HARP Pro: Hierarchical Representation Learning based on global and local features for social networks

Wei Zhang, Jing Yang[1], Fanshu Shang
College of Computer Science and Technology
Harbin Engineering University
Harbin, China

***Abstract*—The purpose of network representation learning methods is to learn the node low-dimensional representations to accomplish node classification, link prediction, network visualization, and so on. Most of the network representation learning methods cannot keep local structural features and global structural features, resulting in poor performance at classification tasks for social networks. To solve this problem, we propose HARP Pro, a novel method for learning latent node representations which can maintain lower-order and higher-order hierarchical structures. On the one hand, HARP Pro coarsens the graph based on the community detection method. It can preserve the relationships between bridge nodes and communities. On the other hand, it presents a graph coarsening method based on Degree to keep the relationships between nodes and their neighborhood. It can capture the lower-order hierarchical structure of the graph. Then it puts the hierarchical information into the network embedding methods iteratively like HARP. Finally, it obtains node representation vectors which integrate global and local structural features. Experimental results on CiteSeer and Blogcatalog dataset show that the performance of HARP Pro is better than HARP and the baseline methods, DeepWalk, LINE, and Node2vec. The results reveal that HARP Pro can sustain local and global structural features.**

Keywords-social network; network embedding; community; graph coarsening; hierarchical structure

I. INTRODUCTION

We are in the era of Big Data. Because of Big Data, there are countless ties between everyone and everything. So long as has the place which the ties exist, the network exists, such as social network, protein network, logistics network, communication network, and electricity network. Getting appropriate representations for nodes and exploring the network is important for content recommendation[12], information diffusion[13], disease-related genes prediction[14], and resource assignment[15]. Hence, network embedding has received considerable attention recently.

The social network is a kind of large-scale network which has massive, sparse data with noise. The complex network data is a major challenge to the traditional network analysis methods. Inspired by word2vec[9], DeepWalk[1] captures path information with random walks. It regards the paths as sentences and puts them into the model to learn the representations. It gets rid of the relational matrix. Thus, its time complexity is low. However, with the limitation of the walk length, DeepWalk can only get the local context information of nodes. Thus, representations it learns just reflect the local structure of the graph. Unlike the depth-first sequence of DeepWalk, Node2vec[3] combines depth-first strategy with breadth-first strategy, enriches the sample space. This method is still lumbered with the walk length. Therefore, the path information it gets is still limited in the neighborhood of nodes. LINE[2] tries to mine the node relationships with first order similarity and second order similarity to deal with the problem. But this method just expands the sample space to the second order area. The global structure cannot be captured by LINE.

Based on the above facts, HARP[4] learns the embedding in smaller graphs which are obtained by coarsening the original graph. In this way, the global structure can be embedded into the original graph iteratively. HARP thinks that the shape of the input graph is the global structure. Then it maintains this shape in the process of graph coarsening. Unfortunately, the external shape is unable to fully reveal the internal hierarchical structure.

Community structure is ubiquitous in social networks. It is a known higher-order hierarchical structure. The relationships between communities constitute the entire network. And inside the community, hub nodes and bridge nodes string together and support the communication in the community. Inspired by this knowledge, we propose HARP Pro, a general meta-method to maintain the lower-order and higher-order hierarchical structures of social networks for network embedding.

First, we proposed a local structure coarsening method based on Degree[10] to maintain the lower-order hierarchical structure. Then, we presented a global structure coarsening method based on a community detection method to keep the higher-order hierarchical structure. Based on these two methods, we can get a set of coarse-grained graphs. Finally, we embed the node representation of the higher-order graph into the lower-order graph to get the final network representation with global and local features.

II. RELATED WORK

HARP[4] method consists of three parts: graph coarsening, graph embedding, and graph representation refinement. The process of HARP is listed in Table I.

[1] Corresponding author: Jing Yang (email:yangjing@hrbeu.edu.cn).
DOI reference number: 10.18293/SEKE2021-145.

In Step 1, HARP provides two graphs coarsening strategies. One is edge collapsing. They introduce the edge collapsing algorithm[5] to coarsen the equivalent edges. This strategy can keep the peer-to-peer structure. The other is the star collapsing. This strategy merges the leaf nodes to sustain the shape of the star topology.

