Net2Vec: Weighted Learning Node Representation based on Struc2Vec

Huimin Jiang, Guozi Sun, and Huakang Li
1 Jiangsu Key Lab of Big Data and Security and Intelligent Processing
School of Computer Science, NUPT, Nanjing, 210023, China
2 Collaborative Innovation Center for Economics crime investigation and prevention technology, Nanchang, 330103, China
3 State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214125, China
huakanglee@njupt.edu.cn

***Abstract*— Network Representation Learning (NRL) shows a great importance in analyzing data. Through transformation, it turns complex network information into structured multidimensional features so that machine learning methods can be used to implement more convenient algorithm applications. Recent works have breakthrough in defining how to walk in the graph, calculating structure similarities and learning weighted relationship. However, calculating structure similarities and learning weighted relationship always apply to their own way and have difficulty to combine structure similarities and normal relationship. Fist, the contaction frequency in the relation network is defined as the weight of the edges. Then, we modified the similarity fucntion using weighted edges of Struc2Vec. In order to display the weight relationship more observably, we propose a gradient visual evaluation method of node importance representation. Experimental results shown that Net2Vec can find similar roles in the relationship network more intuitively.**

Keywords— NRL; weighted relationship; structure similarity; Struc2Vec; Net2Vec

I. Introduction

Network Representation Learning (NRL), also known as Graph Embedding Method (GEM) uses low-dimensional, dense, real-valued vectors to represent nodes in the network has risen the attention of researchers in Machine Learning field. It contains semantic relationships, facilitates computational storage, eliminates manual features and projects heterogeneous information into the same low-dimensional space for downstream calculations. Nodes tend to paly different roles in learning network because they always have their functions. By analyzing the nodes in the network, we can learn some information from it. Nodes could be classified according to their roles in the nodes. However, there is a big challenge to classify these nodes that most of nodes in the real world plays complex roles. Though in some scenario, nodes can be classified by their edges and other attributes, it cannot be extended to the further situations.

Currently, deepwalk [1] and node2vec [2] are proposed to learn the node representations and they perform well in these tasks. However, they prove to fail in classify the nodes that have similar structural identities. In our real scenario, nodes in the neighborhood tends to have intensive similarities in the structural representations, while the nodes far away tend to be different. But deepwalk and node2vec do not capture the structural similarities and rely on the attributes besides structure. Struc2vec [3] is proposed to deal with this problem. This method captures the similarities of nodes structure by evaluating their structures. Multi-layer graph is established to show their similarities. Then, by using random walk, it observes the representations and generates contexts.

Though struc2vec shows advantages in finding the structural similarities, it ignores that there exist weights between nodes in the real world. It only deals with the communities without weights such as communicating frequency. For example, in Facebook, different users contact with friends. Not only we can extract the communication relationship, but also we have their communicating frequency. User A may communicate with B for 100 times and only communicate with C for 1 time. Obviously A serves different role to B and C, but in stru2vec, they serve similar roles.

Our contributions include three parts. First, we define a new research network. In the past works, the researchers always work on the nodes and graphs. They tend to focus on the single relations between nodes and nodes in graphs, while we add other features like weight to transform graph into network. Second, we proposed a method named net2vec based on the idea of struc2vec. We modify mathematic formula of struc2vec and add the weights in to this framework. Third, we define a new standard to evaluate the performance of new proposed method. By using our method, we can classify the labels with different attributes into different rages. Our new method shows better performance in learning weighted-graph representations base on struc2vec.

Section II overviews the recent related work on learning latent representations of nodes. Section III gives detailed discussion of the net2vec. Section IV shows the experiments and compares our result to other methods. Finally, Section V gives a conclusion to the paper.

II. Related Work

They are mainly 3 common kind of network representation learning categories methods.

The first kind is Matrix decomposition. Matrix decomposition is a traditional node vectorization method. Its idea is to reduce the dimension of the adjacency matrix of the network and generate a low-dimensional representation for each node. It includes Laplacian Eigenmaps, Graph Factorization and High-Order Proximity preserved Embedding(HOPE). The goal of Laplacian Eigenmaps [5] is to map the two nodes with high similarity to a similar distance in the low dimensional space. Compared with Laplacian Eigenmaps above, the GF [6] algorithm uses the weights of the reconstructed edges of the two vectors as a measure. The two vectors may have multiple reconstruction methods, such as direct dot multiplication or cosine similarity. HOPE[7] introduces a new function to consider the high-order similarity.

