A Node-collaboration-informed Graph Convolutional Network for Precise Representation to Undirected Weighted Graphs

Ying Wang  
School of Computer Science and Technology, Chongqing University of Posts and Telecommunications  
Chongqing, 400065, China  
15032382660@163.com

Ye Yuan  
College of Computer and Information Science, Southwest University  
Chongqing, 400715, China  
yuanyekl@swu.edu.cn

Di Wu  
College of Computer and Information Science, Southwest University  
Chongqing, 400715, China  
wudi.cigit@gmail.com

Abstract—An undirected weighted graph (UWG) is frequently adopted to describe the interactions among a solo set of nodes from big data-connected applications like the user contact frequency from a social network services system. A graph convolutional network (GCN) is widely adopted to perform representation learning to a UWG for subsequent pattern analysis tasks such as clustering or missing data estimation. However, existing GCNs mostly neglects the latent collaborative information hidden in its connected node pairs. To address this issue, this study proposes to model the node collaborations via a symmetric latent factor analysis model, which is thus incorporated into a GCN model as a node-collaboration module for supplementing the collaboration loss. Based on this essential idea, a Node-collaboration-informed Graph Convolutional Network (NGCN) model is proposed with three-fold ingredients: a) Learning latent collaborative information from the interaction of node pairs via a node-collaboration module; b) Building the residual connection and weighted representation propagation to obtain high representation capacity; and c) Implementing the model optimization in an end-to-end fashion to achieve precise representation to the target UWG. Empirical studies on four UWGs emerging from real applications demonstrate that owing to its efficient modeling of node-collaborations, the proposed NGCN significantly outperforms state-of-the-art GCNs in addressing the task of missing weight estimation. Meanwhile, its high scalability ensures its compatibility with more advanced GCN extensions, which will be further investigated in our future studies.

Keywords—Undirected Weighted Graph, Collaborative Information, Graph Convolutional Network, Weight Estimation

I. INTRODUCTION

Recent years have witnessed the great success of graph convolutional network (GCN) [1] in representation learning for graph-structure data [22, 36, 38]. Owing to its effectiveness in graph representation learning, a GCN has been applied to solve a plethora of real-world problems competently. For instance, James [2] proposes a novel spatial-temporal GCN, which adopts a spatial-temporal attention mechanism for traffic flow forecasting effectively. Yu et al. [3] propose an adversarial GCN, which adopts adversarial training for precise recommendation. Azadifar et al. [4] utilize a deep GCN for gene essentiality prediction.

An undirected weighted graph (UWG) is frequently encountered graph-structure data in real applications. In particular, a UWG assigns a quantifiable index to the interaction between two nodes as its weight, which represents the correlation or strength of the interaction between a pair of nodes. For instance, the weight can describe the citation counts between authors [5] in citation network, the confidence of interactome between proteins in a protein network [6, 7], and the contact rate between users in a social network [8]. From this point of view, we clearly see that node-node interactions in a UWG contain massive useful knowledge to help us better understand inter-relationship among nodes. However, when addressing a UWG, a GCN heavily relies on node features without considering the latent collaborative information hidden in its connected node pairs, which is crucial for missing weight estimation [9]. Therefore, present empirical GCN show poor performance in missing weight estimation of a UWG, and is even outperformed by a linear representation learning model [10]. In line with the aforementioned discoveries, we have encountered the research question:

**RQ.** Is it possible to incorporate collaborative information learned from the interaction of node pairs, thereby improving the GCN’s performance on missing weight estimation?

---

This research is supported in part by the National Natural Science Foundation of China under grants 62002337, in part by the Chongqing Natural Science Foundation under grant CSTB2022NSCQ-MSX1486, and in part by the Chongqing Natural Science Foundation under grant cstc2020jcyj-msxmX0792. (Corresponding author: Ye Yuan)

To answer this question, this study proposes a Node-collaboration-informed Graph Convolutional Network (NGCN), which can exploit the latent collaborative
information from the interactions of node pairs effectively. Specifically, it consists of the following three-fold ideas:

a) Learning latent collaborative information from the interaction of node pairs via a node-collaboration module;

b) Building the residual connection and weighted representation propagation to obtain high representation capacity; and

c) Implementing the model optimization in an end-to-end fashion to achieve precise representation to the target UWG.

