2021 IEEE 6th International Conference on Intelligent Computing and Signal Processing (ICSP 2021)

Community Detection Based On Graph Neural Network

Wang P, Yunnan University
School of Information Science & Engineering
Kunming, China
hnnywp@163.com

Kong B*, Yunnan University
School of Information Science & Engineering
Kunming, China
kongbing@ynu.edu.cn

Bao C.M, Yunnan University
School of Software
Kunming, China
cywang@ynu.edu.cn

Zhou L.H, Yunnan University
School of Information Science & Engineering
Kunming, China
lhzhou@ynu.edu.cn

Wang C.Y, Yunnan University
Institute of Ecology and Geobotany
Kunming, China
cywang@ynu.edu.cn

***Abstract*: Community detection has always been an important and challenging task in the analysis of complex networks. In real life, many networks are heterogeneous which contain different types of nodes and edges. In the past few years, people have become more and more interested in the extension of deep learning techniques on heterogeneous network community detection.**

Keywords: heterogeneous; meta-path; graph convolution;

I. Introduction

The traditional community detection algorithms simply use the existing features of nodes to calculate, and the feature selection of different task nodes often has different effects on the success of the task. Therefore, researchers began to consider embedding learning on graph networks [1] in order to obtain potential representations of node features.With the development of convolutional god network, researchers begin to consider introducing deep learning technology into graph network embedding learning, whose representative achievements include Deepwalk [2], Node2vec [3], LINE [4]. Deepwalk first uses deep learning in network embedding. It borrows from the idea of word vector in natural language processing and uses random walk model for data sampling, which promotes the development of graph neural network.

In recent years, deep learning on heterogeneous graph network has achieved rapid development, and this work is called heterogeneous graph neural network[5]. Influenced by idea of Deepwalk , researchers began to consider realize deep learning on heterogeneous networks. Metapath2vec [6] was born in such a context. Metapath2vec used meta-path to guide the sampling direction of random walk, proposed a data sampling method in heterogeneous network, and combined with SkipGram [7] model, obtained the feature representation of nodes, which promoted the development of deep learning in heterogeneous network.

However, Metapath2vec implements deep learning in heterogeneous networks, but it does not use convolution model. In 2013, researchers proposed the concept of graph convolution neural network at the first time [8], the related work is mainly based on spectral graph theory, generally known as the spectral method of graph convolution operation. However, due to the high space and time complexity of this method, the Graph Convolution Network (GCN) method [9] was proposed in 2016. It optimized the convolution kernel, greatly reduced the space and time complexity, and made the Graph Convolution Neural Network develop rapidly. Although GCN is classified as a spectral approach, it has begun to consider the structure of graphs in spatial domain. In 2017, the GAT [10] and GraphSage [11] models proposed successively that they considered the information of neighbor nodes in the graph network from the spatial domain.

Despite the rapid development of GCN, it still cannot be applied to heterogeneous networks. In order to solve the problem that graph neural network can not be used in heterogeneous network, this paper adopts the concept of user-guided meta-path to transform heterogeneous network into homogeneous network according to the requirements. At the same time, according to the information of the original network structure, different weights are given to the edges between nodes in the homogeneous network. When we calculate the importance of neighbor nodes, edge weight information is taken into account. Therefore, we propose a Weighted Graph Attention network (WGAT) based on symmetric meta-path [12]. WGAT can capture the potential characteristics of nodes, and then realize clustering and classification. Experiments show that WGAT has better performance than GCN and GAT.

II. Problem definition

Community detection is a basic task in data mining. Our goal is to realize convolution operation on heterogeneous

978-1-6654-0413-6/21/$31.00 ©2021 IEEE

2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP) | 978-1-6654-0413-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICSP51882.2021.9408828

network for cluster analysis. We first give the definition of heteronetworks and meta-paths

A. Heterogeneous information networks

A heterogeneous information network can be defined as a directed graph or an undirected graph： $G = (V, E)$， Where the mapping relation of node types is $\phi: V \rightarrow \mathcal{A}$， The mapping of the edge type is $\psi: E \rightarrow \mathcal{R}$. For any node $v \in V$ The node type mapping relation to which it belongs is $\phi(v) \in \mathcal{A}$, For any edge $e \in E$, The mapping relation of the edge type to which it belongs is $\psi(e) \in \mathcal{R}$. For a heterogeneous network $G = (V, E)$, The network mode refers to the abstract representation between the node type mapping $\phi: V \rightarrow \mathcal{A}$ and the edge type mapping $\psi: E \rightarrow \mathcal{R}$ in the network, which can be denoted as $T_G = (\mathcal{A}, \mathcal{R})$.

