Incorporate Group Information to Enhance Network Embedding

Jifan Chen, Qi Zhang, Xuanjing Huang
Shanghai Key Laboratory of Data Science
School of Computer Science, Fudan University
825 Zhangheng Road, Shanghai, P.R.China
{jfchen14, qz, xjhuang}@fudan.edu.cn

ABSTRACT

The problem of representing large-scale networks with low-dimensional vectors has received considerable attention in recent years. Except the networks that include only vertices and edges, a variety of networks contain information about groups or communities. For example, on Facebook, in addition to users and the follower-followee relations between them, users can also create and join groups. However, previous studies have rarely utilized this valuable information to generate embeddings of vertices. In this paper, we investigate a novel method for learning the network embeddings with valuable group information for large-scale networks. The proposed methods take both the inner structures of the groups and the information across groups into consideration. Experimental results demonstrate that the embeddings generated by the proposed methods significantly outperform state-of-the-art network embedding methods on two different scale real-world networks.

Keywords

Network Embedding; Group Information; Large Scale

1. INTRODUCTION

Many types of relations and processes in physical, social, and information systems can be naturally modelled by networks, such as communication, citation, and social information. However, in real-world applications, the size of many networks is extremely large. For example, Twitter has 316 million monthly active users [1] in 2015. Methods that process these networks directly through vertices and edges may encounter efficiency issues when solving practical real-world problems. Hence, network embedding, which is used to represent each vertex of a network with a low-dimensional vector that can preserve the similarities between them, has attracted continuous attention and has been successfully used in various applications, including image processing, knowledge graph, recommendation, etc.

Along with the increasing requirements, a variety of researchers have studied the network embedding construction problem from different aspects [10, 3, 7, 9, 8]. Classical methods (e.g., IsoMap [10] and Laplacian eigenmaps [1]) usually transform this task into a constrained optimization problem. Hence, the usefulness of these methods may be heavily impacted by the computation consumption of processing hundreds of millions of nodes. To process large-scale networks, DeepWalk [7] uses a shallow neural network architecture. LINE [9] uses both first-order and second-order proximity to train the embedding, and negative sampling methods to reduce the computational requirement. However, most of the previous studies focused on a classical network and took only vertices and edges into consideration.

In practical tasks, in addition to vertices and edges, many networks contain groups or communities. For example, in social media (e.g., Youtube and Facebook), users can create groups that other users can join. Previous literatures [11] also show that this kind of network is common in real-world social, collaboration, information, and many other kinds of networks. More than 200 different kinds of large real-world networks where nodes explicitly state their group memberships were studied in the work done by Yang and Leskovec [11]. Although communities or groups in networks can provide valuable information, previous network embedding studies rarely took this information into consideration.

In this paper, we study the problem of incorporating group information for network embedding generation. We think that the generated embedding of vertices should be placed closely in low dimensional space if they share similar neighbours or the group they joined are similar. In order to meet the requirements, we then propose a novel method to achieve the task. First, a random walk cross groups is adopted to gain the information between vertices. Then the inner structures of groups are obtained by random sample vertices in that group. Finally, we propose to use a group vector to preserve the information between vertices as well as the information of groups.

2. THE PROPOSED METHODS

Inspired by the work of DeepWalk [7] and the idea of modelling document [4, 2] in natural language processing, our model contains two main stages, sampling and training. We will then illustrate the two steps in details.

In the sampling stage, we uniformly take a random vertex v_i as the root of a random walk W_{v_i}, then from the root we sample the neighbors of the last vertex visited until we reach the maximum length L. After this step, we can get $W_{v_i} = v_i, v_{i+1}, v_{i+2}, ..., v_L$, where v_{i+n} is one of the neighbours of v_{i+n-1}. In our experiment, we set the length L to be fixed, but there is no restriction for these walks to be the same length. Then, for each generated random walk

[1] https://about.twitter.com/company

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA
© 2016 ACM. ISBN 978-1-4503-4073-1/16/10. . . $15.00
DOI: http://dx.doi.org/10.1145/2983323.2983869

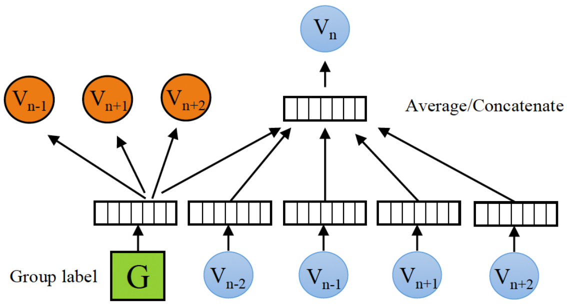

Figure 1: The architecture of the proposed model, the blue circles represent the vertices sampled by random walk and labeled by group G, the orange circles represent the vertices random sampled in group G, and the green square represents the group label.