This paper mainly contributes in the following perspectives:

a) We propose an NGCN model. Different from popular GCNs that rely solely on smoothed node features, it is able to incorporate learnable collaborative information hidden in the interactions of node pairs to enhance the representation ability on a UWG;

b) The proposed NGCN are comprehensively investigated on extensive benchmark datasets demonstrate it outperforms the state-of-the-art baselines in missing weight estimation.

Section II states the preliminaries. Section III presents the proposed NGCN. Section IV conducts the empirical studies. Section V is the related work. Finally, Section VI concludes this paper.

II. PRELIMINARIES

A. Problem Formulation

Note that a UWG is defined as [11]:

Definition 1: Let \( V = \{v_i \mid i = 1, 2, \ldots, N \} \) represent a set of \( N \) nodes, \( E = \{e_{ij} \mid v_i, v_j \in V \} \) represent a set of edges where the nodes \( v_i, v_j \in V \) are connected. In particular, for a UWG \( G=(V, E) \), assigns a quantifiable index to the edge between two nodes as its weight. Hence, it is denoted the adjacency matrix \( A = [a_{ij}] \in \mathbb{R}^{N \times N} \), where \( a_{ij} \) is the non-zero weight if \( e_{ij} \in E \) and zero otherwise. Each node is described by a feature vector \( x_i \in \mathbb{R}^d \), where \( X \in \mathbb{R}^{N \times d} \) is the collection of such vectors in a matrix form.

The missing weight estimation for a UWG is defined as [12]:

Definition 2: Given a UWG \( G=(V, E) \), a representation learning model learns meaningful representation \( \hat{h}_i \in \mathbb{R}^d \) for each node \( i \) by an iterative learning process, where \( H \in \mathbb{R}^{N \times d} \) is the collection of such vectors in a matrix form. Subsequently, \( H \) is used for predicting the link weight of each link \((i, j)\) as follows:

\[
\hat{d}_{ij} = \langle \hat{h}_i, \hat{h}_j \rangle,
\]

where \( \langle \cdot, \cdot \rangle \) denotes the inner product of two vectors, \( \hat{d}_{ij} \) denotes the predicted weight. Note that we estimate the missing weight rather than the probability of link between two nodes. Thus, the non-linear activation function is omitted here [13].

B. Graph Convolutional Network

As described in prior studies [14, 15], a GCN learns low dimensional representations of nodes via aggregation of features from neighbors using non-linear transformations. Its forward propagation for node representation is defined as:

\[
H_{i}^{l+1} = \sigma(D^{-0.5} (A+I) D^{-0.5} H^{l} W^{l})
\]

(2)

where \( H_0 \) is the node feature matrix \( X \), \( H_0 \) denotes the representation of nodes at \( l \)-th layer, \( H_{i}^{l} \) denotes the output representation after stacking \( l \) GCN layers, \( H_{i}^{l} \in \mathbb{R}^{d_{l}} \) is a trainable weight matrix, \( I \) denotes the identity matrix, \( D \) denotes the degree matrix of \((A + I) \), \( \sigma(\cdot) \) denotes the activation function.

III. THE PROPOSED NGCN MODEL

We now introduce the proposed NGCN model, whose framework for missing weight estimation is illustrated in Fig. 2. It consists of the following three main components:

a) The residual and weighted representation propagation module employs the residual connection and weighted representation propagation into GCN to offer better node representation capacity;

b) The collaborative information learning module extracts the useful collaborative information from the interactions of node pairs via a symmetric latent factor analysis model; and

c) The estimation module combines collaborative information and node representation to estimate the missing weight.

A. Weighted Representation Propagation Module

Given a UWG in Definition 1, this module takes the node feature matrix \( X \) and weighted adjacency matrix \( A \) as the input. Following with the forward propagation process in (2), we have that:

\[
H_0 = X,
\]

(3a)

\[
\hat{H}_{i}^{(l+1)} = \text{ReLU} \left( D^{-0.5} A D^{-0.5} H_{i}^{(l)} W^{(l)} \right),
\]

(3b)

\[
H_{i}^{(l+1)} = \hat{H}_{i}^{(l+1)} + H_{i}^{(l)}
\]

(3c)

where we adopt ReLU as the activation function in this module, \( \hat{H}_{i}^{(l+1)} \) denotes the output of \( l \)-th layer, \( H_{i}^{(l+1)} \) and \( H_{i}^{(l)} \) denotes the input of \((l+1)\)-th and \( l \)-th layer, respectively.