B. Meta-path

A meta-path $\mathcal{P}$ is a path defined in a heterogeneous network model $T_G = (\mathcal{A}, \mathcal{R})$, its expansion is $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} A_3 \cdots A_L \xrightarrow{R_L} A_{L+1}$, which A represents node, R represents the relationship between the nodes. Let P represent one meta-path instance under the specified meta-path $\mathcal{P}$. We usually simplify the relationship between meta-path, such as $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} A_3 \cdots A_L \xrightarrow{R_L} A_{L+1}$ can be shorthand for $\mathcal{P} = (A_1 A_2 \ldots A_{l+1})$.

In any networks, we can a meta-path $\mathcal{P} = APC$ .If R is symmetric between node A and node P, i.e. $AP = PA$. We assume that the relationship between node P and node C is also symmetric.Then the meta-path $\mathcal{P}_1 = APC$ is equal to the meta-path $\mathcal{P}_2 = CPA$. So we can link $\mathcal{P}_1$ and $\mathcal{P}_2$, that is $\mathcal{P} = APCPA$. We call such meta-path $\mathcal{P}$ a symmetric meta-path.

III. Modeling Based on Meta-Paths

A. Subgraph extraction

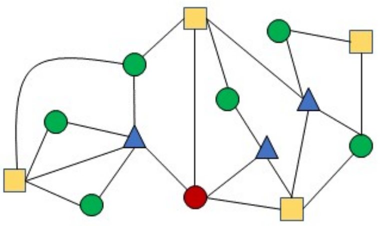

Figure 1. A heterogeneous network

As shown in Figure 1, this heterogeneous network contains three node types, and there is no link relationship between any different type of node. However, in some meta-path relationships, nodes of the same type can be linked together by conduction of other types of nodes. For example, for a specified in the heterogeneous network nodes (red), the same type to other nodes (green), there are multiple meta-paths between them. We can regard green circle, blue triangles and green circle as a meta-path, as shown in figure 2, then we can extract the link relationship of the same node under this path from the original network. In this way, we can extract the nodes in the heterogeneous network that satisfy the specific meta-path into the homogeneous network. For example, in a shopping network, when we want to know what payment method will the buyer adopt, we can transform the heterogeneous shopping network into a homogeneous network by using the simple symmetric meta-path of buyer-payment method-buyer. This homogeneous network contains only one type of buyer. Obviously, there is no side connection between the buyers who pay using WeChat and those who pay using credit cards. Therefore, we can mine out different communities under the meta-path of user-guided relationship according to the homogeneous network.

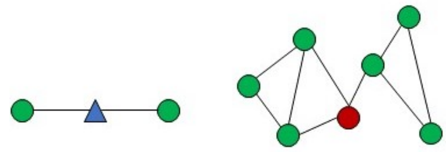

Figure 2. Meta-path and homogeneous networks

B. Edge weight calculation

Although a symmetric meta-path can extract nodes of the same type from the heterogeneous network under the specified meta-path relationship, as shown on the right of Figure 2. But we can not rigidly regard the extracted homogeneous network as equal edge relations. In a citation network, for example, under the cooperation meta-path (author - paper - writer), the researchers A and B jointly published 20 papers, researchers A and C jointly published 1 paper, obviously A and B should be more similar, so after the conversion of homogeneity in the network, we need to redefine the edge weights.

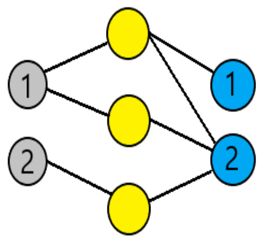

Figure 3. Number of meta-path instances

As shown in Figure 3, it is easy to calculate the number of meta-path from gray node 1 to blue node 2 as 2. This shows homogeneous network extracted by symmetric meta-path from heterogeneous network, its edge between node is inequitable, the value of each edge represents the actual number of meta-path instances. For an edge with a higher value, we can define it as the weight of the edge or the characteristic conductivity, that is, it is easier to conduct attribute between nodes with more meta-path instances.