TABLE I. HARP

Input: The original network $G_0=(N_0, E)$; Arbitrary graph embedding algorithm EMBED() **Output:** node representations of G_0, φ
Step 1. Coarsen G_0 to get smaller graphs $G_1, G_2, \dots, G_L$. Step 2. Put G_L in EMBED() to get the node representations of G_L, φ_{GL}. Step 3. **For** i=L-1 to 0 **do** Step 4. Set φ_{GL} as the input embedding of network G_i, and learn the node representations φ_{GL}; Step 5. **end for** Step 6. **return** φ_{GL}

III. HARP PRO

HARP Pro consists of three parts: graph coarsening, graph embedding, and graph representation refinement. It is the same with HARP. The difference between HARP Pro and HARP is the process graph coarsening. In order to capture the lower-order and higher-order hierarchical structures, we develop two kinds of graph coarsening methods.

As we have mentioned above, the community is the higher-order hierarchical structure of social networks. From a macro perspective, communities and the relationships between them form a giant net, which can make information spread through the whole network. Bridge nodes play a vital role in transmission between communities. From a micro perspective, the hub nodes and their followers form the community. Then the neighborhood of the community member is the lower-order hierarchical structure that we want to capture. To maintain these two kinds of hierarchical structures, we need to coarsen the network based on hub nodes and bridge nodes. Hub node owns a large number of followers. We can find them with Degree. Hence, we present a local structure coarsening method based on Degree.

A. A local structure coarsening method based on Degree

The major steps of the graph coarsening method for local structure are listed in Table II.

First, we lock the hub node with Degree. Then, we start the process of coarsening with these nodes. By merging the hub node and its neighbors, we can coarsen the first-order neighborhood structure of the original graph G_0 and get a new coarse-grained graph G_1. Running this algorithm on graph G_1, we can coarsen the second-order neighborhood structure of the original graph G_0. The output of the algorithm is the new coarse-grained graph G_2.

B. A global structure coarsening method based on community structure

Utilizing the above method, we can sustain the local structure in the coarse-grained graphs. For the global structure, we introduce Louvian[6], a community detection method to get the community structure of the original graph. Louvian believes that there are more links inside the community. And outside the community, links are more sparse.

Firstly, the community structure is employed to find bridge nodes. Then, we keep them and set them as the new nodes in the new coarse-grain graph. The rest nodes of the community will be merged, and regard as the new node in the new graph. In this way, we can obtain the higher-order hierarchical structure of the original graph. This process is listed in Table III.

TABLE II. COARSENING METHOD FOR LOCAL STRUCTURE

Input: The network G_j. **Output:** This coarse-grained graph, G_i.
Step 1. Sort the nodes in G_0 with Degree in descending order, and put the result in the list, DL; Step 2. **For** each node v in DL **do** Step 3. Compare the degree of node v, dv with the max degree of its neighbors, dv_m; **If** $dv > dv_m$ **do** Step 4. Combine node v with the reset of its neighbors, set them as a new node in G_i, and remove these node from DL; Step 5. **If** $dv=dv_m$ **do** Step 6. The neighbor nodes with the max degree will be kept in in G_i. Combine node v with the reset of its neighbors, set them as a new node in G_i, and remove these nodes from DL; Step 7. **return** G_i

TABLE III. COARSENING METHOD FOR GLOBAL STRUCTURE

Input: The original network G_0; **Output:** This coarse-grained graph, G_3.
Step 1. Use Louvian to get the community attribute of each node; Step 2. **For** each community com_i **do** Step 3. Find the bridge nodes in community com_i, set them as new nodes in G_3; Step 4. Merge the rest nodes of community com_i, set them as a new node in G_3; Step 5. **return** G_3

C. The Framework of HARP Pro

Based on Table II and Table III, we can get three coarse-grained graphs, G_1, G_2, and G_3. After this procedure, we can start the graph embedding and graph representation refinement. The framework of HARP Pro is listed in Table IV.

