Academic Network Representation Learning Based on Metapath Tree

Wei Zhang\textsuperscript{1,2}, Ying Liang\textsuperscript{1} and Xiangxiang Dong\textsuperscript{1,2}

\textsuperscript{1} Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
\textsuperscript{2} University of Chinese Academy of Sciences, Beijing, China
Email: yida_915@163.com, liangy@ict.ac.cn, theabelx@163.com

Abstract. Network representation learning aims to use low-dimensional dense vectors to represent nodes in the graph, which can reflect the graph structure and can be used in a variety of machine learning tasks. The academic network contains richer information, which most of the current methods are unable to capture. This paper proposes a method to get better vectors in the academic network. This method first uses the metapath tree to guide the random walk process, and adds a sampling process to preserve multiple metapath information. The vector representation of the nodes is obtained by training using the skip-gram model on the academic network. Experiment results show that the proposed model outperforms several traditional network representation learning models in multi-label classification and clustering tasks.

1. Introduction

Graph data, also called network data, is an important form of data on the Internet. Compared with traditional relational data, graph data can express richer semantic information. Network representation learning is a very important task in graph data mining [1]. Taking node representation learning as an example, its main purpose is to obtain a dense vector representation of each node in the graph through model training. Compared with the traditional graph adjacent matrix, the node vectors greatly reduces the amount of data. The node vectors can be easily applied in machine learning tasks, such as node classification [2], community discovery [3], information retrieval [4], recommendation system [5] and other fields.

With the advancement of neural network research, especially the rapid development of deep learning, network representation has made great progress. Inspired by the power of the Word2vec model[6][7] in the field of word embedding, some authors proposed random-walk-based network representation learning models, such as Deep Walk[8], node2vec[9]. Tang et al. [10] have proposed LINE based on shallow neural network, which can be applied to the training of large-scale networks. In addition, inspired by CNN’s effects in the image field, Thomas et al. [11] have proposed GCN model, which carefully designed a convolution operation for graph data. There are also a series of studies [12][13] based on this work, which have achieved good results.

However, these models are all aimed at homogeneous networks and cannot capture rich information in heterogeneous networks such as academic networks. The types of nodes in a heterogeneous network are different, resulting in the diversity of information. Some studies have been done on network representation learning based on metapath in the network. Dong et al. [14] have proposed a metapath-based representation learning method. Sun et al. [15] have proposed a method combining metapath and meta-graph. Although these studies have achieved good results in some datasets, it is still difficult to deeply explore the deep semantics hidden in the metapath.
To solve this problem, this paper proposes an academic network representation learning method based on metapath tree. By designing random walk and sampling rules in the metapath tree, network structure information and metapath information between nodes are more fully captured. The dataset is trained by using the skip-gram model, and finally the low-dimensional real vector representation of the node is obtained.

2. Preliminaries

2.1. Network and Metapath

We first define networks and metapaths.

**Definition I. Network.** A network is a directed graph $G = (V, E, T, \varphi)$, where $V = \{v_1, v_2, \ldots, v_n\}$ indicates the set of nodes, $E = \{e_{ij}\}_{1 \leq i, j \leq n}$ indicates the set of edges connecting each of $v_i$ and $v_j$, $T$ represents the set of types to which the node belongs. $\varphi : V \rightarrow T_v$ is a type mapping function for nodes, each node $v_i$ is mapped to a single node type in $T_v$, i.e. $\varphi(v_i) \in T_v$. When $|T_v| > 1$, indicating there’s more than one type of node, like an academic network, the network is also called as heterogeneous network, otherwise it is called as homogeneous network.

**Definition II. Metapath.** In a heterogeneous network, a metapath $\rho$ is defined as a sequence of edges and nodes connecting two nodes $v_k$ and $v_l$: $\rho = v_k \xrightarrow{e_{kl}} v_{k+1} \xrightarrow{e_{kl}} \cdots \xrightarrow{e_{kl}} v_l$. Starting from different nodes, we can generate a variety of instances of metapath $\rho$ in heterogeneous network.

In this paper, we use node type sequence to represent the metapath, that is $\rho = \varphi(v_k) \varphi(v_{k+1}) \cdots \varphi(v_l)$. The example of metapaths APA and PCP in academic network are shown in figure 1.

![Figure 1. Semantic contained in metapaths](image)

Figure 1 contains multiple nodes, including author (A), paper (P), conference (C). We regard academic network as an undirected graph, so there’s no arrow in Figure 1. An edge between node A and P, denoting author A writes paper P, can be viewed as that the paper P is written by the author A. Different metapaths convey different semantic information. APA represents that two authors simultaneously publish a paper, they have a cooperative author relationship; PCP represents that two papers belongs to the same conference proceeding.