W_{v_i}, we randomly sample a group label g_j from the groups v_i belongs to and assign it to W_{v_i}, those labeled walks are then used for training.

The framework of our training model is shown in Figure 1, as we can see, every group label and vertex is mapped to a unique vector and all the vertex embeddings are shared among different groups. The group vector and the vertex vectors are averaged or concatenated to predict the center vertex in a sliding window over a random walk. We also use the group vector to predict the vertices randomly sampled in that group. More formally, the objective of our proposed model is defined to maximize the following log probability:

$$\begin{aligned}\mathscr{L} = \sum_{g_i \in C} (\alpha \sum_{W \in W_{g_i}} \sum_{v_j \in W} \log p(v_j|v_{j-k},...,v_{j+k},g_i) + \\ \beta \sum_{\hat{v_j} \in \hat{W_{g_i}}} \log p(\hat{v_j}|g_i)),\end{aligned} \quad (1)$$

where C is the set of different groups and g_i is the label of the ith group, W_{g_i} contains random walks W labeled with g_i, $\hat{W_{g_i}}$ contains vertices randomly sampled from group g_i, α and β are the weights that specifies a trade-off between the information cross groups and the information in that group, the probability $\log p(v_j|v_{j-k},...,v_{j+k},g_i)$ and $\log p(\hat{v_j}|g_i)$ are respectively defined as:

$$\log p(v_j|v_{j-k},...,v_{j+k},g_i) = \frac{\exp(\bar{u}^T u'_j)}{\sum_{n=1}^{M} \exp(\bar{u}^T u'_n)}, \quad (2)$$

$$\log p(\hat{v_j}|g_i) = \frac{\exp(u_{g_i}^T \hat{u_j})}{\sum_{n=1}^{M} \exp({u_{g_i}}^T \hat{u_n})}, \quad (3)$$

where M is the number of vertices in the network, u' and u are respectively the output and input vector representation of v, and $\bar{u}$ is the averaged vector representation of the context and label g_i, defined as follows,

$$\bar{u} = \frac{1}{2k}(u_{g_i} + \sum_{-k \le p \le k, p \ne 0} u_{j+p}), \quad (4)$$

where k is the size of sliding window and u is the input representation of vertex v.

The computation of gradient $\nabla \log p(v_j|v_{j-k},...,v_{j+k},g_i)$, $\nabla \log p(\hat{v_j}|g_i)$, is proportion to the number of vertices in the network G, and in real-world networks, there can be hundreds of millions of vertices, so it is impractical to compute these gradients directly. To address this problem, we adopt the approach of negative sampling proposed in [5] to reduce the computational requirement.

We then give a brief explanation to our proposed model. Since we adopt random walks starting from different vertices in the network, and there should be similar walks for the vertices sharing similar neighbours, thus, those vertices will be placed closely. Also, as we use the group vector to predict the vertices randomly sampled in that group, vertices in that group will be placed closely. Moreover, because of the special form of assigning group labels to random walks, vertices from different groups may be assigned with the same group label, the group vector acts as a memory cell that contains both the information of vertieces in that group and across groups, resulting in a closer distance between some vertices in same groups. We name our proposed model **GENE**, which means group enhanced network embedding.

3. EXPERIMENT

In this section, we report the experimental results of our methods on two large scale real-world networks. We use two different group recommendation tasks to evaluate the quality of the generated embeddings.

3.1 Datasets

We use two large real world datasets for evaluating the proposed methods[2].

- **Amazon:** This network is provided by Yang [11] and collected by crawling Amazon website. In this network, if a product i is frequently co-purchased with product j, the graph contains an undirected edge from i to j. Each product category provided by Amazon defines each ground-truth community.

- **Youtube:** This network is provided by Alan Mislove [6]. Youtube is a video-sharing web site that includes a social network. In the Youtube social network, users form friendship each other and users can create groups which other users can join. Such user-defined groups are considered as ground-truth communities.

3.2 Baseline Methods

We compare our proposed methods with some of the state-of-art existing network embedding methods to validate the performance.

- **DeepWalk [7]:** It uses information obtained from truncated random walks to learn latent representations by treating walks as the equivalent of sentences.

- **LINE [9]:** It is a network embedding method, which tries to preserve both the local and global network structures.

