**PROD: A New Algorithm of DeepWalk Based On Probability**

**Lijun Cai¹, Yongbao Xu¹², Tingqin He¹, Tao Meng¹ and Huimin Liu¹**

¹College of Information Science and Engineering, Hunan University, Changsha 410082, China
E-mail: ybxu@hnu.edu.cn

**Abstract.** Network representation learning is a method of expressing vertexes in a graph in the form of vectors, which facilitates further clustering and classification. Attributed graph is a kind of graph. In the attributed graph, each vertex has some text attribute. Through the network structure of these text attributes and attributed graph itself, it can be well clustered and classified. With the development of deep learning, the network representation learning field combined with deep learning has also produced a series of excellent algorithms, such as DeepWalk and Node2vec, etc. However, the randomly generated sequence will lose the text attribute of the vertex. This paper proposes a new algorithm PROD which enable the deepwalk algorithm to combine the probability to generate the corresponding sequence. Later experiments show that in the case of a large number of edges, this algorithm generates low-dimensional representations of vertices that are better for clustering with the kmeans algorithm than for deepwalk.

1. Introduction

With the rise of machine learning technology, how to better represent the points in the network and apply it to machine learning algorithms has become a new research task. Network representation learning [1] is a method of expressing nodes in the network as low-dimensional vectors, in order to better apply to community detection [7], link prediction [8], classification [10] and clustering and other fields. The development of deep learning [6] in recent years has also promoted the development of this field.

Typical network representation learning algorithms based on machine learning in recent years are DeepWalk [3] and Node2vec [4] algorithms. Deepwalk algorithm applies the technology of deep learning to the network representation learning field for the first time.

2. Random Walks Base on Probability

2.1. The Limitations of Deepwalk

Deepwalk considers the connections of nodes and their neighbors by means of random walks, and then considers the local connections in a network. However, if we only consider the connection between two vertexes, this text attribute [2] is ignored for nodes.

![Figure 1. The example of citation data set.](image-url)
Among them, the vertex A is connected with the other three vertexes, indicating that the A vertex has a reference relationship with the other three vertexes. We represent the text attributes of the four points with a vector of length 5. Assumptions: A=(1,1,1,1,1), B=(1,0,0,0,0), C=(1,1,1,0,0), D=(1,1,1,1,1).

Then, D and A should be the most similar, and the distance between the two points should be the nearest. For the above figure, the probability ratio of point A to point B, C, and D should be 1:3:5. However, if you use deepwalk directly, you will get a probability ratio of 1:1:1. Based on this, this paper proposes a new algorithm PROD.

2.2. Probability-Based Random Walk Algorithm

2.2.1. Data Processing. For the citation data set used in this paper, the connection between the two points indicates that there is a reference between the two articles, and then it can be considered that there is a high similarity between the two articles. There is another attribute that is the words appearing in the two articles. If there are many identical words in two articles, it can be regarded as similar between the two articles. However, the citation dataset has many blocks, which is not conducive to the generate a random sequences of the deepwalk algorithm. Therefore, we make the following adjustments:

We use text attributes. First, we set a threshold $\alpha$ and then compare the text attributes of all nodes. If the number of similar words in these two points exceeds this threshold, we think that these two points have a higher degree of similarity. We connect these two vertexes, and the number of edges of the entire graph will increase. In subsequent operations, operate on this expanded graph.

We assign a weight to all sides, representing the degree of similarity between the two points. This weight is defined as how many words are repeated between two points.

If we only expand this graph according to the coincidence rate of text attributes, the resulting new graph will lose its original network structure. In order to maintain the original network structure, we increase the weights of the two edges originally connected to each other according to the proportion before the connection.

We describe this algorithm:

- $e(v_i, v_j)$: the edge connecting vertex $v_i$ and vertex $v_j$. If $e = 0$, point $v_i$ and $v_j$ are not connected at two points before expansion. If $e = 1$, points $v_i$ and $v_j$ are connected at two points before expansion.

- $\text{Word}_\text{similar}(v_i, v_j)$: the degree of similarity between two points (calculated by counting how much of the textual attributes of the two points are the same).

- $E(v_i)$: indicates the set of neighbor nodes connected to $v_i$.

- $W(v_i)$: Represents the sum of the weights of the edges formed by all the points connected to $v_i$.

- $W'(v_i)$: represents the sum of the weights of $v_i$ points and all points connected to it in the original graph.

\[
W(v_i) = \sum_{v_j \in E(v_i)} \text{word}_\text{similar}(v_i, v_j)
\] (1)

\[
W(v_i)' = \sum_{v_j \in E(v_i)} e(v_i, v_j) \times \text{word}_\text{similar}(v_i, v_j)
\] (2)

We compute new weight value for edge which existence in original graph:

\[
\text{word}_\text{similar}(v_i, v_j) =
\sum_{v_j \in E(v_i)} \left[ \frac{e(v_i, v_j) \times W(v_i) \times \text{word}_\text{similar}(v_i, v_j)}{W'(v_i)} + (1 - e(v_i, v_j)) \times \text{word}_\text{similar}(v_i, v_j) \right]
\] (3)

Correspondingly, we update $W(v_i)$:

\[
W(v_i) = \sum_{v_j \in E(v_i)} \text{word}_\text{similar}(v_i, v_j)
\] (4)

2.2.2. Get Probability. According to the method of 2.2.1, we can get the degree of similarity of each adjacent point pair, we think that
word \_\similar(v_i, v_j) = word \_\similar(v_j, v_i) \tag{5}

We define the weight distribution of connections for each point as:

\[ P(v_i|v_j) = \frac{\text{word\_simil}ar(v_i, v_j)}{W(v_i)} \tag{6} \]

This weight distribution is the deepwalk random walks probability distribution which we control. We fuse this probability into the random walk process to achieve the PROD algorithm. We defined: Q(v_i) is a list, each element in the list consists of two parts, the first part is the neighbor of the v_i node, the second part is the weight of the edge of the neighbor and v_i.