As can be seen from the above, the matrix obtained by multiplying the adjacency matrices under the symmetric meta-path is expressed as the number of specific meta-path instances of different nodes β_{ij} under this relationship. Therefore, we can define this number as the weight of edges between nodes of homogeneous graph after transformation.

Calculating the number of meta-path instances can be obtained by multiplying the adjacency matrix in the meta-path. But in a large-scale network, this approach is obviously not advisable, We can get the weight coefficients through random walk sampling and parallel calculation.

IV. WEIGHTED GRAPH ATTENTION MODEL

By specifying a meta-path, a heterogeneous network can be transformed to a homogeneous network, which also keeps the structure information of the original network at the same time, the weight of the edge between node number is based on its meta-path. It is clear from the above that meta-path can not only homogenize a heterogeneous network, but also obtain the weight information of the edge in the subgraph.

A. Label detection

After we convert the heterogeneous network into the homogeneous network, we must obtain the node labels for the convenience of subsequent training. We use the method in [13] to get the label of the node.First, all nodes in the network are assigned a unique label, and then, for each node, its label is changed to the label of a neighbor node. After the change, the modularity degree will change. When the value of modularity degree increment is correct, we recognize the alteration of the label. Repeat the above operation for each node, and when the modularity of all communities does not change, we can get the label of each node in the network.

B. Weighted Graph Attention network

For any network, Its node characteristics are defined as $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}$, By introducing self-attention mechanisms, we can calculate the importance of neighbor nodes as shown in Figure 4. Through the aggregation of neighbor node information, we can get the new feature representation of node.

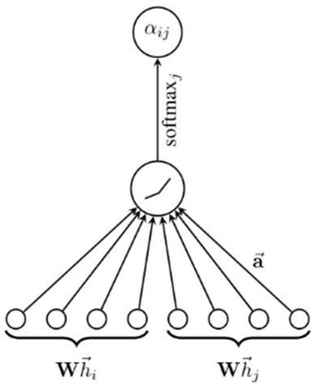

Figure 4. Attention mechanism

For any two nodes, the attention coefficient is:

$$e_{ij} = a\left(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j\right) \tag{1}$$

We used the softmax function to normalize it:

$$\alpha_{ij} = \mathrm{softmax}_j\left(e_{ij}\right) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp\left(e_{ik}\right)} \tag{2}$$

In the experiments, We initialize a vector with dimensions of $2F'$, and introduce nonlinear activation function LeakyReLU, formula (2) to:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)} \tag{3}$$

After obtaining the attention coefficient, we can use the multi-head attention mechanism (as shown in Fig. 5), then add them together to get a new representation of node:

$$\vec{h}_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right) \tag{4}$$

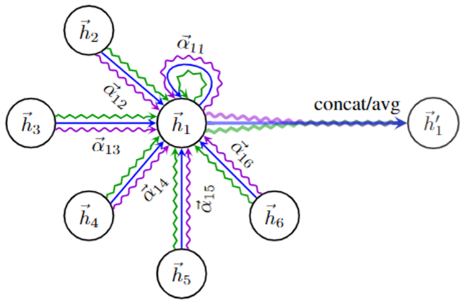

Figure 5. Multi-head mechanism

GAT introduces the self-attention mechanism, which can calculate the similarity coefficient between nodes and neighbor nodes. However, as we pointed out in the last section, the weight of edges between different nodes is different, so when we aggregate the features of neighbor nodes, we should also aggregate neighbor features according to the weight of edges.Referring to the above definition, after calculating the similarity coefficient between nodes, we should add the weight coefficient between nodes.

After we get formula (4), we multiply the node similarity coefficient with the edge weight coefficient to get our final formula:

$$\vec{h}_i' = \sigma\left(\frac{1}{k}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}(\alpha_{ij}^k\beta_{ij}^k)\mathbf{W}^k\vec{h}_j\right) \tag{5}$$

V. EXPERIMENT

A. Experimental setup and datasets

In this experiment, Eq. 5 is used for the convolution operation and a two-layer WGAT Convolution Model is used. Each layer consists of K=8 heads of attention, L2 regularization is adopted, where λ=0.0005. At the same time, the dropout of each layer is 0.6, and the initial adoption learning rate is 0.005.

