Information Network Embedding with Hybrid Approach: Nove2vec, Doc2vec, and TF-IDF

Yunseob Shin
IIS Lab., Sungkyunkwan University.

1. Introduction

Network embedding is emerging topic in data mining and machine learning. Mapping a unique vector representation of each vertex in a graph-structured network in a low-dimensional space makes the dataset amenable to more machine learning toolsets and mathematic techniques generally used for other tasks. There are interesting existing frameworks and methods for network embedding with various type of approaches such as random walk, deep learning and matrix factorization. Those methods mainly focus on preserving the high-order proximity between vertices in the network.

In this project, I propose a hybrid approach for embedding information network which not only utilizes the linkage structure of network but also its contents information and relative importance between vertices. I implement this approach by tuning node2vec model, a random walk-based network learning framework, added with doc2vec, the embedded representation of each web document, and the tf-idf weights. The evaluation of embedding results was evaluated by node classification and link prediction, comparing with the original embedding method. In the experiments, although it did not show better performance in link prediction task, the proposed hybrid approach outperformed existing two methods in node classification task.

2. Objective

Network embedding is one of the promising subjects in network analysis field since it can make graph-structured datasets applicable to much more mathematic techniques and machine learning models. The goal of network embedding is

1) to reconstruct original networks, and
2) support network inference.

Several interesting network embedding model have been suggested for the two goals such as node2vec, SDNE, GCN, etc. Node2vec is one of the powerful network embedding methods which is based on the skip-gram model as its name's similarity with the word2vec. Node2vec takes the random walks of vertices from a network as the sentences of words from a corpus so that just putting them into skip-gram model learns the vector representation of vertices. In this project, I suggest a hybrid approach for information network embedding mixing the existing embedding model (node2vec) with contents information and link weights between documents.

3. Proposed Model

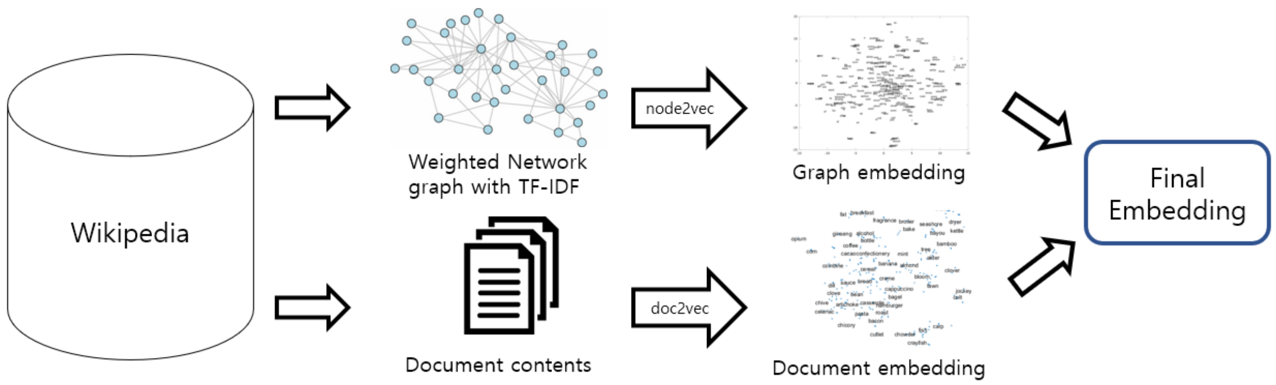

Figure 1. The overall procedure for the proposed method using Wikipedia dataset

In this project, I propose a mixed model for information network embedding combining node2vec, doc2vec, and TF-IDF. Each representation of an information network implies certain characteristics of the network. The vector embedding by node2vec preserves the high-order proximity, which can be simply thought as the relations of documents. Doc2vec learns an embedding of a document by its contents. TF-IDF can used to measure the weight of the relationship between documents.

3-1. Node-level TF-IDF edge weighting

In this project, I add a TF-IDF term in node level by giving edge weight to the original graph data as TF-IDF value between pairs of documents by following strategy

$$w(u,v) = \frac{1}{Z} * \left(\frac{1}{2} + \frac{TF(u,v)}{\log(IDF(v)+1)}\right) \text{ where } \frac{1}{Z} \text{ is for normalization}$$

The directed weight $w(u,v)$ is used for random walk in addition to node2vec's BFS-DFS interpolating random walk strategy. $TF(u,v)$ means the number of references of a document v from the document u, and $IDF(v)$ means the number of references of v from the entire documents.