2.2. Problem Definition

This paper aims to learn nodes’ representation on academic network, so we define the problem as follows.

**Definition III. Academic Network Representation Learning.** Given an academic network $G = (V, E, T, \varphi)$, where $T_v = \{A, P, C\}$. The goal of academic network representation learning is to learn a mapping function $\phi(V) : V \rightarrow \mathbb{R}^d$, which maps each node to a $d$ dimensional vector, where $d << |V|$.

The vectors can further be used in several downstream machine learning tasks.

2.3. Metapath Tree

To make full use of metapath information, we propose the following definitions.

**Definition IV. Metapath Tree.** A metapath tree is a tree structure composed of $m$ metapaths of the same type.
Tree - \rho = \{\rho_1, \rho_2, ..., \rho_m\}
\text{s.t.} \begin{cases} \rho_{1,j} = \rho_{2,j} = ... = \rho_{n,j} & (i) \\ \phi(\rho_{1,j}) = \phi(\rho_{2,j}) = ... = \phi(\rho_{n,j}) & (ii) \end{cases} (1)

Where \( \rho_{i,j} \) is the j-th node in i-th metapath. Constraint (i) restricts all metapaths to share the starting node, and Constraint (ii) restricts the same-level nodes in the metapath tree to be the same type.

Figure 2 shows the metapath tree Tree-CPA composed of 4 metapath, which indicates that there are multiple papers in the conference, and each paper contains multiple authors. Tree-CPA describes multiple metapath relationships simultaneously. Compared with the metapath CPA, more information can be used in the academic network representation learning modeling process.

**Figure 2.** Multiple metapath in metapath tree

**DEFINITION V. Symmetric Metapath Tree.** In real-world academic network, authors may propose different papers in different conferences. Symmetric metapath tree \text{CPAPC} describes such information, as is shown in figure 3. It is formed by two metapath tree Tree-CPA sharing same authors.

**Figure 3.** Symmetric metapath tree example

This tree represents the relationship between conferences, papers, and authors in an academic network. We can see that the authors \( A_1 \), \( A_2 \) and \( A_3 \) write a total of three papers in different partnerships, which are published in conferences \( C_1 \) and \( C_2 \).

3. **Methodology**

3.1. **Random Walk Based on Metapath Tree**

In the field of network representation learning, Perozzi et al. first proposed the concept of random walk in their DeepWalk model, trying to approximate the local and global features of the node in the graph data through a series of walk sequences of a node. This paper uses the symmetric metapath tree \text{CPAPC} to guide the process of random walk. The algorithm walks in the tree based on transition probability. For example, if the current node is \( v_i \), then the next hop is \( v_{i+1} \). The transition probability is defined as:
\[ P(v_{i+1} | v_i) = \begin{cases} 
\frac{1}{|N_v|} & e_{v,v_{i+1}} \in E, \phi(N_v) = T^v_{\text{meta}} \\
0 & \text{otherwise} \end{cases} \] (2)

Where \( N_v \) are the neighbor nodes of \( v_i \), \(|N_v|\) is the number of neighbor nodes, \( T^v_{\text{meta}} \) indicates the expected node type of next hop node based on metapath tree \( \text{meta} \) (CPAPC in this paper), when the current node is \( v_i \). If the current node is at the conference node \( (C) \), the next node must be at the paper node \( (P) \).

When a neighbor node satisfies the type requirement, there is a possibility of walking to the node; otherwise, it is impossible. The design of the symmetric metapath tree ensures that the walking process can be repeated until the preset walk length is satisfied.

3.2. Siblings Sampling

The definition of the transition probability enables walking process to walk along the predefined metapath tree, but the obtained sequence of nodes contains only limited information. For nodes in Figure 3, it is possible to obtain a sequence like \( C_P A_P A_P C \), but this sequence does not fully reflect the nature of the corresponding metapath tree. Therefore, for the intermediate nodes of the symmetric metapath tree, this paper samples \( k \) nodes with the probability defined as:

\[ P_k(v_{i} | v_i) = \frac{1}{|S_{v_i}|} \phi(S_{v_i}) = \phi(v_i) \] (3)

Where \( v_{i} \) is the sampled node, \( S_{v_i} \) are the sibling nodes of \( v_i \). The parent node corresponds to the node of previous step \( v_{i-1} \).

