Social Network Analysis Based on GraphSAGE

Lizhong Xiao[1],Xinke Wu[1],Guangzhong Wang[2]
1.School of Computer Science and Information Engineering
Shanghai Institute of Technology
Shanghai, China
2. Shanghai Academy of Educational Sciences
Shanghai, China
e-mail: wxk0903@foxmail.com

***Abstract*—Social network node classification has important application value in real life. Traditional iterative application classifiers and random walk based tag propagation algorithms do not fully consider the interaction information between nodes. Graphsage is an effective graph neural network model that can be used for social network analysis. This paper uses the graphsage model to analyze social networks, compares them with traditional social network analysis methods, discusses the values of some parameters of the model, and makes some improvements in the model sampling part.**

Keywords-GraphSage; social network; node classification; GNN

I. INTRODUCTION

With the rapid development of Internet, social network has become an indispensable part of people's life. With the global penetration of online and mobile social platforms, people have witnessed the impact of social network analysis in areas such as advertising [1] , Presidential Elections [2] , and monitoring of public opinion. And the information on social network is numerous and varied, among them spammy information appears without exception, make a person upset. For labels of, finding them will lead to better social platforms and people's social lives..

Graph neural network is a very popular technology in recent years. Graph is a widely used data structure[3]. Nodes represent objects and edges represent relations. It has a wide application in machine learning Such as social networks[4], bioengineering[5], Knowledge Atlas[6], mainly for node classification[7], link prediction[3], clustering[3] and other tasks. Graph Neural Network (GNNS) is a connectionist model that uses neural network technology based on graph to capture graph relationships by message passing between graph nodes. Unlike a standard neural network, a graph neural network can collect information from adjacent nodes at any depth around a node. Neural network can process graph structure data.

II. MATERIAL AND EXPERIMENTAL DATA PROCESSING

The Social Network Data Uses Sina Weibo data, all of which is crawled from Sina Weibo using a crawler, and personal information is processed.

The data includes two parts, the first part of the user information, mainly including the following fields: User ID, user name, user followee number, user follower number, user followee list, user follower list, account level. The second part is the information of partial posts, including each Weibo's retweet, comment, like number, as well as interactive user Id.

The experiment required undirected filtering of User Interactions, including retweets, comments, and likes of tweets that were followed. The user is the node, and the partial information of the user is the feature vector of the node. The specific areas they belong to.

III. MODEL

A. Problem description

After data collection and processing we get a user relationship adjacency table and a user information table. The goal of the experiment is to predict labels of users based on user relationships and user information.

For a better description of the problem and the model, the symbols and their meanings are given in table I.

Table I Definition of symbols

Notations	Descriptions
G = (V, E, X)	By node set V, Edge set E, node characteristic matrix X structure Into a social network, G
V	A set of nodes in a social network, |V| for the number of nodes
E	The set of edges of a social network, |E| , represents the number of edges
X	The feature vector of social network node
L	Node label set
v_i	The ith node in G, $v_i \in V$
$e_i = (v_s, v_r)$	The ith side of G, $e_i \in E$, v_s is the

978-1-7281-4653-9/19/$31.00 ©2019 IEEE
DOI 10.1109/ISCID.2019.10128

	sending node, and v_r is the receiving node
x_i	The eigenvector of the ith node in G, $x_i \in X$
Z_i	Label distribution of the ith node in G

B. Model descriptions

The goal of this paper is to design a model that can make full use of the attribute information, structure and information of nodes in social network, and mine the implicit mutual information among nodes to realize the task of node classification. Xu[8] points out in his paper that when the mapping function (update function and aggregate function) in a graph network is single, the best effect of the graph neural network can be equivalent to the graph WL ISOMORPHISM test. Inspired by their work, and Learning about graphsage [9],we use an improved GraphSage algorithm to build the model. The framework is shown in figure 1,2,3[9].

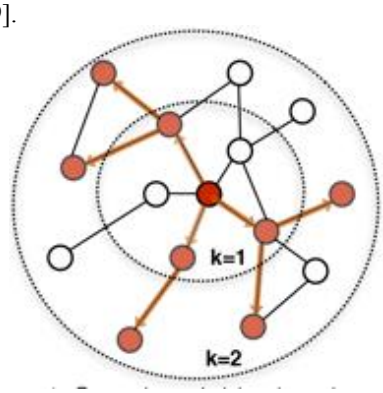

Fig 1.