- **GroupWalk:** It only takes the labeled random walks cross groups mentioned above into consideration. The group vector and the vertex vectors are averaged or concatenated to predict the centre vertex in a sliding window over a random walk.

- **GroupOnly:** It only uses the group vector to predict the vertices sampled in a specific group.

[2] They are publicly available. http://snap.stanford.edu/data/

Table 1: Results of recommendation Task1 on Amazon

	percentage	10%	20%	30%	40%	50%	60%	70%	80%	90%
MAP@5	DeepWalk	52.52%	58.93%	65.80%	69.60%	73.65%	74.64%	76.41%	77.61%	79.01%
	LINE	46.22%	53.26%	60.02%	64.22%	68.38%	70.22%	72.08%	73.12%	74.79%
	GroupWalk	58.85%	66.67%	73.09%	77.26%	81.25%	82.65%	84.17%	85.22%	**87.25%**
	GroupOnly	56.90%	61.03%	63.23%	68.94%	72.98%	74.59%	74.95%	74.26%	70.86%
	GENE	**63.64%**	**70.65%**	**76.25%**	**80.50%**	**84.13%**	**84.79%**	**85.63%**	**85.41%**	84.41%

Table 2: Results of recommendation Task2 on Amazon

	Percentage	5%	10%	15%	20%	25%	30%
MAP@5	DeepWalk	33.94%	51.57%	55.29%	58.53%	62.00%	64.85%
	LINE	30.27%	46.38%	49.88%	53.21%	56.70%	59.64%
	GroupWalk	37.67%	61.61%	63.05%	66.60%	70.07%	72.88%
	GroupOnly	34.86%	58.15%	61.75%	64.85%	64.88%	72.80%
	GENE	**42.48%**	**62.32%**	**67.01%**	**70.07%**	**72.91%**	**75.56%**

3.3 Experiment Protocols

We evaluate the proposed methods on two tasks of recommendation, the ground-truth communities in the datasets are treated as the groups mentioned above.

- **Task 1:** We randomly sample 10% nodes in the network and then remove their communities by the ratio increasing from 10% to 90%. The randomly removed communities are used as the ground-truth for evaluation. The performance of recommendation can be evaluated with them. It is a simulation for the following situation: There are several new nodes joined the network, they have formed the relationship with other nodes but with few communities, then we have to recommend new communities for them.

- **Task 2:** For all of the nodes in the network, we remove their communities by the ratio increasing from 5% to 30%. The randomly removed communities are used as the ground-truth for evaluation. It is a simulation for such situation: The whole network has been well formed and is relatively stable, then we have to recommend new communities for the nodes in the network to join.

The recommend protocol is described as follows: First, for each node n, we find its most similar points set S by computing the cosine similarity between its embedding and other nodes' embedding. Second, for each node r in the set S, find all the communities r belongs to, if there are any group that dose not belong to n, we recommend this group to n.

3.3.1 Parameters Setting

In the random walk phase, we iterate over the network for 10 times. At each iteration, we sample one random walk per node and we set the walk length to be fixed to 15 for both of the networks. For the GENE model, we simply set $\alpha = \beta = 1$. For the negative sampling part in the proposed method, the number of negative samples is set to 5. The number of vertices randomly sampled from groups is also set to 5.

3.4 Results and Analysis on Amazon

The results on Task 1 are shown in Table 1, it is obvious that the MAP@5 of all the methods improve as we increase the ratio of removing communities. It is because when the ratio increases, the number of ground-truth answers also increases, causing a higher precision of top ones.

Another observation from the table is that by adding group vertices, GroupWalk achieves consistently better performance than DeepWalk, and GroupOnly also achieves a fairly good performance. GENE outperforms all of the other methods and achieves the best performance. It not only proves the group information is useful but also shows the way we proposed to incorporate those inner-group and cross-group information is actually useful.

From Table 1, we can also find that the performance of the GroupOnly model improves at first as we remove more communities. But when we remove more than 70% communities, its performance begins to decrease. This is mainly because when we remove a large amount of community relationships from some vertices, then those vertices can only appear in a small set of communities, causing a lower probability to be sampled in sampling stage and thus they can be trained with less communities. So there is no surprise for the decreased performance, and this also explains why there is a drop of performance of the GENE model when we remove the communities by 90%.

Then we compare the GroupWalk and the GroupOnly model, the GroupWalk performs much better than the GroupOnly consistently. The result states that there are more groups recommended by finding nodes sharing the same neighbors than the nodes in the same community on Amazon network.