In Step 1, we run the two graph coarsening method on the original network G_0 respectively, then we can get G_1 with the first-order structure, G_2 with second-order structure, and G_3 with community structure. In Step 3, we start with G_3 to learn its node representations based on the embedding method. In Step 4- Step 7, we prolong and refine the representations. If a node in G_3 also appears in G_2, we replace this node representation in G_2 with G_3's. Otherwise, node a in G_2 is merged into node b in G_3, then a should has identical weights in word2vec as b. Then the new representation of G_2 is obtained. We go through the same process on G_1 and G_0. Finally, we can get new node representations of G_0.

TABLE IV. HARP PRO

Input: The original network, G_0; Arbitrary graph embedding algorithm, EMBED(). **Output:** The node representations of G_0, $\boldsymbol{\varphi}$
Step 1. G_0, G_1, G_2, G_3←GRAPHCOARSENING(G_0); Step 2. **Initialize** φ'_{G_3} by assigning zeros Step 3. $\varphi_{G_3} \leftarrow \text{EMBED}(G_3, \varphi'_{G_3})$; Step 4. **For** i =2 to 0 **do** Step 5. $\varphi'_{G_i} \leftarrow \text{PROLONGATE}(\varphi_{G_{i+1}}, G_{i+1}, G_i)$; Step 6. $\varphi_{G_i} \leftarrow EMBED(G_i, \varphi'_{G_i})$; Step 7: **end for** Step 8: **return** the node representations of G_0, φ

IV. RESULTS AND DISCUSSIONS

A. Data Preparation

Two real-world networks are used to evaluate the performance of different methods. They are CiteSeer[7] and BlogCatalog[8].

CiteSeer is a kind of citation network. In this network, nodes represent the authors of papers. Edges are reference relationships between papers. The labels indicate the research area of the paper. Papers in this network are classified into six kinds: AI, IR, ML, DB, Agents, and HCI.

BlogCatalog is a social network of the BlogCatalog website. Bloggers are the nodes. Edges represent the social relationships between users. The labels indicate the interests of bloggers. The interests are divided into 39 classes.

The information about these two datasets is shown in Table V.

TABLE V. STATISTICS OF NETWORK DATASETS

Network	***N***	***E***	***C***	***T***
CiteSeer	3312	4732	6	Classification
Blogcatalog	10312	333983	39	Classification

N, the number of nodes; E, the number of edges; C, the number of categories; T, multi-label classification task.

B. Baseline Methods

We use the following network embedding methods to conduct three parallel experiments:

DeepWalk——DeepWalk uses random walk to capture the path information of nodes. It treats route sequences as sentences and puts them into the Skip-gram model to learn the node embeddings.

LINE——LINE maps nodes to the vector space based on the density of node relationships. It combines the first-order relationship with the second-order structure to project nodes with a strong connection to a similar location based on the objective function. The Skip-gram model is applied to solve the objective function.

Node2vec——To improve DeepWalk, Node2vec change the way of random walk. By tuning the return parameter p and in-out parameter q, it can explore the node neighborhood with DFS and BFS. The Skip-gram model is also used in this method.

C. Parameter Settings

For each baseline method, we embed it with HARP and HARP Pro to compare the classification performance. Hence, we conduct three parallel experiments: (1) DeepWalk, HARP(DeepWalk), and HARP Pro(DeepWalk); (2) LINE, HARP(LINE) and HARP Pro(LINE); (3) Node2vec, HARP(Node2vec) and HARP Pro(Node2vec).

- Deepwalk Group: The number of random walks is set as 40. The walk length t is 10 and the window size is 5. We set the representation size d as 128.
- LINE Group: The representation size d is set as 64, and the iteration time is 50.
- Node2vec Group: The number of random walks, walk length, window size and representation size are the same as the DeepWalk group. We set the in-out parameter and the return parameter as 1.0.

D. Graph Coarsening

No matter how large the network is, HARP Pro just needs three times coarsening. One time is to get the global structure. The second time is to capture the local feature of the original graph. But for HARP, the coarsening time depends on the scale of the network. For CiteSeer dataset, HARP needs 18 times to reach the preset size. And for BlogCatalog, 24 times coarsening is needed to remain the shape of the network unchanged. We used Gephi, a network topology visualization tool to draw the graphs that we got by HARP Pro.