In order to verify the effectiveness of the method proposed in this paper, the following data sets are used in this paper: ENZYMES dataset, PROTEINS dataset, github_star dataset, and 3 benchmark datasets cora, citeseer and pubmed.

Since this experiment is based on the WGAT model in the form of transductive learning, as a comparison, we use the GCN model, which is also in the form of transductive learning. The GCN model also adopts a two-layer convolutional structure and uses Chebyshev polynomial to approximate the convolutional kernel. The order of the convolutional kernel is 1, that is, only the neighbor nodes of first-order are considered.The comparison results are shown in Table 1.

TABLE I. COMPARISON OF GCN AND WGAT

DATASETS	GCN	WGAT
ENZYMES	63.1%	65.2%
GITHUB_S	49.3%	54.1%
PROTEINS	53.4%	61.5%
CORA	80.4%	86.5%
CITESEER	70.2%	75.6%
PUBMED	78.7%	82.6%

Table 1 shows that the accuracy of WGAT is better than that of GCN model. Meanwhile, through experimental comparison, the accuracy of WGAT on cora, citeseer and pubmed is all higher than that of GAT model. The comparison of accuracy results on the three datasets is shown in Figure 6.

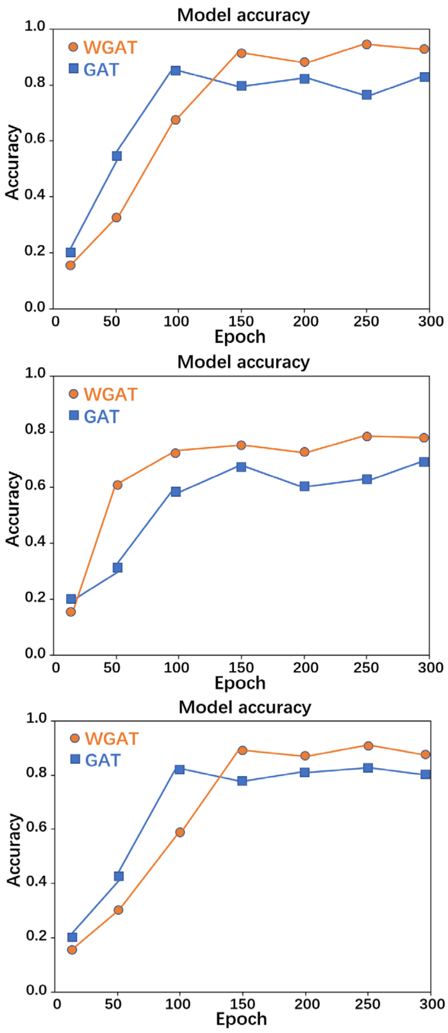

Figure 6. Comparison of WGAT and GAT

B. EADG WEIGHT

In the comparative experiment between WGAT and GAT, it is found that the influence of edge weight on the experimental results is very obvious. With the increase of edge weight, the accuracy rate of WGAT is much higher than that of GAT. Figure 7 shows the comparison between WGAT and GAT on the benchmark dataset when the weights are 5, 10 and 20 respectively.The results fully prove the rationality and accuracy of WGAT.