Select V_i neighbor point by probability.

The algorithm is as follows:

```
Algorithm 1 get_point_base_probabiliy(Q(v_i))
Input: Q(v_i)
Output: point

0: Initialization
sums_weight = 0, position = 1
1: for each vi_value in Q(v_i).value do:
2:     sums_weight = sums_weight + vi_value
3: end for
4: get a point x by random
5: for each (v_j, value) in Q(v_i) do:
6:     if x in range from position to position+value do:
7:         break
8:     else:
9:         position = position + value
10: end if
11: end for
12: Return v_j
```

3. Experiment and Analyse

3.1. Experimental Data
This paper selects two data sets, named Cora data set and Citeseer data set, and using deepwalk algorithm and probability-based deepwalk algorithm to learn the low-dimensional representation of points with the default parameters, and then uses the kmeans [5] algorithm for clustering. And determine the clustering effect for comparison.

Cora dataset: The dataset contains 2708 articles. Each article is represented by a 0-1 vector with a length of 1433 dimensions. Each dimension represents a word. All articles are divided into 7 categories.

Citeseer dataset: The dataset contains 3,312 articles. Each article is represented by a 0-1 vector with a length of 3703 dimensions. Each dimension represents a word, and all articles are classified into 6 categories.

3.2. Evaluation Criterion
This article uses three criteria:

Homogeneity [9]: A clustering result satisfies homogeneity if all of its clusters contain only data points which are members of a single class.

Completeness [9]: A clustering result satisfies completeness if all the data points that are members of a given class are elements of the same cluster.

V\text{-measure} [9]: The V-measure is the harmonic mean between homogeneity and completeness. The formula is as follows
\[ v = \frac{2 \cdot (\text{homogeneity} \cdot \text{completeness})}{(\text{homogeneity} + \text{completeness})} \]  

(7)

3.3. Experiment and Analysis

In this paper, setting the threshold parameter \( \alpha \) means that when the same word in two articles exceeds \( \alpha \), we consider that there is a connection between the two points, the whole graph will become smaller with \( \alpha \), and the number of edges will be more and more. It can be used to simulate the performance of two algorithms when the number of edges is different.

In this experiment, the values of \( \alpha \) take 9, 8, 7, 6, 5, 4, 3, 2 from left to right.

**Table 1.** Cora dataset threshold and the number of edges of the table.

| \( \alpha \) | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    |
|------------|------|------|------|------|------|------|------|------|
| lines      | 10980| 11724| 14678| 26606| 74038| 245932| 772616| 2070892|

**Figure 2.** Figure with score of Cora dataset.

**Table 2.** Citeseer dataset threshold and the number of edges of the table.

| \( \alpha \) | 9    | 8    | 7    | 6    | 5    | 4    | 3    | 2    |
|------------|------|------|------|------|------|------|------|------|
| lines      | 18284| 28472| 55918| 131158| 333400| 842766| 2015708| 4337452|

**Figure 3.** Figure with score of Citeseer dataset.

Based on the above experimental results, we discover that when the threshold is relatively small, there is few difference between the two, because when the number of edges is small, the neighboring nodes are also small. As the threshold decreases, the number of connected edges of the entire graph also gradually increases. The deepwalk algorithm shows that the three criteria generated with the increase of the number of connected edges increase first and then decrease. For the PROD algorithm, it shows the trend of increasing steadily. In summary, PROD algorithm can get better result, in the case of a large number of edges.
4. Conclusion
This paper proposes a new algorithm PROD, which effectively enables the network representation method to be applied to vertex representations of graphs with large number of text attributes, and avoid the problem of discarding vertex text attributes due to random walks. Therefore, for the vertex representation of a graph with a large number of text attributes, the results presented in this paper are better than using the deepwalk method directly.

5. References
[1] Yang C, Zhao D, Zhao D, et al. 2015. Network representation learning with rich text information [C]// International Conference on Artificial Intelligence. AAAI Press, 2015: 2111-2117.
[2] Hartmut Ehrig U P, Taentzer G. 2004. Fundamental Theory for Typed Attributed Graph Transformation. [J]. Lecture Notes in Computer Science, 2004, 3256: 161-177.
[3] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: online learning of social representations [C]/. 2014. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2014:701-710.
[4] Grover A, Leskovec J. node2vec: Scalable Feature Learning for Networks [J]. 2016, Kdd. 2016:855-864.
[5] JinHuaXu, HongLiu. Web User Clustering Analysis based on KMeans Algorithm [C]// 2010 international conference on information, networking and automation. 2010:V2-6-V2-9.
[6] Lecun Y, Bengio Y, Hinton G. 2015. Deep learning[J]. Nature, 2015, 521(7553):436.
[7] Fortunato, Santo. 2009. Community detection in graphs [J]. Physics Reports, 2009, 486 (3):75-174.
[8] Liben-Nowelly D, Kleinberg J. 2007 the Link Prediction Problem for Social Networks [J]. Journal of the American Society for Information Science & Technology, 2007, 58 (7):1019 –1031.
[9] Rosenberg, A., & Hirschberg, J. (2008). V-Measure: A Conditional Entropy-Based External Cluster Evaluation Measure. EMNLP-CoNLL 2007, Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, June 28-30, 2007, Prague, Czech Republic (pp.410 - 420). DBLP.
[10] Zhu S, Yu K, Chi Y, et al. 2007. Combining content and link for classification using matrix factorization[C]// International ACM SIGIR Conference on Research and Development in Information Retrieval. ACM, 2007:487- 494.