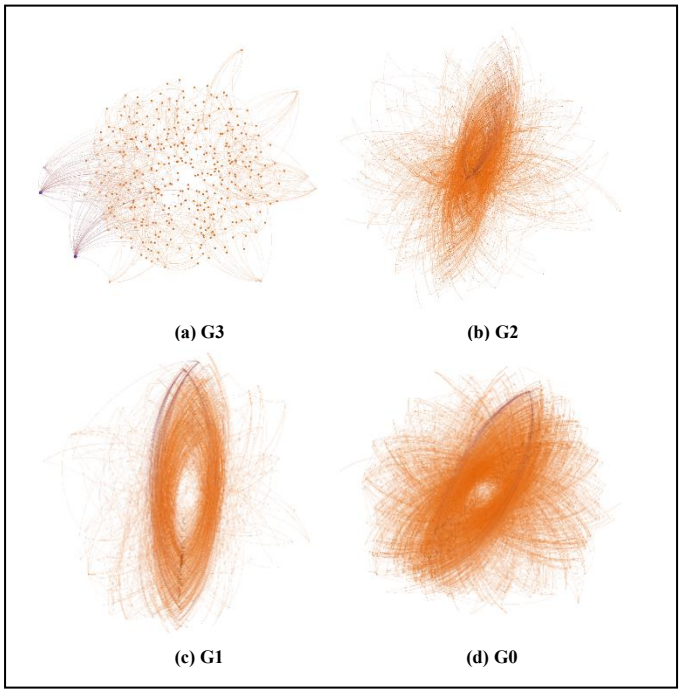

Figure 1. The graph coarsening effect of HARP Pro on CiteSeer dataset.

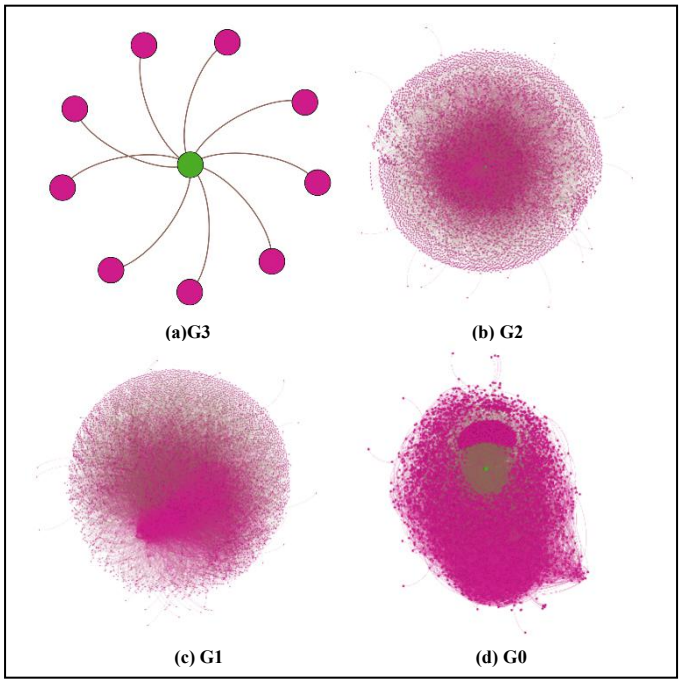

Figure 2. The graph coarsening effect of HARP Pro on BlogCatalog dataset.

Fig. 1 and Fig. 2 show the graph coarsening effect of HARP Pro on the two datasets. G_0 is the original graph. G_1 and G_2 are the subgraphs that we got based on Table II, and G_3 is the subgraph with the community structure that we got based on Table III.

As shown in Fig. 1, CiteSeer network is formed with multiple ring structures. With two times graph coarsening based on Algorithm 2. The longitudinal ring structures are kept in G_1 and G_2. The ring structures in the other orientation are in decline. In G_3, a few ring structures remain. From G_0 to G_3, the scale of the network drastically reduced.

In Fig. 2, BlogCatalog dataset demonstrates a star topology. With the reduction in the number of nodes, this feature becomes more obvious.

