

# DEEP GRAPH CONVOLUTION NEURAL NETWORK WITH NON-NEGATIVE MATRIX FACTORIZATION FOR COMMUNITY DISCOVERY

Shuliang Xu<sup>†</sup>, Shenglan Liu<sup>‡</sup> and Lin Feng<sup>‡,\*</sup>

<sup>†</sup> Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology

<sup>‡</sup> School of Innovation and Entrepreneurship, Dalian University of Technology

slx\_cs@mail.dlut.edu.cn, {liusl, fenglin}@dlut.edu.cn

## ABSTRACT

Community discovery is an important task for graph mining. Owing to the nonstructure, the high dimensionality and the sparsity of graph data, it is not easy to obtain an appropriate community partition. In this paper, a deep graph convolution neural network with non-negative matrix factorization (DGCN-NMF) is proposed for community discovery. DGCN-NMF employs multiple graph convolution layers to obtain the low dimensional embedding. In each layer, the last output and the non-negative matrix factorization's results of the previous outputs are inputted into the current layer. The community partition of the inputted graph can be outputted by an end-to-end approach. The proposed algorithm and the comparison algorithms are conducted on the experimental data sets. The experimental results show that the proposed algorithm outperforms the comparison algorithms on the experimental data sets. The experimental results demonstrate that DGCN-NMF is an effective algorithm for community discovery. The implementation of DGCN-NMF is available at <https://github.com/hmliangliang/DGCN-NMF>.

**Index Terms**— Deep graph convolution neural network, non-negative matrix factorization, self-supervised learning, complex network, community discovery

## 1. INTRODUCTION

Community discovery is an important approach for graph mining. It partitions the nodes of a graph into multiple communities. The nodes in the same community are densely connected and the nodes in different communities are sparsely connected [1, 2]. Community discovery has been applied to many applications such as social network analysis [3], recommended system [4] and geographic information system [5], etc. Adjacency matrix is an important representation of a graph. However, due to the non-structure, the high dimension and the sparsity of graph data, it brings great challenges for community discovery algorithms [6]. How to effectively mine community structure in graph data has attracted much

attention from researchers. Up to now, many community discovery algorithms have been proposed. They can be divided into three categories: modularity optimization, matrix factorization and graph neural network.

Modularity optimization is a heuristic approach to maximize modularity [7] to discover the community structure. It can discover an appropriate community for each node towards the direction of increasing modularity. In modularity-based algorithms, Holloco et al. propose a modularity-based sparse soft graph clustering algorithm [8]. It can discover the community structure by maximizing the modularity of the nodes' partition and uses sparse matrix to record the partition result to improve the efficiency. Wang et al. propose a merging search algorithm based on modularity dominated density [9]. Each node is partitioned into the community with the maximum modularity. After all nodes are divided into communities, a community revision is conducted. Multiple small communities are merged into a large community if the merger can improve the modularity of the community structure. A large community can also be split into multiple small communities if the split can improve the modularity. However, the community discovery algorithms based on modularity optimization only uses the neighborhood relationship of nodes and ignores the high-order information of a graph which can provide more semantic information. In addition, it is not easy to solve the optimization problem of the modularity. Therefore the above disadvantages restrict the applications of the modularity optimization in community discovery.

Matrix factorization is to decompose adjacency matrix or an information matrix related to adjacency matrix into multiple matrices. The community partition of the nodes can be directly obtained from the decomposition result. Therefore matrix factorization has a good explanation for the result of community discovery. For matrix factorization algorithms, Pei et al. propose a nonnegative matrix tri-factorization algorithm [10]. It defines three interactive information matrices. Each interactive information matrix is decomposed into three matrices and introduces graph regularization to improve the performance of the algorithm. The partition result of the nodes is the fusion of the decomposition results of the three in-

---

\*Corresponding author.

teractive information matrices. However, the time complexity of matrix factorization is high and it is not fit for a large scale network.

Graph neural network is a hot topic in recent years [11, 12]. It can obtain the low dimensional embedding vectors of nodes from the neighbors by one or multiple hidden layers. Due to the strong representation ability of neural network, graph neural network can effectively fuse the attribute and the structure information of a graph. In graph neural network algorithms, He et al. propose a community-centric graph convolutional network for community discovery [13]. It utilizes conditional random field model to design autocoder and introduces a local enhancement approach to make the similar nodes in the same community. Sun et al. propose a deep autoencoder-like nonnegative matrix factorization algorithm for community discovery [14]. It uses non-negative matrix factorization as the autoencoder and stacks multiple autoencoders to form the hierarchical decomposition result. However, in a graph, there is no tight connection between each node and the most other nodes and it means the adjacency matrix of a graph is high dimensional and sparse. If adjacency matrix is only as the initial input of graph convolution neural network [15] without attribute information which is called as non-attributed graph, the sparse adjacency matrix cannot provide enough semantic information for graph convolution neural network.