For the purpose of prioritizing authors, we only sample the paper nodes once to get one additional paper node. The number of sampled nodes for an author node is \( k \), which correspond to the coauthors. All coauthors of a paper will be sampled if the number of coauthors is less than \( k \).

The constraint guarantees that the same type of nodes have been sampled. The sampling process is only valid at the intermediate nodes of the symmetric metapath, i.e. the nodes \( PAP \) in \( CPAPC \). After sampling, the new metapath information will be added to the original walk sequence, and its corresponding relationship and meaning are shown in Table 1.

| Sample Node | Parent Node | Metapath info | Meanings |
|-------------|-------------|---------------|----------|
| P           | V           | PVP           | Conference proceeding |
| A           | P           | APA           | Co-authorship |
| P           | A           | PAP           | Papers of an author |

**Figure 4. Example of nodes after sampling**

Figure 4 shows the nodes after sampling. Nodes with shadows are obtained through the sampling process. Sampled paper \( P_{4i} \) is another paper in conference \( C_1 \), which is also the sibling node of \( P_i \) in \( C_1 \).
Co-authors in paper $P_1$, i.e. $A_i$ and $A_j$ are sampled. In addition, paper $P_2$ is sampled because it’s written by the same author of $P_1$.

3.3. Skip Gram in Academic Network

The sampled nodes form a graph structure. In this paper, the graph structure is transformed into a one-dimensional node sequence, and then a skip-gram model is applied. The sampled nodes of the same type will be randomly distributed on both sides of the original node. Taking figure 4 for an example, the node sequence before sampling is $C_P A_1 P_2 C_j$, i.e. the author node $A_j$ in the sequence can be replaced by $A_1 A_2 A_3$, denoting the sampled nodes $A_i$ and $A_j$ are randomly inserted into the left and right side of $A_j$.

In academic network $G = (V, E, T, \phi)$, the skip-gram model implements node modeling by maximizing the co-occurrence probability of nodes within a certain size window:

$$\max \sum_{v \in \text{v}} \sum_{v \in N_v^w} \log p(v_i \mid v_o)$$

Where $N_v^w$ represents the context node of node $v_i$ within the window size $w$. $p(v_i \mid v_o)$ is the probability of predicting the surrounding node $v_i$ according to the central node $v_o$. $V$ represents all the nodes that appear in the walk sequence, and the model needs to maximize the sum of the probabilities of all nodes co-occurring.

This probability can be maximized by using softmax: $p(v_i \mid v_o) = \frac{e^{v_i \cdot w_o}}{\sum_{w \in V} e^{v_i \cdot w_o}}$, where $v_i$ is the vector of node $v_i$, $w_o$ is the vector of node $v_o$.

3.4. Optimization

Inspired by the Negative Sampling technology in word2vec, negative sampling can be used instead of the Softmax process described above. Negative sampling tries to maximize:

$$\log \sigma(w_i \cdot w_o) + \sum_{s=1}^{\text{Samples}} \log p(u)$$

Where $w_i$ represents the central node in the window, the left side of the formula represents the co-occurrence probability of maximizing $w_o$ with another node $w_i$ in the window, and the right side represents the co-occurrence probability of minimizing $w_o$ and the sampled negative node $w_s$. Activation function is $\sigma = \frac{1}{1+e^{-w_i \cdot w_o}}$, $p(u)$ represents the nodes sampled according to the node frequency, and the number of negative samples is set to 5.

4. Experiments

4.1. Data and Evaluation Models

This paper uses the Aminer [16] academic network to verify the correctness of the algorithm. These data are available on the Aminer official website, including data on authors, papers, conferences, and relationship data on author-paper, paper-conference. The network contains 1,693,531 authors, 3,194,405 papers, and 3,883 conferences.

In this paper, Deep Walk, node2vec, LINE, and metapath2vec (abbreviated as metapath in the tables and legends) are selected as the comparison models. Experiments are carried out in the same experimental environment and corpus, and the dimension of node vectors are set to 128.

We use Python to preprocess data and perform random walk and sampling. FastText [17] is used in the training process to implement skip-gram and negative sampling. The effectiveness of the model in
this paper is verified by three tasks: multi-label node classification, author clustering, and similarity search.

4.2. Experiment Results

4.2.1. Multi-label Node Classification

Node classification is a common task in networks and one of the important indicators to measure the quality of node vectors. This paper selects four representative computer domain categories [1. Computer Networks 2. Computer Vision 3. Database & Information Systems 4. Theoretical Computer Science.], and manually annotate 65 conferences with the corresponding label. Next, we label all the papers published in the labeled conference with the same label as the conference. For an author who has published multiple papers, a majority voting algorithm is applied to this situation, i.e. the author is labeled to be the same as the conference in which the largest number of papers published. Through this process, we obtain a total number of 77,153 labeled authors, and these data will be used for the experiment.