In general, the graph coarsening method in HARP Pro can maintain the shape of the original graph.

E. Multi-label Classification

Classification is the most common application of the network embedding method. In order to observe the performance of the classification task, we choose some nodes and their labels randomly as the training dataset. The proportion of the training dataset is varied from 10% to 90% in this experiment. To predict the labels of the remaining nodes, we train a one-vs-rest logistic regression model with L2 regularization. The logistic regression model is implemented by LibLinear[11].

Fig. 3 and Fig. 4 report Marco F1 scores of the three parallel experiments.

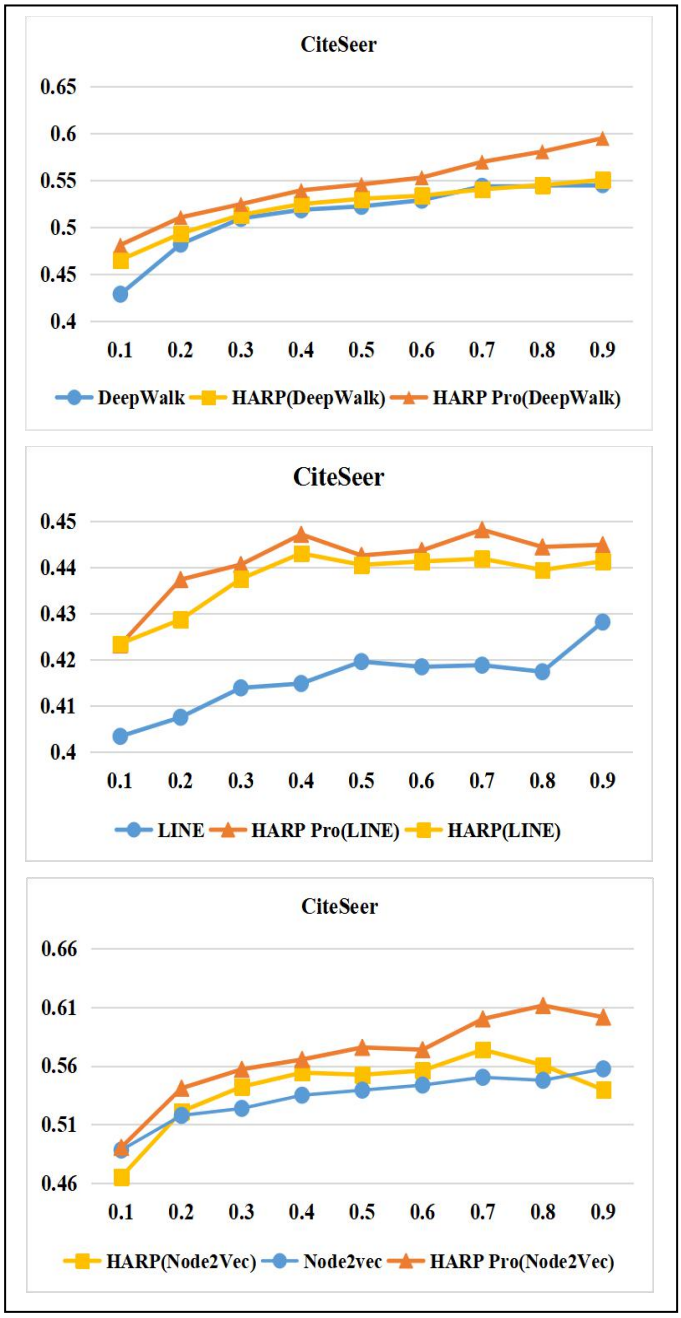

Figure 3. Detailed multi-label classification result on CiteSeer dataset.

CiteSeer. For the group of DeepWalk, as the size of labeled data increases, the Marco F1 score of three methods ascends step by step basically. By contrast, our method owns the highest score all the time. When the ratio of labeled nodes is small(10%-50%), the scores of the three methods are close. But our method is still higher than HARP and Deepwalk. The relative gain of our method is over 1%. When the ratio of labeled data is big(60%-90%), the advantages of our method are obvious. The relative gain of our method is over 5% with 90% labeled data. For the group of LINE, the trend of the curve is similar to the DeepWalk group. The difference is that the score of our method and HARP is relatively close. But our method also consistently outperforms with the relative gain 1%.