Note that in (3c), the residual connection is adopted by combining the input and output of the \( l \)-th layer to obtain the input representation of \((l+1)\)-th layer. It can facilitate shallow layer’s feature reuse and gradient back-propagation, thereby offering better representation capacity. Moreover, this operation considers the effect of self-connection. Hence, the self-connection in (3b) is neglected to avoid the numerical instabilities of representation scale.

In order to better understanding (3a)-(3c), we reformulated them into a vector form. Thus, for node \( i \), we have that:

\[
\hat{h}_{i}^{(0)} = x_{i},
\]

(4a)

\[
\hat{h}_{i}^{(l+1)} = \text{ReLU} \left( \sum_{j \in N(i)} \frac{1}{\sqrt{\hat{d}_{ij} d_{ij}}} h_{j}^{(l)} W^{(l)} \right),
\]

(4b)

\[
h_{i}^{(l+1)} = \hat{h}_{i}^{(l+1)} + h_{i}^{(l)},
\]

(4c)
where $N(i)$ denotes the directly-connected neighbor set of node $i$, $d_i$ is the weighted diagonal degree of node $i$, $\bar{h}_i^{(1)}$, $h_i^{(1)}$ and $h_i^l$ denote the $i$-th row vector corresponding to $\bar{H}^{(1)}$, $H^{(1)}$ and $H_l$, respectively. Note that the task of this study is missing weight estimation. Hence, (4a)-(4c) denotes a weighted representation propagation process, which is different from that of traditional GCN designed for node classification and link estimation in the following aspects:

- For (4a), the node feature $x_i$ is randomly initialized vector and needs to be trained; and
- For (4b), $d_i$ denotes sum of weights between node $i$ and its directly-connected neighbors instead of computing its number of neighbor $|N(i)|$.

### B. Collaborative Information Learning Module

As aforementioned, the proposed NGCN is able to make use of the latent collaborative information from the interactions of node pair, which can be learned outside of GCN. Next, we propose a simple but effective way for NGCN to capture latent collaborative information.

Let $S=[s_{ij}] \in \mathbb{R}^{N \times N}$ denote the collaborative information matrix, where its element $s_{ij}$ is the collaborative information between nodes $i$ and $j$ in a UWG. It can be achieved with the following generalized generating function:

$$s_{ij} = \arg \min_{\Phi(\cdot)} \Phi\left( a_{ij}, \Psi\left( S_{\psi}\right) \right),$$

where $\Phi(\cdot)$ denotes a distance metric, $\Psi(\cdot)$ denotes an operator to construct the approximation of $A=[a_{ij}]$.

In this study, we adopt symmetric latent factor analysis (SLFA) [10, 31, 39] to achieve this purpose. The main reason is that SLFA can extract latent collaborative information from the interactions of node pair accurately since it considers the intrinsic symmetry property of the target UWG. Following the principle of SLFA, let $\Phi(\cdot)$ be the Euclidean distance-based metric and $\Psi(S_{\psi})=(YY^T)_y$, we have that:

$$s_{ij} = \arg \min_{\Phi(\cdot)} \left( a_{ij} - (YY^T)_y \right)^2,$$

where we use a latent matrix $Y \in \mathbb{R}^{N \times d}$ to approximate $S$ to reduce the computational burden, $d$ denotes the latent dimension of $Y$. Note that $s_{ij}$ is encoded via $(YY^T)_y$ and it indicates that a higher collaboration level $s_{ij}$ learned by (6) means that nodes $i$ and $j$ are more similar. It could lead to better node representation in GCN.

### C. Estimation Module

Based on the inferences given in Sections A and B, NGCN combines collaborative information and node representation to estimate the missing weight. As shown in Fig. 2, NGCN computes the link weight $\hat{a}_{ij}$ as follows:

$$\hat{a}_{ij} = \beta \cdot \langle h_i, h_j \rangle + (1-\beta) \cdot \langle y_i, y_j \rangle,$$

where $y_i$ and $y_j$ denote the $i$-th and $j$-th row vector of $Y$, $\omega$ is trainable parameter to balance the importance of collaborative information and node representation.