The data is multi-classified by using the Logistic Regression of the OnevsRest strategy, and divided into training and testing set. By using the stratified sampling strategy, the labels in the training set and testing set are identical to the original dataset. The proportion of the training set is controlled to increase from 10% to 90%. The performance is evaluated with the Macro-F1 and Micro-F1 values. The results are shown in Table 2.

![Figure 5. Parameter sensitivity in multi-label author node classification](image)

| Metric | Model  | 10%  | 20%  | 30%  | 40%  | 50%  | 60%  | 70%  | 80%  | 90%  |
|--------|--------|------|------|------|------|------|------|------|------|------|
|        | DeepWalk | 0.9527 | 0.9541 | 0.9545 | 0.9551 | 0.9553 | 0.9556 | 0.9556 | 0.9560 | 0.9555 |
| Micro  | node2vec | 0.9519 | 0.9532 | 0.9537 | 0.9540 | 0.9542 | 0.9548 | 0.9547 | 0.9546 | 0.9547 |
|        | LINE    | 0.9432 | 0.9462 | 0.9474 | 0.9481 | 0.9487 | 0.9489 | 0.9491 | 0.9489 | 0.9500 |
|        | This paper | 0.9548 | 0.9569 | 0.9575 | 0.9576 | 0.9581 | 0.9586 | 0.9587 | 0.9587 | 0.9585 |
|        | DeepWalk | 0.9512 | 0.9526 | 0.9530 | 0.9536 | 0.9538 | 0.9541 | 0.9541 | 0.9545 | 0.9540 |
| Macro  | node2vec | 0.9503 | 0.9516 | 0.9521 | 0.9525 | 0.9527 | 0.9532 | 0.9531 | 0.9530 | 0.9531 |
|        | LINE    | 0.9415 | 0.9444 | 0.9457 | 0.9463 | 0.9469 | 0.9471 | 0.9473 | 0.9472 | 0.9483 |
|        | metapath | 0.9564 | 0.9584 | 0.9590 | 0.9592 | 0.9597 | 0.9601 | 0.9603 | 0.9603 | 0.9601 |
|        | This paper | 0.9663 | 0.9682 | 0.9686 | 0.9689 | 0.9692 | 0.9693 | 0.9693 | 0.9704 | 0.9702 |

It can be seen from Table 2 that the performance of this paper is better in different training ratios, and the Macro-F1 value reaches 97.04%. Compared with traditional methods, the method proposed in this paper has a 1%-2% increase in Micro-F1 and Macro-F1 values, which proves that more information about academic networks can be captured.
**Parameter Sensitivity.** The crucial hyper-parameter k is mentioned in section 3.2, which means the number of sampled nodes. We conduct a sensitivity analysis on this. Figure 5 shows that higher value of parameter k result in better model effect over different training percent. Large k is costly for sampling process. To find a suitable k, a statistical experiment is performed on the data, as shown in Figure 6.

![Figure 6. Paper number with different coauthors](image)

Figure 6 shows the relationship between the number of papers and coauthors. More than 70% of the papers have 2 to 4 co-authors, and the proportion of single-author papers is only 10%, which indicates that the co-authorship is widely distributed in the academic network. In the paper, the default parameter k is set to 5, which covers over 90% of the papers and achieve promising results.

4.2.2. **Author Clustering**
Clustering reveals features that are hidden in certain data. The author nodes in the classification experiment are clustered using K-means++[18]. There are 4 cluster cores (corresponding to the 4 labels of the conferences). Use the AMI and NMI index to compare the effects of clustering, and then take the average value of 10 times experiments. The results are shown in Table 3. It can be seen from the table that in the unsupervised clustering task, the model of this paper is 3%-9% higher than the traditional model, which indicates that the author node vector quality obtained in this paper is better.

**Table 3. Author clustering results for different model**

|               | Deep Walk | node2vec | LINE   | metapath | This paper |
|---------------|-----------|----------|--------|----------|------------|
| **AMI**       | 0.7651    | 0.7648   | 0.7330 | 0.7952   | 0.8216     |
| **NMI**       | 0.7674    | 0.7671   | 0.7350 | 0.7970   | 0.8238     |

![Figure 7. Parameter sensitivity in node clustering](image)
**Parameter Sensitivity.** A similar sensitivity analysis is conducted on hyper-parameter $k$ over this task. As is shown in Figure 7, larger $k$ leads to better results, which means deeper relationship in coauthors can be captured through sampling process.