In order to solve the reconstruction of the current algorithms and better discover the community structure, a deep graph convolution neural network with non-negative matrix factorization (DGCN-NMF) is proposed in this paper. DGCN-NMF employs multiple graph convolution layers to obtain the low dimensional embedding vectors of nodes. For each graph convolution layer, the outputs of the previous layers are decomposed into the low dimensional and dense matrices. The decomposition results of the previous layers and the output of the current layer are concatenated as the inputs of the next layers. The community partition can be obtained by an end-end approach. The main contributions of this paper are as follows:

- A deep graph convolution neural network with self-supervised learning mechanism is proposed. DGCN-NMF can be trained from unlabeled nodes in a graph.
- Considering the concatenator increases the number of the parameters in a graph convolution neural network which can result in over-fitting, the output of each layer is decomposed into the low dimensional and dense matrices. The decomposition result is as one of the inputs of the next layers.
- The community assignment of each node can be outputted by an end-end-approach. The final community structure can be obtained by softmax approach.

The rests of this paper are organized as follows: Section 2

describes the detail theories and the steps of the proposed algorithm; Section 3 presents the experimental results and analyses. Section 4 concludes this paper and gives some research directions in the future.

## 2. THE PROPOSED ALGORITHM

Let  $G = \langle V, E, \mathbf{A}, \mathbf{X} \rangle$  be a graph.  $V \in \mathbb{R}^n$  is the nodes of the graph  $G$  and  $n$  is the number of the nodes.  $E$  is the edges of  $G$ .  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the adjacency matrix of  $G$ . For  $\forall v_i, v_j \in V$ ,  $A_{ij} = 1$  if  $e_{ij} \in E$ , otherwise,  $A_{ij} = 0$ .  $\mathbf{X} \in \mathbb{R}^{n \times m}$  is the attribute information of the nodes and  $m$  is the dimensionality of the attribute information. The target of DGCN-NMF aims at learning a mapping function  $f : V \mapsto \mathbb{R}^{n \times k}$ . The nodes are assigned to the  $k$  communities. The structure of DGCN-NMF is showed as Fig.1.

### 2.1. The graph convolution of DGCN-NMF

Graph convolution neural network (GCN) can learn the low dimensional embedding vector of each node from the neighbors by the convolution and aggregation operators [16]. DGCN-NMF introduces multiple convolution layers to obtain the low dimensional embedding vectors of nodes. The convolution operator of each layer is as follow:

$$\mathbf{H}_{\ell+1} = \delta \left( \hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}} \mathbf{H}_{\ell} \mathbf{W}_{\ell} \right) \quad (1)$$

where  $\mathbf{H}_{\ell}$  is the output of the  $\ell$ th layer and it is also the input of the  $(\ell + 1)$ th layer.  $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$  and  $\mathbf{I}$  is an identity matrix.  $\hat{\mathbf{D}}$  is a diagonal matrix and  $\hat{D}_{ii} = \sum_{j=1}^n \hat{A}_{ij}$ .  $\mathbf{W}_{\ell}$  is the parameters of the  $\ell$  layer.  $\mathbf{H}_0 = \mathbf{X}$  if the graph  $G$  is an attributed graph.  $\mathbf{H}_0 = \mathbf{I}$  if the graph  $G$  is a non-attributed graph.  $\delta(\cdot)$  is an activation function.

It is known that  $\mathbf{H}_{\ell+1}$  contains more high-level semantic information than  $\mathbf{H}_{\ell}$ . If  $\mathbf{H}_{\ell+1}$  is as the output of DGCN-NMF, it means that only high-level semantic information is used and much low-level semantic information is ignored. For a graph, the low-level semantic information also plays an important role on the final embedding since the low-level semantic information can reflect the local structure information of a graph. Therefore both  $\mathbf{H}_{\ell}$  and the outputs of the previous  $(\ell - 1)$  layers are as the input of the  $(\ell + 1)$ th layer. However, the inputted dimension of the  $(\ell + 1)$  layer is large if the  $\ell$  outputs are concatenated. The high dimension results in more parameters in graph convolution neural network and the over-fitting also appears.