In order to inject the collaborative information in an end-to-end fashion, we jointly train NGCN by considering the loss of each modules. In this study, we utilize the commonly-adopted Euclidean distance to design the objective function:

$$L = \sum_{a_{ij} \in A} \left( a_{ij} - \hat{a}_{ij} \right)^2 + \sum_{a_{ij} \in A} \left( a_{ij} - \langle h_i, h_j \rangle \right)^2 + \sum_{a_{ij} \in A} \left( a_{ij} - \langle y_i, y_j \rangle \right)^2 + \lambda \|\Theta\|_2^2,$$

where $A$ denotes the training dataset, $\lambda$ denotes the regularization coefficient to control the $L_2$ regularization strength to prevent overfitting, $\Theta=\{\omega, X, Y, \{W_{i,j}\}\}$ denotes all trainable model parameters. We adopt mini-batch Adam to optimize the estimation model and update the model parameters.

### IV. EMPIRICAL STUDIES

#### A. General Settings

**Evaluation Protocol.** This paper concerns the missing weight estimation for a UWG. Hence, we adopt the estimation accuracy as the evaluation protocol [16-18, 37]. Commonly, root mean squared error (RMSE) and mean absolute error (MAE) [20, 21, 23, 28, 32-34] are commonly adopted to measure a model’s estimation accuracy:
\[
\text{RMSE} = \sqrt{\frac{\sum_{\forall y \in \text{pl}} a_{\text{y}} - \hat{a}_{\text{y}}^2}{|\text{pl}|}}, \\
\text{MAE} = \left(\frac{\sum_{\forall y \in \text{pl}} |a_{\text{y}} - \hat{a}_{\text{y}}|}{|\text{pl}|}\right).
\]

where $|\cdot|$ calculates the cardinality of an enclosed set, $|\cdot|_{\text{abs}}$ denotes the absolute value of an enclosed number, and $\Gamma$ denotes testing dataset.

**Datasets.** Four UWG datasets [19] are adopted in our experiments, whose details are summarized in Table I. Note that D1 and D3 record hundred plant species leaves classified based on margin and texture respectively, D2 and D4 are about agricultural land-use type in Indiana and speech recognition on Japanese vowels respectively. And we randomly split the known edges set of each UWG dataset into ten disjoint and equally-sized subsets, where seven subsets are chosen as the training set, one as the validation set, and the remaining two as the testing set for 70%-10%-20% train-validation-test setting. The above process is sequentially repeated five times for five-fold cross-validation.

**Baselines.** To demonstrate the effectiveness, we compare our proposed NGCN with the state-of-the-art models for missing weight estimation. Table II records all the compared models.

| No. | Name               | Description                  |
|-----|--------------------|------------------------------|
| M1  | MF                 | A matrix factorization-based model [24]. |
| M2  | NeuMF              | A deep neural network-based MF model [25]. |
| M3  | GCN                | A standard graph convolutional network [1]. |
| M4  | LR-GCCF            | A linear residual GCN [26]. |
| M5  | DGCN HN            | A deep GCN with hybrid normalization [27]. |
| M6  | GCMC               | A graph auto-encoder network [29]. |
| M7  | LightGCN           | A Simplifying GCN [30]. |
| M8  | NGCN               | The proposed model of this study. |

**Training Settings.** For achieving the objective results, following general settings are applied to all involved models:

- **a)** All the compared models are deployed on a GPU platform with two NVIDIA GeForce RTX 3090 GPU cards;
- **b)** We adopt an Adam optimizer and the batch-size is set as 2048;
- **c)** The termination conditions are consistent for all the compared models, i.e., the iteration threshold is 1000, or the training will be terminated after the minimum error of 30 iterations;
- **d)** The node feature matrix of each model are initialized with the same randomly generated arrays, and the dimension of node feature $f=128$;
- **e)** For the proposed NGCN, the latent matrix’s latent dimension $d=128$, the number of propagation layers $L=2$;
- **f)** For all the compared models, we apply a grid search for the learning rate $\eta=$\{0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05\} and $L_2$ regularization coefficient $\lambda=$\{0.00001, 0.00005, 0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1\} to achieve the optimal results.

