The Representation of Large-Scale Graph Based on Semi-Supervised Learning

YANG Tian-qing¹, YANG Jing-zong¹*, WANG Hui¹

¹ Information College, Baoshan University, Baoshan, Yunnan, 678000 , China
*Corresponding author’s e-mail: 2008010602@st.btbu.edu.cn

Abstract. With the development of artificial intelligence, the technology of graph representation has been widely used in graph structure data mining. The traditional graph representation mostly uses adjacency matrix. The representation of adjacency matrix is faced with two major challenges. One reason is extremely inefficient in the storage and computing of large graphs with the massive growth of data; the other is that the characterized data is difficult to retain the original structural information of graphs. Aiming at the above two points, an efficient graph representation method is proposed in this paper. The first-order similarity and second-order similarity of the graph are used as constraints to train the original graph structure data, and the nodes in the graph are mapped to a low-dimensional space, which greatly reduces the cost of storage space and computing while retaining the graph structure information. Experimental results show that the proposed method is significantly better than the Deepwalk and spectral clustering methods in different proportion of label data. Meanwhile, the proposed method can be applied to the semi-supervised learning task with a small number of labels, which provides convenience for the subsequent data analysis.

1. Introduction

With the expansion of entities in the real world and the popularization of information technology, the accumulation of data increases exponentially. How to quickly excavate the potential utilization value of these data has become the primary problem to be solved in the field of machine learning[1]. Compared with linear structure and tree structure, graph data has better ability to express the information in the real world [2]. The traditional graph representation mostly adopts adjacency matrix, which can express the graph structure data more intuitively, but with the improvement of data, it will bring huge storage space and computational complexity [3]. However, feature engineering is a very important link in machine learning, a good feature expression is related to the effectiveness of a model. Therefore, the study of graph representation is particularly important.

A disadvantage of traditional graph representation is the lack of general clustering or classification methods [4]. With the wide application of machine learning, a variety of methods using eigenvectors to classify graph data have been proposed one after another, such as K-means, Bayes classifier and support vector machine [5,6,7]. As early as 2007, a graph-embedding vector space model with n-dimensional features through the selection of graph structure and calculation of editing distance proposed by Kriesen et al. [8]. The author converts the graph structure into another quantitative representation through the similarity and distance of the graph, and this vector space model can be well applied in the classification algorithms such as K-NN [9] and support vector machine. However, due to the modification of the graph structure in the mapping process, the represented vectors often fail to maintain the original characteristic attributes of the graph structure, and the effect is obviously
reduced in the tasks with multiple classifications or fewer labels. With the development of neural network technology and natural language processing technology, some word embedding models such as Word2Vec and skip-gram appeared, which could transform words into embedding vectors while retaining semantic similarity between words [10,11]. In 2014, DeepWalk was proposed by B Perozzi et al.: an algorithm that captures the relationship between nodes through random walk and maps node sequences into vectors through Word2Vec model [12]. However, the random walk can only capture the relationship between adjacent nodes. For the graph data with many categories or fewer labels, a large amount of noise is often generated, leading to the unsatisfactory effect of the upstream task. In addition, there are some matrix decomposition techniques through feature extraction for characterization. B. NaserSharif in 2015 adopted the Laplace eigenmapping method, took the pre-designated graph as the noise observation of the ideal graph, and measured their differences in the objective function with the square Frobenius divergence. Through this method, a learning framework for graph construction and projection optimization at the same time was obtained [13]. Although these techniques have made important contributions to the field of graph representation, there is still room for improvement in the quality of representation, scalability and accuracy of upstream tasks.

In this paper, the first order and second order similarity of graph are used as constraint conditions to train the graph data. Firstly, a semi-supervised graph representation learning scheme is proposed under the constraint of first-order similarity, which enables nodes to map to a low-dimensional vector space while preserving the structure between nodes. Secondly, the second-order similarity information is captured by reconstructing the aggregated adjacency matrix. On the basis of the previous step, the second-order similarity is further used to make the nodes with similar structures maintain their consistency in the low-dimensional space, and finally the high-quality vector representation of each node is obtained. This model can not only maintain the characteristics of graph structure in the process of representation, but also obtain a better clustering structure in the semi-supervised learning task with fewer labels.