In DGCN-NMF algorithm, the outputs of the previous  $(\ell - 1)$  layers are not concatenated directly. For  $\forall \mathbf{H}_i$  ( $i = 0, 1, 2, \dots, \ell - 1$ ), it is decomposed into two matrices by non-negative matrix factorization (NMF) [17] as follow:

$$\mathbf{H}_i = \mathbf{U}_i \mathbf{V}_i \quad s.t. \quad \mathbf{U}_i \geq 0, \mathbf{V}_i \geq 0 \quad (2)$$

where  $\mathbf{H}_i \in \mathbb{R}^{n \times L_i}$ ,  $\mathbf{U}_i \in \mathbb{R}^{n \times d}$ ,  $\mathbf{V}_i \in \mathbb{R}^{d \times L_i}$ ,  $L_i$  is the outputted dimension of the  $i$ th layer and  $d$  is the output dimension of the non-negative matrix factorization.  $\mathbf{U}_i$  is the low dimensional representation of  $\mathbf{H}_i$  and  $d \ll L_i$ . Therefore the final the input of the  $(\ell + 1)$  is as:

$$\mathbf{H}_\ell \leftarrow \mathbf{H}_\ell \parallel \mathbf{U}_0 \parallel \cdots \parallel \mathbf{U}_{\ell-1} \quad (3)$$

where  $\parallel$  is the concatenated operator and it can concatenate two matrices.

The low dimensional embedding vectors of nodes are outputted by GCNs. If the graph  $G$  is an attributed graph,  $\mathbf{X}$  is also concatenated to the final output of GCNs. Then the low dimensional embedding vectors are inputted into a MLP with single hidden layer. Let  $\mathbf{Z} \in \mathbb{R}^{n \times k}$  be the output of MLP where  $k$  is the number of communities. The community assignment of nodes can be determined from  $\mathbf{Z}$  by *softmax*.

## 2.2. The self-supervised learning of DGCN-NMF

In DGCN-NMF, the GCNs and MLP are trained from the unlabeled nodes of the graph  $G$ . Self-supervised learning is introduced in this paper. The output  $\mathbf{Z}$  of DGCN-NMF contains both the high-level and the low-level semantic information of the graph  $G$ . Therefore the adjacency matrix of the graph  $G$  should be restructured from  $\mathbf{Z}$ . The loss of the restruction  $\mathcal{L}_s$  is defined as:

$$\mathcal{L}_s = \|\mathbf{Z}\mathbf{Z}^T - \mathbf{A}\|_F^2 \quad (4)$$

If the graph  $G$  is an attributed graph, the loss of the restruction  $\mathcal{L}_s$  is defined as:

$$\mathcal{L}_s = \|\mathbf{Z}\mathbf{Z}^T - \mathbf{A}\|_F^2 + \|\mathbf{Z}\mathbf{Z}^T - \mathbf{X}\mathbf{X}^T\|_F^2 \quad (5)$$

In Eq.(5), the first term is the restruction loss of the adjacency matrix. The second term is the restruction loss of the attribute information. DGCN-NMF is to restructure the structure and attribute information and minimize the restruction loss  $\mathcal{L}_s$ .

It is known that modularity is a measure for the strength of community structure. A large modularity means a better community structure. DGCN-NMF is to make the modularity of the community partition be as large as possible. The modularity  $\mathcal{L}_m$  is as:

$$\mathcal{L}_m = \frac{1}{2e} \sum_{i,j} \left[ A_{ij} - \frac{k_i \cdot k_j}{2e} \right] \sigma(v_i, v_j) \quad (6)$$

where  $e$  is the number of edges,  $k_i, k_j$  are the degrees of the nodes  $v_i, v_j$ , respectively.  $\sigma(v_i, v_j) = 1$  if  $v_i$  and  $v_j$  are in the same community, otherwise,  $\sigma(v_i, v_j) = 0$ .

Therefore DGCN-NMF is to discover a community partition with the minimum loss. The optimization problem of DGCN-NMF is as follow:

$$\min_{\mathbf{Z}} \mathcal{L} = \mathcal{L}_s - \alpha \mathcal{L}_m \quad (7)$$

where  $\alpha$  is a parameter. After the loss in Eq.(7) is computed, the parameters of the GCNs and MLP in DGCN-NMF are adjusted and optimized by the back propagation approach.

Eqs.(1)-(7) show the main steps of DGCN-NMF. It is known that the conventional GCN only outputs the high-level semantics features of a graph and DGCN-NMF can fuse both the high-level and the low-level semantics features. Therefore DGCN-NMF can use more structure information of a graph.

## 2.3. Time complexity analysis

For DGCN-NMF, it costs  $\mathcal{O}(n^3)$  to compute  $\hat{\mathbf{D}}^{-\frac{1}{2}} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-\frac{1}{2}}$ . For each layer, it costs  $\mathcal{O}(nL_\ell L_{\ell-1})$  to realize the nonlinear transformation. The time complexity of NMF is  $\mathcal{O}(ndL_\ell)$  for the  $\ell$  layer. It costs  $\mathcal{O}(n^2)$  to compute Eqs.(4)-(6). Let  $d'$  be the input dimension of MLP. The time complexity of MLP is  $\mathcal{O}(nd'k)$ . Therefore the time complexity of DGCN-NMF is  $\mathcal{O}(n^3t)$  where  $t$  is the total epoches.