**B. Comparison Performance**

We start by comparing the performance of all the methods. Figure 3 and Table III show the comparison results on RMSE/MAE. From them, we achieve some important verdicts:

- **a)** NGCN possesses remarkable advantage in missing weight estimation of a UWG. As depicted in Table III and Fig. 3, M8, i.e., the proposed NGCN, achieves the lowest estimation errors on seven testing cases out of eight in total. For instance, on D1, M8 achieves the optimal RMSE at 0.03516, which is 14.63% lower than 0.04119 achieved by M1, 8.03% lower than 0.03823 achieved by M2, 8.96% lower than 0.03862 achieved by M3,9.68% lower than 0.03893 achieved by M4, 10.64% lower than 0.03985 achieved by M5, 7.30% lower than 0.03793 achieved by M6, and 8.65% lower than 0.03849 achieved by M7. Similar outcomes are also found on MAE. The situation is a little different on MAE of D4. For this case, M8 is only slightly outperformed by M7. One possible interpretation is that D4 contains valuable linear characteristics. In general, M8’s representation ability to the target UWG is impressive.
TABLE III. THE COMPARISON RESULTS ON RMSE/MAE, WHERE ✪ INDICATES THAT NGCN IS OUTPERFORMED BY THE COMPARED MODELS.

| Case | M1     | M2     | M3     | M4     | M5     | M6     | M7     | M8     |
|------|--------|--------|--------|--------|--------|--------|--------|--------|
| D1   | RMSE   | 0.04119±8E-4 | 0.03823±6E-4 | 0.03862±3E-4 | 0.03893±6E-4 | 0.03985±7E-4 | 0.03793±6E-4 | 0.03849±8E-4 | 0.03516±1E-4 |
|      | MAE    | 0.02301±8E-4 | 0.02235±7E-4 | 0.02336±8E-4 | 0.02269±8E-4 | 0.02170±9E-4 | 0.02221±6E-4 | 0.02171±6E-4 | 0.02052±4E-5 |
| D2   | RMSE   | 0.03710±8E-4 | 0.03366±7E-4 | 0.03417±1E-4 | 0.03603±9E-4 | 0.03581±1E-5 | 0.03355±6E-5 | 0.03471±1E-5 | 0.03207±9E-5 |
|      | MAE    | 0.02054±1E-4 | 0.01864±2E-4 | 0.01902±2E-4 | 0.01987±5E-5 | 0.01916±6E-4 | 0.01887±4E-4 | 0.01906±5E-4 | 0.01780±6E-5 |
| D3   | RMSE   | 0.07564±4E-4 | 0.06987±4E-4 | 0.06928±6E-4 | 0.07450±6E-4 | 0.07495±2E-4 | 0.07092±4E-4 | 0.07093±3E-4 | 0.06225±8E-4 |
|      | MAE    | 0.04135±4E-4 | 0.04127±2E-4 | 0.04153±5E-4 | 0.04279±4E-4 | 0.03942±5E-4 | 0.04140±4E-4 | 0.03932±2E-4 | 0.03475±1E-4 |
| D4   | RMSE   | 0.12687±7E-4 | 0.11633±4E-4 | 0.11857±1E-4 | 0.12447±2E-4 | 0.12253±2E-4 | 0.11877±3E-4 | 0.11845±5E-4 | 0.11449±1E-4 |
|      | MAE    | 0.06240±2E-4 | 0.05845±6E-4 | 0.06369±7E-4 | 0.06685±1E-4 | 0.05870±2E-5 | 0.06583±4E-4 | 0.05880±2E-5 | 0.03847±4E-4 |

| Loss/Win | 0/8 | 1/7 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 | 0/8 |

b) Compared with the MF-based models, i.e., M1 and M2, M8 show significant accuracy improvements. The main reason is that M8 incorporates the information of neighbors by the weighted representation forward propagation process, thereby improving the representation learning to the target UWG. Moreover, compared with the GCN-based models, i.e., M3-M7, M8 not only utilizes the information of neighbors, but also captures collaborative information from the interactions of node pairs, which is crucial for missing weight estimation. Hence, M8 still outperforms them when estimating the missing weight.

C. Ablation Study

We present ablation experiments to verify the benefits of NGCN’s components. First, we evaluate NGCN without SLFA and GCN. Then we examine the effectiveness of residual connection by comparing with a standard GCN, i.e., NGCN’s components. First, we evaluate NGCN without C.