2. Related work

2.1. Graph representation
Graph structured data is widely used in real life and is represented by adjacency matrix in computer. But the adjacency matrix is a sparse matrix, and using it as a representation of a large graph can lead to dimensional disaster. As shown in Figure 1, graph representation is a process of mapping high-dimensional matrices into low-dimensional dense vectors, which can better apply graph structure data to machine learning algorithms. The original topology of the graph, the relationships between nodes, and the attributes of the subgraphs and individual nodes should be captured during the characterization process. At the same time, in most realistic application scenarios, there is a large amount of data including a large number of nodes and edges, and a good representation method needs to maintain high scalability especially when the global attribute of the network needs to be maintained. Secondly, the dimension selection of the whole representation space is particularly important. High-dimensional vectors have higher reconstruction accuracy, while low-dimensional vectors have lower space-time complexity. A good graph representation should have the above characteristics.

2.2. First order similarity
First-order similarity refers to the degree of direct connection between two nodes in a low-dimensional vector space [15]. As shown in Figure 2, we expect the two nodes with larger weights in the figure to be mapped to a low-dimensional space to keep closer in distance, while the nodes with infinite connections are farther away. Our goal is to preserve this similarity in the representational space. In the graph structure, the first-order similarity of its two nodes is expressed as follows:

\[ p(i, j) = \frac{w_{ij}}{W} \] (1)
In Equation (1), \( p(i,j) \) is the first-order similarity between node \( i \) and node \( j \); \( W_{ij} \) is the weight between node \( i \) and node \( j \); \( W \) is the sum of the weights of the whole graph. In the vector space, the first-order similarity between two nodes is represented by the product of Euclidean distance between two vectors and weight, as shown in Equation (2): Where \( v_i \) and \( v_j \) are low-dimensional space vector representations of node \( i \) and node \( j \).

\[
p(v_i, v_j) = W_{ij} (v_i - v_j)^2
\]

2.3. Second order similarity
Second-order similarity is a measure of the similarity degree of neighbor structures of two nodes [15]. If two nodes have the same number and number of neighbor nodes, it has a high second-order similarity value; otherwise, it has a low value.

As shown in Figure 3, nodes with the same color represent the same neighbor structure. These nodes are mapped to a low-dimensional space and should also keep the distance between their two node vectors closer. In the graph structure, the second-order similarity of two nodes is expressed as follows:

\[
p_2(i \mid j) = \frac{W_{ij}}{d_i}
\]

In Equation (3), \( p_2(i,j) \) represents the second-order similarity of node \( v_i \) and node \( v_j \); \( d_i \) is the degree of node \( v_i \). Two nodes in the vector space of second order similarity expressed as shown in Equation (4), The \( \mid V \mid \) is collection of all neighbor node for node \( i \).

\[
p_2(v_i \mid v_j) = \frac{W_{ij} (v_i - v_j)^2}{\sum_{k=1}^{\mid V_i \mid} W_{ik} (v_i - v_k)^2}
\]
3. model construction

3.1. First order similarity objective function
In the vector-based graph representation, the node vector is the most basic representation unit. Therefore, the quality of the whole model often depends on the quality of node vectors. In this paper, a low-dimensional random initialization vector is initially provided for each node, and the semi-supervised learning training is carried out by using the first-order and second-order similarity of the graph as constraints. The node vectors are trained by gradient descent and iterative updating methods according to the objective function. Constraint on the first-order similarity can preserve the information of the tightness between graph nodes in the low-dimensional space. The objective function using the first-order similarity as the constraint condition is shown as follows:

\[ L_1 = \sum_{i=1}^{V} [\gamma (v_i^* - v_i)^2 + \beta \sum_{j=1}^{V} w_{ij} (v_i^* - v_j)^2] \]  

(5)

In Equation (5) : \(v_i^*\) and \(v_i\) represent the values of the ith node vector before and after the correction respectively; \(\gamma\) and \(\beta\) is the equilibrium coefficient, which is used to ensure that the node vector after each correction is not completely close to the center of the node set; \(w_{ij}\) is the offset weight, so that the offset direction of the target node vector can be controlled, gradually moving towards the node vector with larger weight and away from the node vector with smaller weight. The optimal solution of the modified node vector \(v_i\) can be obtained by taking the partial derivative of \(v_i\) with Equation (5) and making its partial derivative 0. Its solution is shown in Equation (6):

\[ v_i = \frac{\gamma v_i^* + \beta \sum_{j=1}^{V} w_{ij} v_j}{\gamma + \beta \sum_{j=1}^{V} w_{ij}} \]

(6)

The iterative process is carried out until the convergence of Equation (5) ends. At this time, the node vector obtained is the final correction result under the constraint of first-order similarity.