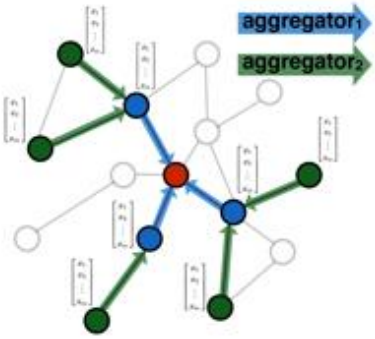

Fig.2

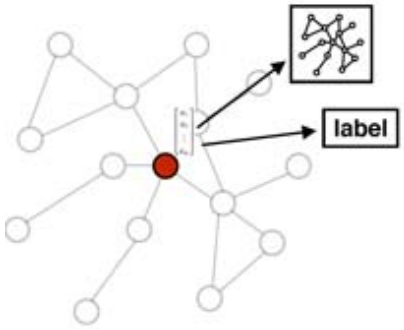

Fig.3

Message Spread: In message propagation, the goal is to propagate the node information to the neighbor and domain nodes, and at the same time to mine the mutual information between the nodes. For a given social network G=(V, E, X).The adjacent nodes of each node are sampled and the feature set is aggregated by an aggregation function, such as Formula 1.

$$h_N^k = AGGREGATE_K(\{h_u^{k-1}, \forall u \in N(v)\})$$

At this time, h_N^k contains the neighbor information and the feature information of the adjacent nodes. h_u^{k-1} here is a sample of neighbor nodes, because in some cases some nodes may have one million or even ten million neighbors (a blogger may have ten million followers). Since the adjacent nodes are themselves an unordered set, there is no need to consider the validity of the sampling. Graphsage generally takes a fixed number of samples, but can take a variable number of samples for a particular problem. There are three kinds of aggregation functions in the formula, Mean aggregation function, pool aggregation function and LSTM aggregation, the formula is as follows.

Mean: $AGG = \sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|}$

Pool: $AGG = \gamma(\{Qh_u^{k-1}, \forall u \in \mathrm{N(v)}\})$

LSTM: $AGG = LSTM([h_u^{k-1}, \forall u \in \pi(N(v))])$

Mining interactive information between nodes at both ends of an edge. Because the information of the edge is closely related to the information of the nodes connected at the two ends of the edge, we use the eigenvectors of the nodes at the two ends of the edge to represent the features of the edge Vector splicing, average pooling, maximum pooling, or summation. The splice function can hold the information of two nodes better. Here we combine the aggregated information with the central node information.

$$\mathrm{CONCAT}(h_v^{k-1}, h_{N(v)}^k)$$

And then embed it with the following formula.

$$h_v^k = \sigma(W^k \cdot CONCAT(h_v^{k-1}, h_{N(v)}^k))$$

σ is a nonlinear activation function, W^k is a weight matrix, their combination can be understood as a single layer neural network. Multi-layer neural network can also be used

here, but single-layer neural network can get preferable results [9] . Each node in the Diagram performs the calculations described above, and each node obtains information about its neighbors and structure. Then the next round of aggregation and embedding, by this way, each node can integrate information far from the center node. The range of integration is related to the number of cycles. The range of integration is related to the number of cycles. After each round of calculations, the resulting features need to be standardized, and the formula is as follows.

$$h_v^k = h_v^k / \| h_v^k \|_2 . \forall v \in V$$

After the above calculation method can be obtained for each node and a certain depth of the adjacent nodes of the vector, this vector contains a certain depth of adjacent node information and the structure of these nodes. Then the vector of the target node is the final eigenvector. The probability of each class can be calculated by the function of Softmax.

$$Z_j = soft\max(h_v^K)$$

In this model, the Cross entropy is used to calculate the model loss, and the formula is as follows.

$$loss = -\sum_{c=1}^{|L|} y_c \log(z_c)$$

According to the model described above, the specific implementation process is shown in Algorithm 1.

Algorithm1: Supervised GraphSAGE

Input : Graph G(V, E, X) ;depth K; weight Matrices W^k ; non-linearity σ ; differentiable aggregator functions $AGGREGATE_k$

Output: Type of social network node Z

1 while Non-convergence do:

2 $h_v^0 = x_v, \forall v \in V$

3 for k = 1…K do

5 for $v \in V$ do

5 $h_N^k = AGGREGATE_K(\{h_u^{k-1}, \forall u \in N(v)\})$

6 $h_v^k = \sigma(W^k \cdot CONCAT(h_v^{k-1}, h_{N(v)}^k))$

7 end

8 $h_v^k = h_v^k / \| h_v^k \|_2 . \forall v \in V$

9 end

10 $Z_j = soft\max(h_v^K)$

11 Sum of cross-entropy loss for M known label samples

$$total\_loss = -\frac{1}{M}\sum_{m=1}^{M}\sum_{c=1}^{|L|} y_c \log(z_c)$$

12 Update model parameters using gradient descent

13end while

IV. EXPERIMENT

In order to verify that the model described above is effective in capturing the complex interactions between nodes in a social network, experiments are designed.