NGCN without GCN or SLFA. We clearly find that NGCN outperforms the two ablative cases, i.e., NGCN (w/o GCN) and NGCN (w/o SLFA), in terms of RMSE and MAE. For instance, as shown in Table IV, on D2, NGCN achieves the lowest RMSE at 0.03207, which is 5.26% lower than 0.03385 achieved by NGCN (w/o GCN), 2.85% lower than 0.03301 achieved by NGCN (w/o SLFA). Note that NGCN (w/o GCN) only uses latent collaborative information of node pairs, and NGCN (w/o SLFA) only relies on node features. This shows that these two components are both positive for missing weight estimation. Moreover, NGCN (w/o GCN) has lower estimation errors than that of NGCN (w/o SLFA). It demonstrates that utilizing key latent collaborative information of node pairs is crucial for missing weight estimation.

Effectiveness of residual connection. As shown in Table IV, even without latent collaborative information of node pairs, NGCN (w/o SLFA) still outperforms a standard GCN. For instance, on D1, NGCN (w/o SLFA) achieves the lowest RMSE at 0.03565, which is 7.69% lower than 0.03862 by a standard GCN. The main reason is that the residual connection is equivalent to concatenate all the layer’s node representation, which can facilitate shallow layer’s feature reuse and gradient back-propagation, thereby offering better representation capacity.

TABLE IV. ABLATION STUDY ANALYZING THE SIGNIFICANCE OF NGCN.

| Case | NGCN (Holomorph) | NGCN (w/o GCN) | NGCN (w/o SLFA) | GCN (Standard) |
|------|------------------|----------------|-----------------|----------------|
| D1   | RMSE             | 0.03516        | 0.03673         | 0.03565        | 0.03862        |
|      | MAE              | 0.02052        | 0.02159         | 0.02155        | 0.02336        |
| D2   | RMSE             | 0.03207        | 0.03385         | 0.03301        | 0.03417        |
|      | MAE              | 0.01786        | 0.01922         | 0.01901        | 0.01902        |
| D3   | RMSE             | 0.06225        | 0.06433         | 0.06423        | 0.06928        |
|      | MAE              | 0.03475        | 0.03617         | 0.03870        | 0.04153        |
| D4   | MAE              | 0.11449        | 0.11752         | 0.11604        | 0.11857        |

*"w/o" means ‘without’.

V. CONCLUSIONS

In this paper, we propose a Collaborative-informed Graph Convolutional Network (NGCN) that learns latent collaborative information from the interactions of node pairs outside of the GNN. It offers flexible incorporation of both node features and latent collaborative information of node pairs, thereby achieving significant performance improvements on missing weight estimation of a UWG.

In future, we will further improve the effectiveness of NGCN by incorporating attention mechanism to measure the importance of latent collaborative information for different node-pairs. Moreover, we plan to theoretically analyze why latent collaborative information of node pairs can help GCN break the limitation of expressive power, whose upper-bounded is proven by the 1-Weisfeiler-Lehman (1-WL) graph isomorphism test [35].

REFERENCES

[1] M. Welling and T. N. Kipf, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. on Learning Representations, 2016.
[2] S. Gao, Y. Lin, N. Feng, C. Song, and H. Y. Wan, “Attention based spatial-temporal graph convolutional networks for traffic flow forecasting,” in Proc. of the AAAI Conf. on Artificial Intelligence, pp. 922-929, 2019.
[3] J. L. Yu, H. Z. Yin, J. D. Li, M. Gao, Z. Huang, and L. Z. Cui, “Enhancing Social Recommendation With Adversarial Graph Convolutional Networks,” IEEE Trans. on Knowledge and Data Engineering, vol. 34, no. 8, pp. 3727-3739, 2022.
[4] S. Azadifar and A. Ahmadi, “A novel candidate disease gene prioritization method using deep graph convolutional al networks and semi-supervised learning,” BMC bioinformatics, vol. 23, no. 1, pp. 1-25, 2022.

[5] D. Yu and T. Pan, “Tracing knowledge diffusion of TOPYSIS: A historical perspective from citation network,” Expert Systems with Applications, vol. 168, no. 2, pp. 114238, 2020.

[6] L. Hu, S. Yang, X. Luo, H. Yuan, K. Sedraoui and M. Zhou, “A Distributed Framework for Large-scale Protein-protein Interaction Data Analysis and Prediction Using MapReduce,” IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 1, pp. 160-172, 2022.