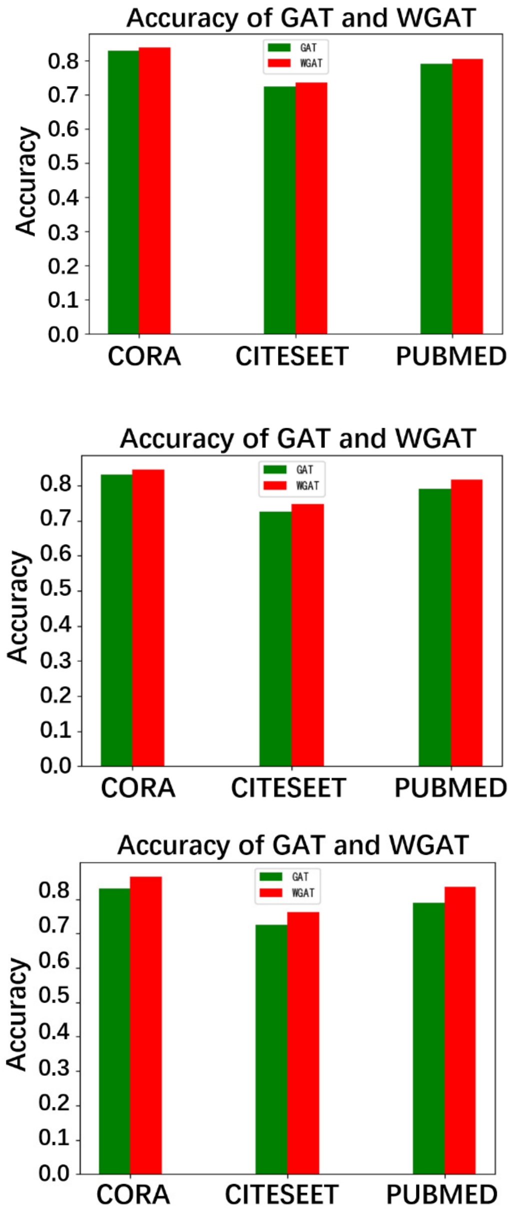

Figure 7. The relation between WGAT and weight

VI. CONCLUSIONS

Experiments show that this method is effective, especially in cora, citeseer and pumbed datasets. After adding weights, the accuracy rate of this method on the benchmark datasets reaches 86.5%, 75.6% and 82.6%, which all exceed the experiment shown in GAT. This indicates that it is necessary to consider the weight of edges. In Section 3, we pointed out that in heterogeneous network, the adjacency matrix obtained by using the symmetric meta-path relation between the same nodes can be regarded as the weight coefficient of the edges between nodes in the homogeneous network. Especially in the case of the larger data, using random walk sampling that we can quickly get edge weight coefficient between nodes, WGAT model can be designed for inductive mode at the same time. In the face of large-scale network, The weight of the edges in the network is difficult to calculate, so in the future we should consider to improve its performance in the face of the large-scale data.

REFERENCES

[1] Qi, J.-S & Liang, X. & Li, Z.-Y & Chen, Y.-F & Xu, Y.. (2018). Representation Learning of Large-Scale Complex Information Network: Concepts, Methods and Challenges. Jisuanji Xuebao/Chinese Journal of Computers. 41. 2394-2420. 10.11897/SP.J.1016.2018.02394.

[2] Perozzi, Bryan & Al-Rfou, Rami & Skiena, Steven. (2014). DeepWalk: Online Learning of Social Representations. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 10.1145/2623330.2623732.

[3] Tang, Jian & Qu, Meng & Wang, Mingzhe & Zhang, Ming & Yan, Jun&Mei,Qiaozhu. (2015). LINE: Large-scale Information Network Embedding. Line: Large-Scale Information Network Embedding.10.1145/2736277.2741093.

[4] Grover A , Leskovec J . node2vec: Scalable Feature Learning for Networks[C]// the 22nd ACM SIGKDD International Conference. ACM, 2016.I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[5] Zhang C , Song D , Huang C , et al. Heterogeneous Graph Neural Network[C]// the 25th ACM SIGKDD International Conference. ACM, 2019.

[6] Dong, Yuxiao & Chawla, Nitesh & Swami, Ananthram. (2017). metapath2vec: Scalable Representation Learning for Heterogeneous Networks. 135-144. 10.1145/3097983.3098036.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. CoRR, abs/1301.3781, 2013.

[8] Bruna, Joan & Zaremba, Wojciech & Szlam, Arthur & Lecun, Yann. (2013). Spectral Networks and Locally Connected Networks on Graphs.

[9] Kipf T N , Welling M . Semi-Supervised Classification with Graph Convolutional Networks[J]. 2016.

[10] Velikovi P , Cucurull G , Casanova A , et al. Graph Attention Networks[J]. 2017.

[11] Hamilton, William & Ying, Rex & Leskovec, Jure. (2017). Inductive Representation Learning on Large Graphs.

[12] Sun, Yizhou & Han, Jiawei & Yan, Xifeng & Yu, Philip & Wu, Tianyi. (2011). PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. PVLDB. 4. 992-1003. 10.14778/3402707.3402736.

[13] Blondel, Vincent & Guillaume, Jean-Loup & Lambiotte, Renaud & Lefebvre, Etienne. (2008). Fast Unfolding of Communities in Large Networks. Journal of Statistical Mechanics Theory and Experiment. 2008. 10.1088/1742-5468/2008/10/P10008.