The second kind is mainly based on random walk. Random walk takes advantage of the sampling of the network structure. It often has good performance in dealing with large-scale network problems, and it can also portray network local information well. In most cases, our observation of a node does not need to consider nodes that are too far away from it. Local information can be used to distinguish differences between nodes. Skip-Gram is an effective technique proposed to learn the graph embedding. Though it was mainly used in natural language learning, it proves to be successful in dealing with nodes. According to Skip-Gram's idea, the most important thing is to define the context, which is the neighborhood. Deepwalk[1] develop the idea of Skip-Gram. It introduces deep learning (unsupervised feature learning) technique in network analysis for the first time, which proved successful in natural language learning. Through imitating the random walk to learn the social representations of graph's vertices, it proves successful to deal with the sparsity of a network's representation. Using local information captures from random walk, it shows great performance in classifying nodes. Node2vec [2] modify the idea of deepwalk later. It proposes a biased second order random walk model. The main innovation is to improve the random walk strategy, by defining two parameters p and q, reaching a balance in BFS and DFS. Since it considers both local and macro information, this method has high adaptability. However, they have a limit that if the distance is larger than Skip-Gram window, the nodes will not share the same context.

The third kind of method is Deep Learning based method. Structural Deep Network Embedding (SDNE)[7] use the node similarities as input and this vector is reduced dimensionally by Auto-encoder. Another method Deep Neural Graph Representations (DNGR)[8] uses the "common path" obtained by random walks as an indicator to measure the similarity of the two nodes. The order relationship serves as an input for similarity. The big problem with this approach is that its input vector dimension is limited to |V|. On the one hand, there is a certain limit on the scale of the network, and on the other hand, the acceptance of new nodes is not good because new nodes may need to retrain the entire network after joining. In the previous models, most of the models use vector structures to encode nodes. In addition, there is a class of methods that emphasize more on encoding attributes or features of nodes themselves and their neighbors into vectors. Such methods can be collectively called Neighborhood Aggregation Algorithms, and they introduce more Multi-feature information, and share some features or parameters among neighbor nodes. GCN[0] (Graph Convolutional Networks) is one of them. Compared to the SDNE class algorithm, the GCN input vector does not have to be limited to |V|.

Struc2vec[3]is a recently proposed approach to learn the structural identities of graphs. The previous work didn't give a research in this aspect, though structural identity is strongly related to the functions or roles played by nodes in the network. Since our work is based on this framework, we will discuss this method in detail in sectionIII.

III. Scheme of Net2Vec

We modify the general framework of struc2vec, which consists of following 4 steps:

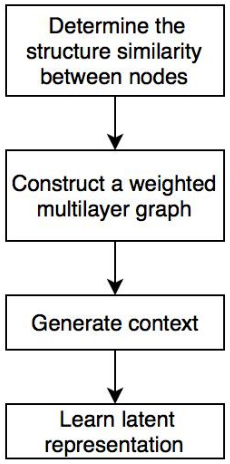
Determine the structure similarity between nodes
Construct a weighted multilayer graph
Generate context
Learn latent representation

Fig. 1. Steps of net2vec

A. Net2vec

1) structure similarity between nodes

Net2vec takes the first step to determine the structure similarities of nodes. Like the previous method, the nodes with similar degree share similar structural identity. However, in this method, if their neighborhoods share the same similarities, they are more structurally similar.

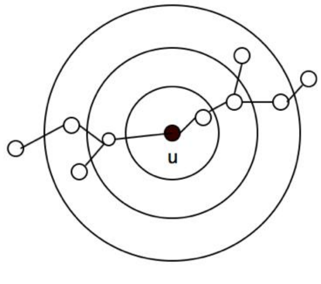

Fig. 2. Example of a neighbourhood

Consider G = (V, E, W), which means the graph consisting of vertexes and edges with weight W. $R_k(u)$ denotes the neighborhood of u at a distance of k. Let f_k (u,v,w) denote the structural distance between u and v by using the formula (1):

$$f_k(u,v,w)=f_{k-1}(u,v,w)+n(s(R_k(u),s(R_k(v))) \qquad (1)$$

As shown in figure2, we choose one node u as the center of neighborhood, k is the diameter of each layer. One circle represents one layer. f is the the number of nodes in each layer.

In the formula, n(H1, H2) denotes the distance between the ordered degree sequences H1 and H2 and f-1=0. The author of struc2vec adopt formula (2) as the distance function:

$$d(a,b) = \frac{\max(a,b)}{\min(a,b)} - 1 \qquad (2)$$

As shown in figure2, we choose one node u as the center of neighborhood, k is the diameter of each layer. One circle represents one layer. f is the the number of nodes in each layer.