[7] L. Hu, J. Zhang, X. Pan, X. Luo and H. Yuan, “An Effective Link-Based Clustering Algorithm for Detecting Overlapping Protein Complexes in Protein-Protein Interaction Networks,” IEEE Trans. on Network Science and Engineering, vol. 8, no. 4, pp. 3275-3289, 2021.

[8] L. Hu, S. Yang, X. Luo and M. Zhou, “An Algorithm of Inductively Identifying Clusters From Attributed Graphs,” IEEE Trans. on Big Data, vol. 8, no. 2, pp. 523-534, 2022.

[9] X. Luo, H. Wu, Z. Wang, J. J. Wang, and D. Y. Meng, “A Novel Approach to Large-Scale Dynamically Weighted Directed Network Representation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, DOI: 10.1109/TPAMI.2021.3132503, 2021.

[10] W. L. Li, R. F. Wang, X. Luo, and M. C. Zhou, “A Second-order Neomorphic Non-negative Factor Model for Undirected Weighted Network Representation,” IEEE Trans. on Network Science and Engineering, DOI: 10.1109/TNSE.2022.3206802, 2022.

[11] M. Manriquez, C. Guerrero-Nuance, F. Martinez, and C. Taramasco, “A generalization of the importance of verticals in an unweighted graph,” Symmetry, vol. 13, no. 5, pp. 902, 2021.

[12] Q. M. Yao, X. N. Chen, J. T. Kwok, Y. Li, C. J. Hsieh, and M. Assoc Comp, “Efficient neural interaction function search for collaborative filtering,” in Proc. of Int. World Wide Web Conference, pp. 1660-1670, 2020.

[13] S. Yun, S. Kim, J. Y. Lee, J. Kang, H. J. Kim, “Neo-GNNs: Neighborhood Overlap-aware Graph Neural Networks for Link Prediction,” in Proc. of Advances in Neural Information Processing Systems, pp. 13683-13694, 2021.

[14] H. B. Pei, B. Z. Wei, K. C. Chang, Y. Lei, and B. Yang, “Geom-GCN: Geometric Graph Convolutional Networks,” in Proc. Int. Conf. on Learning Representations, 2020.

[15] Y. You, T. Chen, Z. Wang Z, and Y. Shen, “When does self-supervision help graph convolutional networks,” in Proc. Int. Conf. on Machine Learning, pp. 10871-10880, 2020.

[16] Z. Liu, X. Luo and Z. Wang, “Convergence Analysis of Single Latent Factor-Dependent, Nonnegative, and Multiplicative Update-Based Nonnegative Latent Factor Models,” IEEE Trans. on Neural Networks and Learning Systems, vol. 32, no. 4, pp. 1737-1749, 2021.

[17] X. Luo, Y. Zhou, Z. Liu and M. Zhou, “Fast and Accurate Non-Negative Latent Factor Analysis of High-Dimensional and Sparse Matrices in Recommender Systems,” IEEE Trans. on Knowledge and Data Engineering, vol. 35, no. 4, pp. 3897-3911, 2023.

[18] Y. Yuan, Q. He, X. Luo, and M. S. Shang, “A Multilayered-and-Randomized Latent Factor Model for High-Dimensional and Sparse Matrices,” IEEE Trans. on Big Data, vol. 8, no. 3, pp. 784-794, 2022.

[19] T. A. Davis, Y. Hu. “The University of Florida sparse matrix collection,” ACM Trans. on Mathematical Software, vol. 38, no. 1, 2011.

[20] D. Wu and X. Luo, “Robust Latent Factor Analysis for Precise Representation of High-Dimensional and Sparse Data,” IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 4, pp. 796-805, 2021.

[21] X. Luo, H. Wu and Z. Li, “Neulff: A Novel Approach to Nonlinear Canonical Polyadic Decomposition on High-Dimensional Incomplete Tensors,” IEEE Trans. on Knowledge and Data Engineering, vol. 35, no. 6, pp. 6148-6166, 2023.

[22] L. Hu, X. Pan, Z. Tang and X. Luo, “A Fast Fuzzy Clustering Algorithm for Complex Networks via a Generalized Momentum Method,” IEEE Trans. on Fuzzy Systems, vol. 30, no. 9, pp. 3473-3485, 2022.

[23] M. Shang, Y. Yuan, X. Luo and M. Zhou, “An α–β-Divergence-Generalized Recommender for Highly Accurate Predictions of Missing User Preferences,” IEEE Trans. on Cybernetics, vol. 52, no. 8, pp. 8006-8018, 2022.

