII.2018.2884211.
[4] C. Hong, J. Yu, D. Tao, M. Wang, Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval, IEEE Trans. Ind. Electron. 62 (6) (2014) 3742–3751.
[5] J. Zhang, J. Yu, D. Tao, Local deep-feature alignment for unsupervised dimension reduction, IEEE Trans. Image Process. 27 (5) (2018) 2420–2432.
[6] J. Yu, D. Tao, M. Wang, Y. Rui, Learning to rank using user clicks and visual features for image retrieval, IEEE Trans. Cybern. 45 (4) (2014) 767–779.
[7] J. Yu, Y. Rui, D. Tao, Click prediction for web image reranking using multimodal sparse coding, IEEE Trans. Image Process. 23 (5) (2014) 2019–2032.
[8] S. Kearnes, K. Mccloskey, M. Berndl, V. Pande, P. Riley, Molecular graph convolutions: moving beyond fingerprints, J. Comput.-aided Mol. Des. 30 (8) (2016) 1–14.
[9] S.A. Rios, F. Aguilera, J.D. Nuñez-Gonzalez, M. Graña, Semantically enhanced network analysis for influencer identification in online social networks, Neurocomputing 326 (2019) 71–81.
[10] J. Yu, C. Zhu, J. Zhang, Q. Huang, D. Tao, Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition, IEEE Trans. Neural Netw. Learn. Syst. (2019), doi:10.1109/TNNLS.2019.2908982.
[11] J. Yu, C. Hong, Y. Rui, D. Tao, Multitask autoencoder model for recovering human poses, IEEE Trans. Ind. Electron. 65 (6) (2017) 5060–5068.
[12] M.T. Mills, N.G. Bourbakis, Graph-based methods for natural language processing and understanding survey and analysis, IEEE Trans. Syst. Man Cybern. Syst. 44 (1) (2013) 59–71.
[13] A.-M. Rassinoux, R.H. Baud, C. Lovis, J.C. Wagner, J.-R. Scherrer, Tuning up conceptual graph representation for multilingual natural language processing in medicine, in: International Conference on Conceptual Structures, 1998, pp. 390–397.
[14] R. Mihalcea, D. Radev, Graph-Based Natural Language Processing and Information Retrieval, Cambridge University Press, 2011.
[15] B. Wu, Y. Liu, B. Lang, L. Huang, Dgcnn: disordered graph convolutional neural network based on the gaussian mixture model, Neurocomputing 321 (2018) 346–356.
[16] H. Yuan, J. Li, L.L. Lai, Y.Y. Tang, Graph-based multiple rank regression for image classification, Neurocomputing 315 (2018) 394–404.
[17] K. Zeng, J. Yu, C. Li, J. You, T. Jin, Image clustering by hyper-graph regularized non-negative matrix factorization, Neurocomputing 138 (2014) 209–217.
[18] J. Yu, B. Zhang, Z. Kuang, D. Lin, J. Fan, Iprivacy: image privacy protection by identifying sensitive objects via deep multi-task learning, IEEE Trans. Inf. Forensics Secur. 12 (5) (2016) 1005–1016.
[19] J. Yu, X. Yang, F. Gao, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. 47 (12) (2016) 4014–4024.
[20] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[21] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 855–864.
[22] C. Tu, H. Liu, Z. Liu, M. Sun, Cane: context-aware network embedding for relation modeling, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 1, 2017, pp. 1722–1731.
[23] D. Wang, P. Cui, W. Zhu, Structural deep network embedding, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1225–1234.
[24] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems, 2016, pp. 3844–3852.
[25] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral networks and locally connected networks on graphs, International Conference on Learning Representations, 2013.
[26] N. Khan, U. Chaudhuri, B. Banerjee, S. Chaudhuri, Graph convolutional network for multi-label vhr remote sensing scene recognition, Neurocomputing 357 (2019) 36–46.
[27] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: International Conference on Learning Representations, 2017.
[28] N. Yadati, M. Nimishakavi, P. Yadav, A. Louis, P. Talukdar, Hypergcn: hypergraph convolutional networks for semi-supervised classification, in: IEEE International Conference on Multimedia and Expo, 2019.
[29] C. Zhuang, Q. Ma, Dual graph convolutional networks for graph-based semi–supervised classification, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, 2018, pp. 499–508.
[30] J. Atwood, D. Towsley, Diffusion-convolutional neural networks, in: Advances in Neural Information Processing Systems, 2016, pp. 1993–2001.
[31] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.
[32] M. Niepert, M. Ahmed, K. Kutzkov, Learning convolutional neural networks for graphs, in: International Conference on Machine Learning, 2016, pp. 2014–2023.
[33] J. Ou, Y. Li, Vector-kernel convolutional neural networks, Neurocomputing 330 (2018) 253–258.
[34] S.P. Adhikari, C. Yang, K. Slot, M. Strzelecki, H. Kim, Hybrid no-propagation learning for multilayer neural networks, Neurocomputing 321 (2018) 28–35.
[35] A. Liu, Y. Laili, Balance gate controlled deep neural network, Neurocomputing 320 (2018) 183–194.
[36] W. Liu, X. Ma, Y. Zhou, D. Tao, J. Cheng, P-laplacian regularization for scene recognition, IEEE Trans. Cybern. 49 (8) (2018) 2927–2940.
[37] K.I. Kim, F. Steinke, M. Hein, Semi-supervised regression using hessian energy with an application to semi-supervised dimensionality reduction, in: Advances in Neural Information Processing Systems, 2009, pp. 979–987.
[38] X. Ma, W. Liu, S. Li, D. Tao, Y. Zhou, Hypergraph p-laplacian regularization for remotely sensed image recognition, IEEE Trans. Geosci. Remote Sens. 57 (3) (2019) 1585–1595.
[39] S. Saito, D.P. Mandic, H. Suzuki, Hypergraph p-laplacian: a differential geometry view, in: 32nd AAAI Conference on Artificial Intelligence, 2018.
[40] L.H. Tran, L.H. Tran, Un-normalized hypergraph p-laplacian based semi-supervised learning methods, 2018, arXiv preprint arXiv:1811.02986.
[41] Y. Huang, Q. Liu, S. Zhang, D.N. Metaxas, Image retrieval via probabilistic hypergraph ranking, in: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 3376–3383.
[42] J.Y. Zien, M.D. Schlag, P.K. Chan, Multilevel spectral hypergraph partitioning with arbitrary vertex sizes, IEEE Trans. Comput.-aided Des. Integr. Circuits Syst. 18 (9) (1999) 1389–1399.