2) Construct a weighted multilayer graph

The second step of net2vec is to construct the weighted multilayer graph. By using k-hop neighborhood, M is constructed as multilayer graph. As is method in step1, k is the diameter of network. For each layer k, there is a weighted undirected graph. The weights between two nodes are defined in the formula (3):

$$w_k(u,v,w)=e^{-fk(u,v,w)}*weight_k\,(u,v,w\,) \qquad (3)$$

We add the real weight to in this step. Since the struc2vec only consider the edges without their real weight. Some communications like frequency is ignored in the learning process. Therefore, we extract the second step of struc2vec and multiple the weight in the formula (3).

Besides the weight between different layers, there is also the weight between vertexes which are in different layers. The weight between different layers is define in formula (4)(5):

$$w(u_k,u_{k+1}) = log\Gamma_k(u)+\ e, k=0\ldots k^*-1 \qquad (4)$$

$$w(u_k,u_{k-1}) = 1,\ ,k=1\ldots k^* \qquad (5)$$

Γ_k (u) is the number of edges to u that have larger weight than average edge weight in layer k.

3) Generate context for nodes

net2vec uses random walk to step onto the nodes and then, generate structural context of the given nodes. The random walk steps the graph according to the possibility of formula (6):

$$p_k(u,v) = \frac{e^{-f_k(u,v)}}{z_k(u)} \qquad (6)$$

while z_k (u) is the normalization factor for vertex u, which is given by (7):

$$z_k(u) = \sum_{\substack{v\exists V \\ v\neq u}} e^{-f_k(u,v)} \qquad (7)$$

4) Learn the feature

Use skip-gram to learn the world embedding by learning meaningful representations for a variety of data.

B. Evaluation standard

In this paper, we define a new evaluation standard to evaluate the performance of classifying the nodes. As is shown in Figure 3, there are 3 kinds of nodes in this figure, and they are divided into 3 layers. We assume there is a center, and each node has a different distance from the center. We divide layers according to different distances from the center, and the number of nodes in each layer is different. However, nodes with similarities are closer together and thus gather in one level. These levels have physical meaning and reflect the connection between nodes and nodes. The red node in the figure is at the innermost layer, the yellow node is at the middle layer, and the green node is at the outermost layer. The three types of nodes have different functions and are therefore divided into different areas.

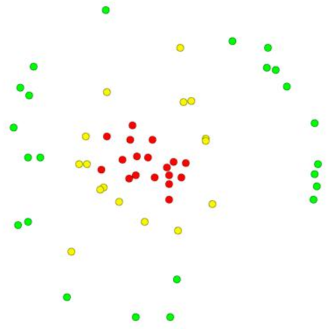

Fig. 3. A good performance example in our standard

This method is used as the evaluation criterion because many times the scatter plot does not clearly indicate which classifier works better, and when the area is clearly divided, the effect of the classifier is very obvious.

The steps of this evaluation system are:

(1) Get the result data, the data structure is: (node, node coordinates).

(2) Calculate the result data and calculate the position of the center node based on the loss function.

(3) According to the central node demarcation range, the image is output and the result is analyzed.

IV. Experiment Result

We use the Zachary's Karate Club[10] dataset to do the experiments. Zachary's Karate Club is a network of 34 nodes and 78 edges, each of which represents a club member and the edge represents whether the two members interact at the club. In this network, there are two central nodes that do not communicate with each other. There are two groups of nodes that communicate within the karate club and nodes that communicate outside the karate club.

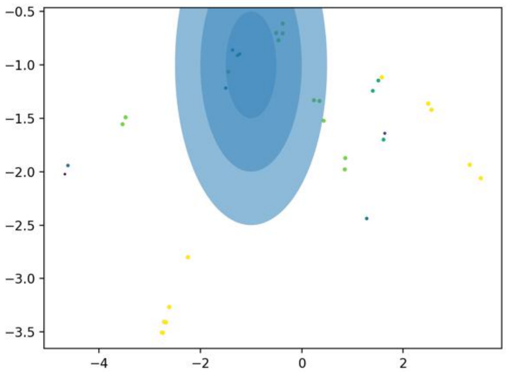

Fig. 4. Result of Zachary's Karate Club

We built a network of Karate Club datasets, and Figure 4 shows the network processed by net2vec. The figure shows that net2vec can group structurally equivalent nodes in the latent space. From the figure we can clearly see that the yellow and green nodes are the two most numerous nodes and represent the majority of the two club members. As can be seen from the figure, the two nodes are clearly separated. That is, nodes having the same color remain close in potential space, and nodes having different colors are separated into other levels in the latent space.