[24] X. Luo, D. X. Wang, M. C. Zhou, and H. Y. Yuan, “Latent factor-based recommenders relying on extended stochastic gradient descent algorithms,” IEEE Trans. on Systems, Man, and Cybernetics: Systems, vol. 51, no. 2, pp. 916-926, 2021.

[25] X. N. He, L. Liao, H. Zhang, et al., “Neural collaborative filtering,” in Proc. of the Int. Conf. on World Wide Web, pp. 173-182, 2017.

[26] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, “Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach,” in Proc. of the AAAI Conf. on Artificial Intelligence, pp. 27-34, 2020.

[27] W. Guo, Y. Yang, Y. C. Hu, C. Y. Wang, H. F. Guo, Y. X. Zhang, R. M. Tang, W. N. Zhang, X. Q. He, “Deep graph convolutional networks with hybrid normalization for accurate and diverse recommendation,” in Proc. of Workshop on Deep Learning Practice for High-Dimensional Sparse Data with KDD, 2021.

[28] X. Luo, Y. Zhou, Z. Liu, L. Hu and M. Zhou, “Generalized Nesterov’s Acceleration-Incorporated, Non-Negative and Adaptive Latent Factor Analysis,” IEEE Trans. on Services Computing, vol. 15, no. 5, pp. 2809-2823, 2022.

[29] R. V. Berg, T. N. Kipt, and M. Welling, “Graph convolutional matrix completion,” arXiv preprint arXiv:1706.02263, 2017.

[30] X. N. He, K. Deng, X. Wang, Y. Li, Y. D. Zhang, and M. Wang, “LightGCN: Simplifying and powering graph convolution network for recommendation,” In Proc. of the Int. ACM SIGIR Conf. on research and development in Information Retrieval, pp. 639-648, 2020.

[31] X. Luo, Y. R. Zhong, Z. D. Wang, and M. Z. Li, “An Alternating-direction-method of Multipliers-Incorporated Approach to Symmetric Non-negative Latent Factor Analysis,” IEEE Trans. on Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2021.3125774, 2021.

[32] J. Chen, X. Luo and M. Zhou, “Hierarchical Particle Swarm Optimization-incorporated Latent Factor Analysis for Large-Scale Incomplete Matrices,” IEEE Trans. on Big Data, vol. 8, no. 6, pp. 1524-1536, 2022.

[33] Z. Li, S. Li, O. O. Bamagas, A. Althothali and X. Luo, “Diversified Regularization Enhanced Training for Effective Manipulator Calibration,” IEEE Trans. on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2022.3153039.

[34] X. Luo, M. Chen, H. Wu, Z. Liu, H. Yuan and M. Zhou, “Adjusting Learning Depth in Nonnegative Latent Factorization of Tensors for Accurately Modeling Temporal Patterns in Dynamic QoS Data,” IEEE Trans. on Automation Science and Engineering, vol. 18, no. 4, pp. 2142-2155, 2021.

[35] K. Xu, W. H. Hu, J. Leskovec, S. Jegelka, “How Powerful are Graph Neural Networks?, “ in Proc. of Int. Conf. on Learning Representations, 2019.

[36] D. Wu, P. Zhang, Y. He and X. Luo, “A Double-Space and Double-Norm Ensembled Latent Factor Model for Highly Accurate Web Service QoS Prediction,” IEEE Trans. on Services Computing, vol. 16, no. 2, pp. 802-814, 2023.

[37] D Wu, X Luo, Y He and M. Zhou, “A prediction-sampling-based multilayer-structured latent factor model for accurate representation to high-dimensional and sparse data,” IEEE Trans. on Neural Networks and Learning Systems, 2022.

[38] H. Wu, X. Luo, M. Zhou, M. J. Rawa, K. Sedraoui and A. Albeshti, “A PID-incorporated Latent Factorization of Tensors Approach to Dynamically Weighted Directed Network Analysis,” IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 3, pp. 533-546, 2022.

[39] Z. Liu, G. Yuan and X. Luo, “Symmetry and Nonnegativity-Constrained Matrix Factorization for Community Detection,” IEEE/CAA Journal of Automatica Sinica, vol. 9, no. 9, pp. 1691-1693, 2022.