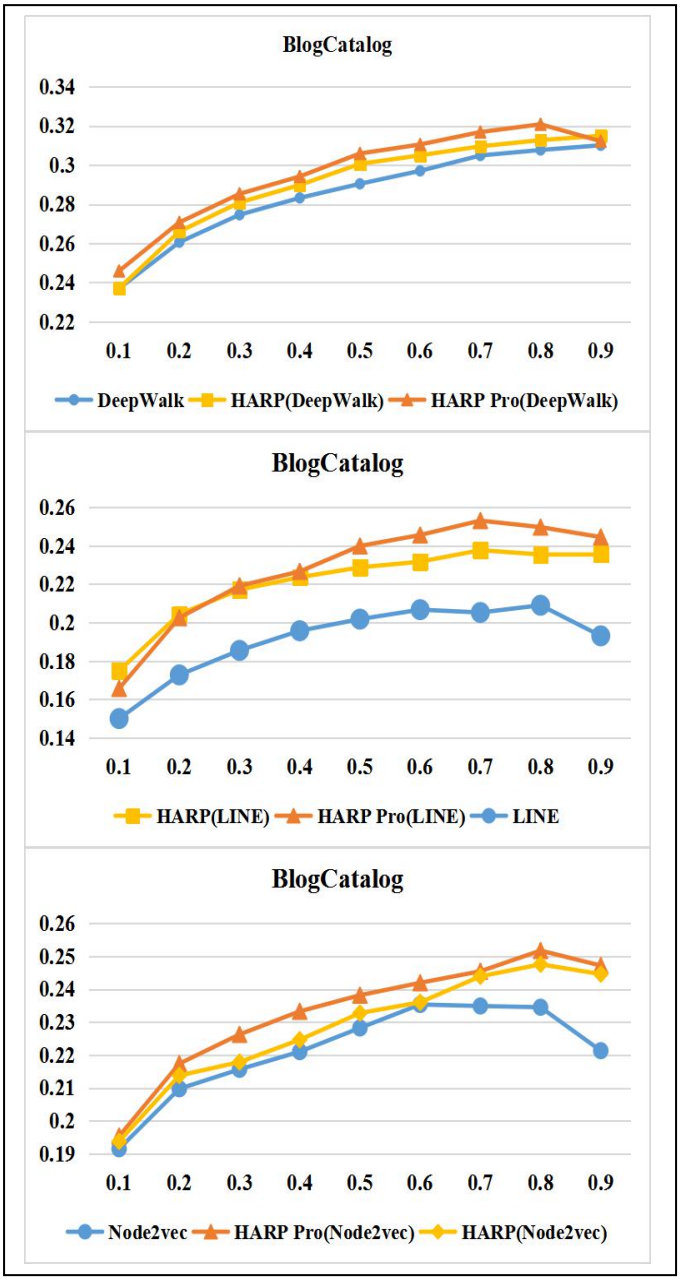

Figure 4. Detailed multi-label classification result on BlogCatalog dataset.

For the group of Node2vec, our advantage is more obvious. When the ratio of labeled data is small(10%-60%), the relative gain of our method remains close to 2% through that period. When the ratio of labeled data becomes bigger, HARP shows a downward trend. However, the rising rate of our method is even greater. In general, our method maintains the leading position in three parallel experiments.

BlogCatalog. For the group of DeepWalk, both three methods show the tendency to ascend. The score of the three methods is close. But the relative gain of our method is 1% with 10% and 80% labeled data. As for the LINE group, the situation is a bit different. When the ratio of labeled data is small(10%-40%), the scores are close. But when the ratio of labeled data becomes bigger, the difference between our method and HARP becomes wider. The relative gain of our method is over 3% with 70% labeled data. HARP and our method reveal the trend of a fast increase in Node2vec group. Our method also performs better than Node2vec and HARP. Generally, the accuracy of HARP Pro in classification is better performance than HARP and the other baseline methods.