V. Conclusion

Network representation learning is very important in node analysis, where nodes are identified based on the network structure.

We proposed net2vec, a novel and flexible framework that defines a weighted graph to form a new research network. We added weight to the existing struc2vec. Struc2vec evaluates the structural similarity of a pair of nodes by considering the hierarchical metric defined by the ordered sequence of nodes and uses a weighted multi-layer graph to generate the context. We propose a new criterion for evaluating classification nodes to analyze experimental results.

We have proven that net2vec is good at capturing the structural identity of a node and then weighting the classification. It overcomes struc2vec's limitations by explicitly focusing on weights.

Future research work can be increased in weighting factors, including the impact of time on relationships.

Acknowledgment *(Heading 5)*

This work was supported by the NSFC (Grant No.61502247, 11501302,6150224361571238), China Postdoctoral Science Foundation (GrantNo.2016M600434, 2016M591840, 2017M621795), Natural ScienceFoundation of Jiangsu Province (Grant No. BK20150856), and Postdoctoral Science Foundation of Jiangsu Province (Grant No. 1601128B, 1501054B,1701167B), Opening Project of Collaborative Innovation Center for Economics crime investigation and prevention technology (Grant No. JXJZXTCX-015), and Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing (Grant No.2017A10).

References

[1] [1] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online

[2] Learning of Social Representations. In ACM SIGKDD.

[3] [2] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable Feature Learning for Networks. In ACM SIGKDD.

[4] [3] Leonardo Ribeiro, Pedro Saverese and Daniel Fifueuredo. 2017. Struc2vec: Learning Node Representations from Structural Identity. In ACM SIGKDD.

[5] [4] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.

[6] [5] Mikhail Belkin, Partha Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, MIT Press, 2003.

[7] [6] Amr Ahmed , Nino Shervashidze , Shravan Narayanamurthy. 2013. Distributed large-scale natural graph factorization. International Conference on World Wide Web.

[8] [7] D Wang, P Cui, W Zhu. Structural Deep Network Embedding. Acm Sigkdd International Conference on Knowledge Discovery & Data Mining,2016.

[9] [8] S Cao, W Lu, Q Xu.Deep Neural Networks for Learning Graph Representations. 2016.

[10] Francois Lorrain and Harrison C White. 1971. Structural equivalence of individ- uals in social networks. e Journal of mathematical sociology 1 (1971).

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Je rey Dean. 2013. E cient Estimation of Word Representations in Vector Space. In ICLR Workshop. [13] T Mikolov, I Sutskever, K Chen, G Corrado, and J Dean. 2013. Distributed

[12] Representations of Words and Phrases and their Compositionality. In NIPS. [14] A Narayanan, M Chandramohan, L Chen, Y Liu, and S Saminathan. 2016. sub- graph2vec: Learning Distributed Representations of Rooted Sub-graphs from

[13] Large Graphs. In Workshop on Mining and Learning with Graphs. [15] Pedram Pedarsani and Ma hias Grossglauser. 2011. On the privacy of

[14] anonymized networks. In ACM SIGKDD. [16] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. DeepWalk: Online

[15] Learning of Social Representations. In ACM SIGKDD. [17] Narciso Pizarro. 2007. Structural Identity and Equivalence of Individuals in Social

[16] Networks Beyond Duality. International Sociology 22 (2007). [18] T Rakthanmanon, B Campana, A Mueen, G Batista, B Westover, Q Zhu, J Zakaria, and E Keogh. 2013. Addressing big data time series: Mining trillions of time

[17] series subsequences under dynamic time warping. ACM TKDD (2013). [19] Lee Douglas Sailer. 1978. Structural equivalence: Meaning and definition, com-

[18] putation and application. Social Networks (1978). [20] S Salvador and P Chan. 2004. FastDTW: Toward accurate dynamic time warping

[19] in linear time and space. In Workshop on Min. Temp. and Seq. Data, ACM SIGKDD. [21] N Shervashidze, P Schweitzer, E van Leeuwen, K Mehlhorn, and K Borgwardt.

[20] 2011. Weisfeiler-Lehman Graph Kernels. JMLR (2011). [22] R Singh, J Xu, and B Berger. 2008. Global alignment of multiple protein interaction

[21] networks with application to functional orthology detection. PNAS (2008). [23] Jian Tang, Meng , Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei.

[22] 2015. LINE: Large-scale Information Network Embedding. In WWW. [24] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural Deep Network