### 4.2.3. Similarity Search

In network representation learning, the similar nodes have similar distances in the vector space. This paper selects the list of Tsinghua University natural language processing team [http://nlp.csai.tsinghua.edu.cn/site2/index.php/en/people] and selects three author nodes. The KNN [19] algorithm is used to calculate the five authors most similar to the selected authors. The results are shown in Table 4.

| Rank | Authors            | Authors            | Authors            |
|------|--------------------|--------------------|--------------------|
| 1    | MaosongSun         | ZhiyuanLiu         | YankaiLin          |
| 2    | YabinZheng         | XianceSi           | WanxiangChe        |
| 3    | XinxiangChen       | YabinZheng         | ZhiyuanLiu         |
| 4    | XianceSi           | WanxiangChe        | MaosongSun         |
| 5    | WanxiangChe        | JingboZhu          | HelenO’Horan       |

From the list of laboratory personnel on the website, it can be seen that ZhiyuanLiu and MaosongSun are teachers of the same laboratory and have published a number of papers together. The search shows that the two authors have a high degree of similarity. CunchaoTu is a doctoral student in the laboratory. The author node with the most similarity is another doctoral student YankaiLin with similar research direction. As is seen from this example, similar authors have similar node vectors, which is in line with expectations.

### 5. Conclusion

This paper proposes an academic network representation learning method based on metapath tree. The algorithm walks and samples in the academic network according to the structure of the metapath tree, which contains more semantic information than the metapath. By using skip-gram and negative sampling, the performance of the algorithm is guaranteed, which makes the algorithm suitable for tasks of larger networks. The effect on classification and clustering is better than traditional models, indicating that the proposed algorithm can produce higher quality node vectors.

We plan to proceed from two aspects for further research. On the one hand, the algorithm model can be improved to reduce the information lost during the training process. On the other hand, we plan to consider more information in the academic network, such as the title and abstract on the paper, the basic information of the author, like academic influences, etc. Then we apply it to the network representation learning process to improve node vector quality.

### 6. Acknowledgments

This work is supported by the National Key Research and Development Plan (2018YFB1004700, 2016YFB0800403).

### 7. References

[1] Han, J., Pei, J., & Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.
[2] Jian, L., Li, J., & Liu, H. (2018). Toward online node classification on streaming networks. Data Mining and Knowledge Discovery, 32(1), 231-257.
[3] Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., & Yang, S. (2017, February). Community Preserving Network Embedding. In AAAI (pp. 203-209).
[4] Ganguly, D., Roy, D., Mitra, M., & Jones, G. J. (2015, August). Word embedding based generalized language model for information retrieval. In Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval (pp. 795-798).

[5] Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016, August). Collaborative knowledge base embedding for recommender systems. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 353-362). ACM.

[6] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

[7] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).

[8] Perozzi, B., Al-Rfou, R., & Skiena, S. (2014, August). Deepwalk: Online learning of social representations. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 701-710). ACM.

[9] Grover, A., & Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864). ACM.

[10] Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., & Mei, Q. (2015, May). Line: Large-scale information network embedding. In Proceedings of the 24th International Conference on World Wide Web (pp. 1067-1077). International World Wide Web Conferences Steering Committee.

[11] Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[12] Gao, H., Wang, Z., & Ji, S. (2018, July). Large-scale learnable graph convolutional networks. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 1416-1424). ACM.

[13] Li, Q., Han, Z., & Wu, X. M. (2018). Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning. arXiv preprint arXiv:1801.07606.

[14] Dong, Y., Chawla, N. V., & Swami, A. (2017, August). metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 135-144). ACM.

[15] Sun, L., He, L., Huang, Z., Cao, B., Xia, C., Wei, X., & Philip, S. Y. (2018, November). Joint embedding of meta-path and meta-graph for heterogeneous information networks. In 2018 IEEE International Conference on Big Knowledge (ICBK) (pp. 131-138). IEEE.

[16] Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., & Su, Z. (2008, August). Arnetminer: extraction and mining of academic social networks. In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 990-998). ACM.

[17] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

[18] Arthur, D., & Vassilvitskii, S. (2007, January). k-means++: The advantages of careful seeding. In Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms (pp. 1027-1035). Society for Industrial and Applied Mathematics.

[19] Dudani, S. A. (1976). The distance-weighted k-nearest-neighbor rule. IEEE Transactions on Systems, Man, and Cybernetics, (4), 325-327.