3-2. Hybrid embedding with node2vec and doc2vec

The mixing strategy is to ensemble the results of each vector embedding with trainable parameters. The final embedding can be calculated as below.

$$e(v) = \alpha * ne(v) + (1-\alpha) * de(v), 0 \leq \alpha \leq 1$$

$ne(v)$ is the vector embedding of document v learned from node2vec, and $de(v)$ is that learned from doc2vec. The hyperparameter α controls the balance between two results.

4. Evaluation

In this project, I used Wikipedia data as an information network, where each node is an article, and the hyperlink between two nodes indicates their relationship. The evaluation of network embedding is to measure how well we can infer the original network. In this project, I evaluated the proposed approach in two mostly used inferring tasks: node classification and link prediction. The baseline embeddings to compare were node2vec and doc2vec, which are the cases of the proposed model where the α is 1 and 0.

4-1. Node Classification

Node classification is a downstream task that infers attributes of nodes in a network by its given features. The evaluation of node classification performance was measured by test accuracy resulted from a Support Vector Machine classifier with RBF kernel (RBF-SVM).

Node Classification (# of nodes: 12,305, # of labels: 73)											
alpha	0.0 (n2v)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0 (d2v)
Unweighted	0.4970	0.5481	0.5810	0.6086	0.6248	0.6326	**0.6370**	0.6289	0.6200	0.5863	0.5359
Weighted (TF-IDF)	0.4953	0.5469	0.5757	0.6013	0.6236	0.6338	0.6228	0.6188	0.6131	0.5867	0.5396

Table 1. Node classification performance

The node classification results are shown in Table 1. I sampled 12,305 nodes with 73 class. I set 80% of nodes as training set and 20% where the ratio of two datasets are equal for every class. The results showed that the more similar the two embeddings from node2vec and doc2vec are mixed, the RBF-SVM classifier performs better.

In my interpretation, the node classification performance if better in doc2vec than node2vec since doc2vec embedding concentrates the internal contents whereas node2vec only considers the linkage structure of network. Based on this situation mixing two feature vectors successfully lead to better performance by remapping feature vectors closer to others closely connected with in the original network but still considering the internal contents. TF-IDF weighting, however, was not effective on the performance.

4-2. Link Prediction

Link prediction is a traditional task in graph-structured network data which predicts the future or hidden links between entities in a network. The evaluation of link prediction performance was measured by “precision at degree”, where the link prediction model infers the neighbors of each node as many as the number of its neighbors in the original graph.

Link Prediction (# of node: 26592)											
alpha	0.0 (n2v)	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0 (d2v)
Unweighted	**0.6193**	0.6179	0.5947	0.5376	0.4446	0.3315	0.2355	0.1740	0.1430	0.1309	0.1272
Weighted (TF-IDF)	0.6148	0.6153	0.6026	0.5599	0.4704	0.3506	0.2447	0.1785	0.1457	0.1318	0.1272

Table 2. Link prediction performance

The link prediction results are shown in Table 1. I sampled 26,592 nodes with 194,685 edges. I set 80% of nodes as training set and 20% where the ratio of two datasets are equal for every class. The results showed that since doc2vec is weak for link prediction because it does not consider any linkage information, the mixing approach just lead to worse performance of node2vec. In addition, TF-IDF could affect when two embeddings are mixed but did not make changes for entire performance.

5. Conclusion

In this project, I propose a hybrid approach for embedding information network such as Wikipedia. I applied an edge-weighting scheme based on the idea of TF-IDF in node level, and I mixed features learned by two existing feature embedding methods: node2vec and doc2vec. In node classification task, the classifier with my mixed features performed better than the two features with the same training and test datasets whereas it did not perform well in link prediction task. In further study, I will find a novel embedding method which converge two methods in lower level i.e., a novel loss function a novel learning structure.

6. References

[1] Goyal, P., & Ferrara, E. (2018). Graph embedding techniques, applications, and performance: A survey. Knowledge-Based Systems, 151, 78-94.
[2] 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.
[3] Le, Q., & Mikolov, T. (2014, January). Distributed representations of sentences and documents. In International Conference on Machine Learning (pp. 1188-1196).
[4] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems (pp. 3111-3119).
[5] Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, pp. 133-142).