The GENE model achieves the best performance except the 90% remove of communities, it proves our assumption that both inner-group and cross-group information are helpful for network embedding construction, and they act like a supplement to each other. GENE model has taken both of the information into consideration, so it performs better than the GroupWalk and GroupOnly model alone. The performance of the GENE decreases as the GroupOnly model, but in a much small range, showing this model is also stable.

The results for Task 2 are shown in Table 2, the results are almost in the same patterns as what shows in Table 1. The GENE model outperforms all of the other methods, and in general, GroupWalk and GroupOnly also achieve better results than DeepWalk and LINE. It proves that our proposed method can do better in recommendation for the stable vertices in a network as well as the vertices which join the network recently.

Table 3: Results of recommendation Task1 on Youtube

	percentage	10%	20%	30%	40%	50%	60%	70%	80%	90%
MAP@5	DeepWalk	11.71%	14.39%	21.32%	27.96%	32.48%	34.61%	36.75%	39.10%	40.25%
	LINE	7.34%	7.88%	14.52%	17.56%	20.92%	22.46%	23.56%	25.72%	29.31%
	GroupWalk	13.79%	19.15%	22.68%	31.55%	33.54%	38.46%	40.93%	44.21%	**49.71%**
	GroupOnly	16.50%	21.56%	29.39%	33.48%	37.37%	42.65%	38.30%	37.43%	36.27%
	GENE	**23.31%**	**24.92%**	**32.00%**	**39.52%**	**42.21%**	**45.31%**	**49.99%**	**51.93%**	48.23%

Table 4: Results of recommendation Task2 on Youtube

	percentage	5%	10%	15%	20%	25%	30%
MAP@5	DeepWalk	7.74%	11.51%	13.44%	15.78%	18.79%	21.75%
	LINE	4.02%	5.96%	8.41%	9.35%	10.88%	12.91%
	GroupWalk	9.57%	12.67%	15.09%	18.69%	21.53%	24.87%
	GroupOnly	12.88%	16.77%	18.65%	21.88%	25.57%	27.85%
	GENE	**13.56%**	**18.62%**	**20.92%**	**24.51%**	**28.20%**	**32.08%**

3.5 Results and Analysis on Youtube

The Youtube network is quite different from Amazon especially for its sparse group membership. Facing such situation, let's look at our results in Table 3 and Table 4.

The results in Table 3 are much worse than the results of Amazon network shown in Table 1, but it is reasonable since the group membership per node decreases a lot, it is harder for recommendation. Once again, GENE achieves the best performance, and incorporating group information significantly improves the performance.

Take a comparison between the GroupWalk and GroupOnly model, things go to the opposite side of what shows in Amazon network. The GroupOnly model generally performs better than GroupWalk model, except for the cases when we remove more than 70% of communities. Such results show that there are more groups recommended by finding nodes in the same community than the nodes sharing the same neighbours on Youtube network. From this result, we can also see the difference between the two networks. The best performance of GENE model once again proves it can preserve both the inner-group and cross-group information. The results shown in Table 4 are consistent with the results in Table 3.

Overall, from these results, we conclude that with the test under two different networks, our network embedding trained with group information is more effective on the task of recommendation than the previous methods for training network embedding.

4. CONCLUSIONS

In this work, we study the problem of enhancing network embeddings using group information for the networks which explicitly contain groups or communities. We assume that a good network embedding with group information should take the inner-group information as well as the cross-group information into consideration. Hence, we propose a model which incorporates both the inner structure within a group and the information cross groups to train the network embedding. Experimental results demonstrate that the embeddings generated by the proposed method outperforms state-of-the-art network embedding methods on two different scale real-world networks.

5. ACKNOWLEDGEMENT

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61532011, 61473092, and 61472088), the National High Technology Research and Development Program of China (No. 2015AA015408).

6. REFERENCES

[1] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, volume 14, pages 585–591, 2001.

[2] N. Djuric, H. Wu, V. Radosavljevic, M. Grbovic, and N. Bhamidipati. Hierarchical neural language models for joint representation of streaming documents and their content. In *Proceedings of the 24th International Conference on World Wide Web*, pages 248–255. International World Wide Web Conferences Steering Committee, 2015.

[3] Y. Jacob, L. Denoyer, and P. Gallinari. Learning latent representations of nodes for classifying in heterogeneous social networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 373–382. ACM, 2014.

[4] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of The 31st International Conference on Machine Learning*, pages 1188–1196, 2014.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[6] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 29–42. ACM, 2007.

[7] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM, 2014.

[8] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174. ACM, 2015.

[9] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee, 2015.

[10] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[11] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *2012 IEEE 12th International Conference on Data Mining*, pages 745–754. IEEE, 2012.