[43] J.A. Rodriguez, On the laplacian spectrum and walk-regular hypergraphs, Linear Multilinear Algebra 51 (3) (2003) 285–297.
[44] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, S. Belongie, Beyond pairwise clustering, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2, 2005, pp. 838–845.
[45] A. Banerjee, I. Dhillon, J. Ghosh, S. Merugu, D.S. Modha, A generalized maximum entropy approach to bregman co-clustering and matrix approximation, J. Mach. Learn. Res. 8 (Aug) (2007) 1919–1986.
[46] O. Duchenne, F. Bach, I.-S. Kweon, J. Ponce, A tensor-based algorithm for high-order graph matching, IEEE Trans. Pattern Anal. Mach. Intell. 33 (12) (2011) 2383–2395.
[47] D. Zhou, J. Huang, B. Schölkopf, Learning with hypergraphs: clustering, classification, and embedding, in: Advances in Neural Information Processing Systems, 2007, pp. 1601–1608.
[48] M. Bolla, Spectra, euclidean representations and clusterings of hypergraphs, Discrete Math. 117 (1–3) (1993) 19–39.
[49] T. Bühler, M. Hein, Spectral clustering based on the graph p-laplacian, in: Proceedings of the 26th Annual International Conference on Machine Learning, 2009, pp. 81–88.
[50] D. Luo, H. Huang, C. Ding, F. Nie, On the eigenvectors of p-laplacian, Mach. Learn. 81 (1) (2010) 37–51.
[51] D. Zhou, B. Schölkopf, Regularization on discrete spaces, in: Joint Pattern Recognition Symposium, 2005, pp. 361–368.
[52] W. Liu, X. Yang, D. Tao, J. Cheng, Y. Tang, Multiview dimension reduction via hessian multiset canonical correlations, Inf. Fusion 41 (2017).
[53] H. Zhang, C.L. Giles, H.C. Foley, J. Yen, Probabilistic community discovery using hierarchical latent gaussian mixture model, in: 21st AAAI Conference on Artificial Intelligence, 7, 2007, pp. 663–668.
[54] W. Cohen, P. Ravikumar, S. Fienberg, A comparison of string metrics for matching names and records, in: Kdd workshop on Data Cleaning and Object Consolidation, 3, 2003, pp. 73–78.
[55] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: International Conference of Learning Representation, 2014.
[56] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics, 2010, pp. 249–256.
[57] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (Nov) (2006) 2399–2434.
[58] J. Weston, F. Ratle, H. Mobahi, R. Collobert, Deep learning via semi-supervised embedding, in: Neural Networks: Tricks of the Trade, Springer, 2012, pp. 639–655.

Sichao Fu is currently pursuing the master's degree with the College of Information and Control Engineering, China University of Petroleum, Qingdao, China. His research interests include pattern recognition and computer vision.

Weifeng Liu (M12CSM17) received the double B.S. degrees in automation and business administration and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2002 and 2007, respectively. He was a Visiting Scholar with the Centre for Quantum Computation and Intelligent Systems, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia, from 2011 to 2012. He is currently a Full Professor with the College of Information and Control Engineering, China University of Petroleum, Qingdao, China. He has authored or coauthored a dozen papers in top journals and prestigious conferences, including four Essential Science Indicators (ESI) highly cited papers and two ESI hot papers. His research interests include computer vision, pattern recognition, and machine learning. Prof. Liu serves as an Associate Editor for the Neural Processing Letters, the Co-Chair for the IEEE SMC Technical Committee on Cognitive Computing, and a Guest Editor for special issue of the Signal Processing, theIET Computer Vision, the Neurocomputing, and the Remote Sensing. He also serves over 20 journals and over 40 conferences.

Yicong Zhou (M07SM14) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA. He is currently an Associate Professor and the Director with the Vision and Image Processing Laboratory, Department of Computer and Information Science, University of Macau, Macau, China. His research interests include chaotic systems, multimedia security, image processing and understanding, and machine learning. Dr. Zhou was a recipient of the Third Price of Macau Natural Science Award in 2014. He served as an Associate Editor for the Neurocomputing, the Journal of Visual Communication and Image Representation, and the Signal Processing: Image Communication. He is a Co-Chair of Technical Committee on Cognitive Computing in the IEEE Systems, Man, and Cybernetics Society.

Liqiang Nie received the B.E. and Ph.D. degrees from Xian Jiaotong University, Xian, China, and National University of Singapore (NUS), Singapore, in 2009 and 2013, respectively. He is currently a Professor with the School of Computer Science and Technology, Shandong University, Jinan, China. He was with the NUS as a Research Fellow for more than three years. His research interests include information retrieval, multimedia computing, and their applications in the field of healthcare.