F. Discussion

Compared with Fig. 3 and Fig. 4, we can see the advantage of our method in CiteSeer is more obvious than Blogcatalog for the multi-label classification tasks. As shown in Fig. 2, Blogcatalog data owns plenty of star structures. Star structure is not a typical community structure. Because the links inside and outside it are both sparse, the star structure cannot be distinguished by the community detection method. The global structure cannot be kept in the graph completely. As for CiteSeer, the distribution of links between nodes is relatively uniform. Community structures can be well identified with Louvain. Thus, the advantage of HARP Pro on BlogCatalog is smaller.

In summary, HARP Pro is good at learning node representation of the network with significant community structures.

V. CONCLUSION

From a microcosmic perspective, nodes and their neighbors form their own communities. From a macro perspective, the entire network is constituted by a number of communities. Inspired by this fact, we propose a networking embedding method based on community structure and neighborhood structure, HARP Pro. HARP Pro starts with Degree to coarsen the first-order and second-order neighborhood structure. It maintains the local hierarchical information in the subgraph. Then, HARP Pro merges nodes with community features. It captures the high-order hierarchical structure. Finally, it obtains node representation vectors by capturing global and local structural features. Experimental results on CiteSeer and Blogcatalog dataset show that HARP Pro performs better than HARP and the other baseline methods.

Because of the instability of the community detection method, we introduce more uncertain factors to the model. In the future, we would like to find a more stable approach to capture the community feature. And we will take into account the node influence and the role of the node to learn the network representation.

ACKNOWLEDGMENT

This paper is supported by the National Natural Science Foundation of China (No. 61672179, 61370083 and 61402126), the Natural Science Foundation of Heilongjiang (No. F2015030), the Distinguished Young Scholars of Heilongjiang (No. QC2016083), the Postdoctoral Science Foundation of Heilongjiang (No. LBH-Z14071).

REFERENCES

[1] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '14). Association for Computing Machinery, New York, NY, USA, pp. 701-710, 2014.

[2] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. LINE: Large-scale Information Network Embedding. In Proceedings of the 24th International Conference on World Wide Web (WWW '15). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, pp. 1067‑1077, 2015.

[3] Aditya Grover and Jure Leskovec. Node2vec: Scalable Feature Learning for Networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, pp. 855‑864, 2016.

[4] Chen, H., Perozzi, B., Hu, Y., & Skiena, S. HARP: Hierarchical Representation Learning for Networks. AAAI, 2018.

[5] Hu, Y., Efficient, High-Quality Force-Directed Graph Drawing. Mathematica journal, vol. 10, pp. 37‑71, 2006.

[6] Blondel V D , Guillaume J L , Lambiotte R , et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics Theory & Experiment, 2008.

[7] Sen, P.; Namata, G. M.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. Collective classification in network data. AI Magazine, vol. 29, pp. 93‑106, 2008.

[8] Lei Tang and Huan Liu. Relational learning via latent social dimensions. In Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09). Association for Computing Machinery, New York, NY, USA, pp. 817‑826, 2009.

[9] Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR. 2013.

[10] Bonacich P F. Factoring and weighting approaches to status scores and clique identification[J]. Journal of Mathematical Sociology, vol. 2, pp. 113-120, 1972.

[11] Fan R E, Chang K W, Hsieh C J, et al. LIBLINEAR: A Library for Large Linear Classification[J]. JMLR. vol. 9, pp.1871-1874, 2008.

[12] Li J, Yang G. Network embedding enhanced intelligent recommendation for online social networks[J]. Future Generation Computer Systems. vol. 119, pp. 68-76, 2021.

[13] Bl A, Dpa B , Yl A , et al. Multi-source information fusion based heterogeneous network embedding[J]. Information Sciences. vol. 534, pp. 53-71, 2020.

[14] Xiang J, Zhang N R, Zhang J S, et al. PrGeFNE: Predicting disease-related genes by fast network embedding[J]. Methods, 2020.

[15] Hejja K , Hesselbach X. Online power aware coordinated virtual network embedding with 5G delay constraint[J]. Journal of Network and Computer Applications. vol. 124, pp. 121-136, 2018.