3.2. Second order similarity objective function
In the process of representation, the second-order similarity of the graph structure can be constrained to ensure that the nodes are mapped to the low-dimensional space and still keep the information of the node's neighbor structure. Since each node of the second-order similarity calculation involves the traversal of the entire graph node, the computational complexity is very high, we adopt KL discretization to calculate the distance according to Equation (4), and the objective function is shown in Equation (7).

\[ L_2 = -\sum_{i=1}^{V} w_{ij} \log p_x(v_i | v_j) \]

(7)

Equation (7) is iteratively updated in the same way of gradient descent until it converges.

3.3. Model optimization
The optimization of this model involves two aspects, one is the dimension selection of node vector. Vectors with a lower number of dimensions have lower computational complexity, but the expressive power of node vectors is also reduced. On the contrary, a higher number of dimensions tends to have high quality representation vectors, but its computational complexity is also high. How to find the right measure between the two will directly affect the performance of the whole model. For this reason, we will conduct experiments in sequence from the dimension of 50 to 400 with an increase of 30 as a step in the experimental process, and make a comparative analysis of the appropriate dimension value between the model accuracy rate and the computational complexity.
The steps between Equations (5) and (6) are independent of each other. If two separate vector spaces are used for training, then the method of splicing the corresponding node vectors of the two vectors will greatly improve the computational complexity of the model. Therefore, we first carry out the training under the first-order similarity constraint and then carry out the training under the second-order similarity constraint on the results. The overall objective function is shown as follows:

$$L=\alpha L_1 + (1-\alpha) L_2$$  \hspace{2cm} (8)

In Equation (8) : $\alpha$ is the equilibrium coefficient, used to adjust the weight of $L_1$ and $L_2$.

4. Experiment and Analysis

4.1. Data set
The Cora dataset contains 2708 vertices, 5429 edges and 7 categories. 2933 vertices, 54903 edges and 13 categories of Mich datasets; A total of 19717 vertices, 44338 edges and 3 categories of PubMed datasets [16]. The Cora dataset is close to the Mich dataset in number of vertices, but the Mich dataset has a more complex graph structure. The PubMed dataset is a larger graph than the previous two because it has more vertices and edges.

4.2. Vector dimensions
The selection of a moderate vector dimension is the premise of graph representation model. We conducted experiments from 50 dimensions to 400 dimensions with a step size of 30 respectively. Then K-means clustering was carried out on the results, and finally the calculation time and classification accuracy of each dimension were counted. The results were shown in Figure 4 and Figure 5. It can be seen from Figure 4 that as the dimension of the vector increases, the time taken to calculate the representation vector of the corresponding dimension also shows an exponential growth. When the vector dimension reaches 290 dimensions, its growth rate is very large, which will make the calculation efficiency of the model low. Especially on the PubMed dataset, due to the large number of nodes and edges in the dataset. When the dimension is 320, the calculation time has reached 7645.5 seconds. The relationship between vector dimensions and model accuracy is reflected in Figure 5. When the vector dimension exceeds 250, the increase in accuracy on each dataset is low. Therefore, we chose 290 as the vector dimension of the subsequent experimental model as a compromise.

4.3. Comparison of classification
We compared the proposed model with the DEEPWALK and spectral clustering model in three datasets. Similarly, K-means was used to cluster the representation vectors obtained by the model in this paper and the Deepwalk model, and then the classification accuracy was calculated respectively.
The experiment was repeated for 5 times and the average value was taken as the test result, as shown in Table 1.

| model           | Cora     | Mich     | PubMed   |
|-----------------|----------|----------|----------|
| DeepWalk        | 0.9024   | 0.8832   | 0.8416   |
| Spectral clustering | 0.8321   | 0.7135   | 0.5434   |
| Our model       | 0.9145   | 0.9056   | 0.8966   |

The experimental results of Cora dataset show that the proposed model is not much ahead of the DeepWalk model. However, with the increase of graph structure complexity, the accuracy of DeepWalk and Spectral clustering models on Mich dataset and PubMed dataset has been greatly reduced. This is because the Deepwalk model uses a random walk to capture the information between nodes, but with the increase of the category and number of nodes, it tends to generate a lot of noise, leading to the degradation of the quality of node vectors. The model proposed in this paper is superior to the previous two in terms of overall results, and with the increase of graph size or complexity of graph structure, the model in this paper has strong stability.