A. Compared to traditional methods

Logistic Regression, LR: Logical regression model is a traditional classification model, which is often used in the traditional social network node classification algorithm based on iteration. The model uses only the characteristics of the nodes as the input of the classifier, and does not use the structural information of the data.

Deepwalk: Deepwalk is a classic graph mining algorithm based on random walk. It uses graph data to learn the vector representation of nodes. The resulting vector representation can be applied to the classification of nodes in social networks and others In the graph mining task. This model only uses the network structure information of graph data, and does not use the characteristic information of nodes.

The results of LR and Deepwalk experiments on the Dataset are shown in Table 3. Compared with the LR model which only uses node information, the graph sage improves about 10 percentage points in the data set. This model can capture the structure information of social network effectively and achieve the node classification effect more effectively Compared with the Deepwalk model, which only uses network structure information, GraphSage model has been improved by more than 10% in the data set of popular micro blogs, which shows that GraphSage model can effectively integrate nodes and achieve better classification effect.

Model	Accuracy rate
LR	42.5%
DeepWalk	32.12%
GraphSage(GCN)	53.87%

Tab. 2 Result of GraphSage model and traditional

B. Comparison of different parameters

This part carries on the experiment separately to the Model K each layer sampling quantity adjustment. Because this experiment is a multi-classification problem, it needs a large number of samples in order to contain enough adjacent node information. The results show that the effect of classification is little improved, but the running time is much longer. Change the value of K to allow the model to integrate a wider range of information. There has been little improvement in classification, perhaps because, in social networks, distant users have had little impact on current users. The results are shown in Table 3.

K	S1	S2	S3	Rate
2	20	10	-	53.87
2	25	20	-	55.03
3	20	10	10	53.96

Tab. 3 Result of different parameters

C. New sampling method

Considering that the number of adjacent nodes in a social network varies greatly, better results may be obtained if the

number of samples is positively correlated with the number of adjacent nodes. $S_i = c + [N_i \times w]$, N_i is number of adjacent node, c is a constant, w is weight. In order to ensure the number of samples in a certain range, when the number of samples is greater than a value, take the value as the number of samples.

With reference to the parameters used above, c=5, w=0.001, max number of samples is 40. After using the new sampling method, the experimental results are improved slightly.

Method	Rate
Fixed number	53.87
Variable number	54.13

V. CONCLUSION

The classification of social network nodes has important application value in real life. The traditional application classifier based on iteration and label propagation Algorithm based on random walk do not fully consider the mutual information between nodes. Graphsage extracts higher level node features according to the interaction information with the neighboring nodes and the node's own attribute characteristics, and finally classifies the nodes.

This paper uses graphsage model to analyze social network, and compares it with traditional social network analysis method. Some parameters of the model were discussed, and some improvements were made in the sampling part of the model. There is still much room for improvement in this model. In the future, we will consider increasing the amount of data and other features to improve the effectiveness of the model.

ACKNOWLEDGMENT

This research was supported by National Key R&D Program of China(2018YFB1402905) and Shanghai. All support is gratefully acknowledged.

REFERENCES

[1] Eytan Bakshy, Dean Eckles, Rong Yan, and Itamar Rosenn. 2012. Social influencein social advertising: evidence from field experiments. InEC '12. ACM, 146–161.

[2] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, CameronMarlow, Jaime E Settle, and James H Fowler. 2012. A 61-million-person exper-iment in social influence and political mobilization.Nature489, 7415 (2012),295.K. Elissa, "Title of paper if known," unpublished.

[3] Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., and Sun,M. Graph neural networks: A review of methods andapplications.arXiv preprint arXiv:1812.08434, 2018.

[4] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. 2018.DeepInf: Social Influence Prediction with Deep Learning. InKDD. 2110–2119.

[5] Li J , Rong Y , Cheng H , et al. Semi-Supervised Graph Classification: A Hierarchical Graph Perspective[J]. 2019.

[6] Xu K , Wang L , Yu M , et al. Cross-lingual Knowledge Graph Alignment via Graph Matching Neural Network[J]. 2019.

[7] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Interface Prediction using GraphConvolutional Networks. Number Advances in Neural Information Processing Systems, pp.6533–6542, 2017.

[8] XU, K., HU, W., LESKOVEC, J., & JEGELKA, S. (2018). How powerful are graph neural networks? [J]. arXiv preprint arXiv: 1810.00826..

[9] W.L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs.arXiv preprint,arXiv:1603.04467, 2017.