## 3. THE EXPERIMENTAL RESULTS AND ANALYSES

In order to test the performance of DGCN-NMF, STCD [18], MDDDBMS [9], MNMF [19] and ESCG [20] are selected as the comparison algorithms. For DGCN-NMF, it uses double hidden layers and the outputted dimensions are as [150, 100]. The activation function of MLP is *sigmoid*.  $d = 64$  and  $\alpha \in [0, 100]$ . DGCN-NMF-s is the DGCN-NMF algorithm with *sigmoid* activation function for GCNs. DGCN-NMF-r is the DGCN-NMF algorithm with *ReLU* activation function for GCNs. Adjnoun, football and politics are selected as the experimental data sets. Owing to the restriction of the layout, the descriptions of the data sets are not showed in this paper and the detailed descriptions can be seen from the websites<sup>1,2</sup>.

In this paper, Jaccard index (J), Folkes and Mallows index (FM), Kulczynski index (K), recall and F1-measure (F1) are as the evaluation criteria. The descriptions of the evaluation criteria can be seen from the reference [21].

**Table 1.** The experimental results of the algorithms on adj-noun data set.

	J	FM	K	recall	F1
STCD	0.0135	0.1292	0.3202	0.0272	0.0522
MDDDBMS	0.0145	0.1208	0.2646	0.0292	0.0551
MNMF	0.2458	0.4942	0.4942	0.4953	0.4942
ESCG	0.3953	0.6271	0.6463	0.7967	0.6086
DGCN-NMF-r	<b>0.4781</b>	<b>0.6911</b>	<b>0.7297</b>	<b>0.9637</b>	<b>0.6546</b>
DGCN-NMF-s	0.2476	0.4954	0.4954	0.4990	0.4954

Tables 1-3 shows the experimental results. From the results, it is known that DGCN-NMF outperforms the compar-

<sup>1</sup><http://memetracker.org/data/index.html>

<sup>2</sup><http://www-personal.umich.edu/~%7Emejn/netdata/>



- [3] Liang Jiang, Leilei Shi, Lu Liu, Jingjing Yao, Bo Yuan, and Yongjun Zheng, "An efficient evolutionary user interest community discovery model in dynamic social networks for internet of people," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9226–9236, 2019.
- [4] Siyuan Liu and Shuhui Wang, "Trajectory community discovery and recommendation by multi-source diffusion modeling," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 4, pp. 898–911, 2016.
- [5] Polina Rozenshtein and Aristides Gionis, "Mining temporal networks," in *KDD*, 2019, pp. 3225–3226.
- [6] Fanzhen Liu, Shan Xue, Jia Wu, Chuan Zhou, Wenbin Hu, Cecile Paris, Surya Nepal, Jian Yang, and Philip S Yu, "Deep learning for community detection: Progress, challenges and opportunities," in *IJCAI*, 2020, pp. 4981–4987.
- [7] Mark EJ Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006.
- [8] Alexandre Hollocou, Thomas Bonald, and Marc Lelarge, "Modularity-based sparse soft graph clustering," in *AISTATS*, 2019, pp. 323–332.
- [9] Zhongyu Wang and Jinwen Ma, "Modularity dominated density based merging search for community discovery," in *IEEE SMC*, 2015, pp. 2742–2748.
- [10] Yulong Pei, Nilanjan Chakraborty, and Katia Sycara, "Nonnegative matrix tri-factorization with graph regularization for community detection in social networks," in *IJCAI*, 2015, pp. 2083–2089.
- [11] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2020.2978386, 2020.
- [12] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski, "Graph convolutional networks: a comprehensive review," *Computational Social Networks*, vol. 6, no. 1, pp. 11, 2019.
- [13] Binbin Zhang, Zhizhi Yu, and Weixiong Zhang, "Community-centric graph convolutional network for unsupervised community detection," in *IJCAI*, 2020, pp. 3515–3521.
- [14] Fanghua Ye, Chuan Chen, and Zibin Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *CIKM*, 2018, pp. 1393–1402.
- [15] Thomas N Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," in *I-CLR*, 2017.
- [16] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *NIPS*, 2016, pp. 3844–3852.
- [17] Daniel D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [18] Liang Bai, Jiye Liang, Hangyuan Du, and Yike Guo, "A novel community detection algorithm based on simplification of complex networks," *Knowledge-Based Systems*, vol. 143, pp. 58–64, 2018.
- [19] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang, "Community preserving network embedding," in *AAAI*, 2017, vol. 17, pp. 203–209.
- [20] Jialu Liu, Chi Wang, Marina Danilevsky, and Jiawei Han, "Large-scale spectral clustering on graphs," in *IJCAI*, 2013, pp. 1486–1492.
- [21] Juanying Xie, *Unsupervised Learning Methods and Applications*, Publishing House of Electronics Industry, 2016.