4.4. **Comparison of semi-supervised learning**

In order to verify the semi-supervised learning ability of each model, we deleted the labels in the data set at different proportions, and conducted experiments successively from 10% to 90% of the retained labels. We conducted experiments in three data sets according to different proportion of reserved labels successively, and the test results are shown in Figure 6. As can be seen from the results of Cora dataset, when the proportion of labels in the dataset gradually decreases, the model in this paper is relatively stable compared with the other two models. Especially when the label proportion is reduced to 10% and 20%, the accuracy of the model in this paper can still be kept above 60%. Due to the large number of edges in the Mich data set, the structure of the whole graph is relatively complex and there are many nodes with similar local structures. Reducing the proportion of labels in a graph in such a dataset is a big challenge for multi-classification tasks. The Deepwalk model focuses on sampling the structure of neighbor nodes while neglecting the weight on the edges, which results in poor performance in large-scale complex graph structures, semi-supervised learning with small number of labels, and multi-classification tasks. Because the model in this paper retains both the order similarity and the second order similarity in the graph structure, it can still maintain high accuracy in the more complex graph structure. Especially in the experiment where the label proportion is less than 40%, the model in this paper can achieve high classification accuracy by fast node vector convergence clustering based on its second-order similarity. In the large-scale PubMed data set with graph structure, the model in this paper also has better results compared with the other two models in different label proportions.

![Figure 6. Comparison of results of semi-supervised learning.](image)

Therefore, the model in this paper can maintain a high consistency with the original graph structure in the node vector and is suitable for the calculation scenarios of large-scale complex graph structure.
5. conclusion
(1) Based on the first-order similarity and second-order similarity of graph structure, a graph representation method is proposed to map nodes to low-dimensional space vectors. In this way, the large-scale graph data can be mapped into low-latitu de space and the original structural characteristics of the graph can be well maintained, which can be well applied to various machine learning tasks.

(2) The tradeoff optimization is carried out for the dimension of node vector in terms of model representation quality and computational complexity, which can ensure that the node vector has a higher representation quality and its corresponding computational complexity is reasonable.

(3) The experimental results show that the proposed model is better than the Deepwalk and Spectral clustering model in each evaluation index. Moreover, it is more suitable scenarios for large scale complex graph structure and low label proportion.

Acknowledgments
Joint Special Project of Basic Research of Local Universities in Yunnan Province(2018FH001-125).
Joint Special Project of Basic Research of Local Universities in Yunnan Province(2019FH001-121).

References
[1] Cook, D. J. , & Holder, L. B. . (2000). Graph-based data mining. IEEE Intelligent Systems, 15(2), 32-41.
[2] Aggarwal, C. C. , & Wang, H. . (2010). Managing and mining graph data: a survey of clustering algorithms for graph data. , 10.1007/978-1-4419-6045-0(Chapter 9), 275-301.
[3] Wang, H. , Jia, W. , Wang, J. , Miao, Z. , & Guo, M. . (2017). Graphgan: graph representation learning with generative adversarial nets. IEEE Transactions on Knowledge and Data Engineering, PP(99).
[4] Xu, J. H. , & Hong, L. . (2010). Web user clustering analysis based on KMeans algorithm. International Conference on Information. IEEE.
[5] Huai, M. , Huang, L. , Wei, Y. , Lu, L. , & Qi, M. . (2015). Privacy-Preserving Naive Bayes Classification. International Conference on Knowledge Science. Springer International Publishing.
[6] Saunders, C. , Stitson, M. O. , Weston, J. , Holloway, R. , Bottou, L. , & Scholkopf, B. , et al. (2002). Support vector machine. Computer ence, 1(4), 1-28.
[7] Riesen, K. . (2007). Graph embedding in vector spaces by means of prototype selection. Lecture Notes in Computer Science, 4538.
[8] Fan, H. , Wu, K. , Parvin, H. , Beigi, A. , & Pho, K. H. . (2021). A hybrid recommender system using knn and clustering. International Journal of Information Technology & Decision Making.
[9] Goldberg, Y. , & Levy, O. . (2014). Word2vec explained: deriving mikolov et al.’s negative-sampling word-embedding method. arXiv.
[10] Gray Gavin.(2000).Modelling The Second Stage of Transition From Communism: A Closer Look at the Balkans. MOCT-MOST: Economic Policy in Transitional Economies(2),. doi:10.1023/A:1026712106070.
[11] Perozzi, B. , Al-Rfou, R. , & Skiena, S. . (2014). Deepwalk: online learning of social representations. ACM.
[12] Nasersharif, B. . (2015). Laplacian Eigenmaps modification using adaptive graph for pattern recognition. International Symposium on Telecommunications. IEEE.
[13] Goyal, P. , & Ferrara, E. . (2017). Graph embedding techniques, applications, and performance: a survey. Knowledge-Based Systems, 151(JUL.1), 78-94.
[14] Grover, A. , & Leskovec, J. . (2016). node2vec: Scalable Feature Learning for Networks. ACM. ACM.