SANE: Scalable Attribute-aware Network Embedding

Weiyi Liu, Zhining Liu*
University of Electronic Science and Technology of China, China
IBM T.J.Watson Research Center, U.S.A
weiyiliu@us.ibm.com,zhining.liu@ibm.com

Toyotaro Suzumura
IBM T.J.Watson Research Center
NY, 10598, USA
suzumura@acm.org

Guangmin Hu
University of Electronic Science and Technology of China
Chengdu, Sichuan, China
hgm@uestc.edu.cn

ABSTRACT

Network in the real world generally contains topology information and attribute information together. How to leverage these two information simultaneously has gradually become one of the research focuses on network embedding. In this position paper, we propose SANE: scalable attribute-aware network embedding to learn the joint embedding representations efficiently. The present preliminary results show that, by enforcing the alignment of the locally linear relationship between each node and its K-nearest neighbors on topology and attribute space, the embedding representations learned by SANE are more informative comparing with the one generated from topology or attribute information alone. In addition, comparing matrix factorization based network embedding methods with quadratic time complexity, SANE can easily support scalable learning for its quasilinear time complexity.

CCS CONCEPTS

• **Computing methodologies → Learning latent representations**; *Deep belief networks*;

KEYWORDS

representation learning; attributed graph; graph embedding; scalability

ACM Reference Format:
Weiyi Liu, Zhining Liu, Toyotaro Suzumura, and Guangmin Hu. 2018. SANE: Scalable Attribute-aware Network Embedding. In *Proceedings of KDD Mining and Learning with Graphs Workshop (KDD MLG'18).* ACM, London, United Kingdom, Article 4, 4 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Network embedding (NE) intends to learn latent, dense and low dimensional representations of nodes on the network, through topology information, such as node-pairs. Due to the broad applicability of the method, NE has become one of an active area of research and innovation within academia as well as the industry. The generated embedding representations pave the way for various applications, such as node classification [4, 9, 12], link prediction [3, 5, 17, 20], node clustering [1, 2, 18], name disambiguation[19, 21, 22], and visualization [13, 16], etc.

With the deepening of research, more and more works have begun to use attribute information of nodes as well as the topology information, to learn joint embedding representations. As attribute information is another non-negligible source which is capable of describing the characteristic of a particular node, besides, the attribute information also commonly exists on a graph in the real world. The advantage of learning such joint embedding space is evident, as the embedding results can fuse effective information from different perspectives simultaneously, which provides us a more comprehensive way to understand the essential nature of the target graph.

Hence, a lot of works have explored attributed graph embedding. By considering the two kinds of information as two different views of vertices, [10] proposed to apply co-regularized spectral clustering on the multi-view data collectively, and two co-regularization schemes, which are pair-wise and centroid based co-regularization respectively, are proposed to accomplish this. [11] proposed DANE to capture individual properties and correlations of topology and attribute. By enforcing consistency of the pairwise similarity in the original space and embedded space, DANE can learn the embedding representations for each space. Meanwhile, DANE tries to maximize correlations of embeddings (or equivalently minimize their disagreements) to seek a consensus embedding. Furthermore, [8] proposed a label informed attributed network embedding (LANE) framework to enhanced DANE through incorporating label information. By jointly embedding topological structure, node attributes and label information into a low-dimensional representation, LANE achieves significantly better performance compared with the state-of-the-art network embedding algorithms. But please be aware that, unlike other regular attribute-based NE methods that only rely on topology and attribute information, LANE also leverages the label information, which may explain their better performance.

However, most of these proposed methods are based on matrix factorization (e.g. eigen-decomposition) which presents obstacles in 1).a relatively slow procedure of eigen-decomposition, and 2).to require a large enough space to process the adjacency matrix.

To tackle the aforementioned challenges, we propose ***SANE***, a *locality-based* attribute-aware network embedding algorithm for **scalable** learning the joint representations from topological features among node pairs, as well as attribute information for each node. To be specific, on the topology space, we use truncated random walk to build a set of local neighbors to depict the target node. On attribute space, we focus on a small set of nodes that locate in

[*] These two authors are co-first authors. Datasets and a reference implementation of *SANE* are available via an email to first authors.

KDD MLG'18, August 2018, London, United Kingdom
© 2018 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06...$15.00
https://doi.org/10.475/123_4

the vicinity of the target node. As a result, this locality-based algorithm inherently has scalability. Then, by using stochastic gradient updates, the objective can be efficiently optimized. Moreover, by enforcing the alignment of a locally linear relationship between each node and its K-nearest neighbors on topology and attribute space, *SANE* returns joint embedding representations that maximize the likelihood of preserving the effective information from both topology and attributes space.

2 PROPOSED METHODS: SANE

Given a network $G = (V, E, A)$ with vertex set V, edge set E and attribute matrix A, *SANE* learns a mapping function $f : V \rightarrow \mathcal{R}^d$ to transfer the relationships among nodes into a vector space $\mathcal{R}^d$, and such space $\mathcal{R}^d$ reveals a much denser and more representative way to indicate the particular node relationships of interest. Here, d is the chosen dimension of the vector space.

To learn the topological features of the network, we choose a famous locality based network embedding method – node2vec [6], which learns embedding representations based on the corpus C of a graph generated from truncated second-order random walks. On the other hand, to leverage the attribute information on the network, we borrow the idea from "locally linear embedding (LLE)"[15], as LLE holds the assumption that the linear relationship between each node and its K-nearest neighbors should be preserved in the newly projected space. Similarly, in order to incorporate the features (represented as the linear relationship) extracted on node attributes into the final embedding representations, we consider the linear relationship as a proxy to transfer the effective information extracted from node attributes into the process of running node2vec. Then, through enforcing the alignment of the locally linear relationship from attribute and topology, *SANE* learns a jointly embedding space from both sides simultaneously. To achieve so, the objective is designed to maximize the degree of agreement of topology and attribute information by measuring the alignment of a linear relationship between the target node and its K-nearest neighbors. Eq.1 gives the objective function of *SANE*.

$$\begin{aligned} L = \sum_{u\in V}\sum_{v\in V} \#(u,v)\{ & \log\sigma(\overrightarrow{W^u_{node}}\cdot\overrightarrow{W^v_{context}}) \\ & + k\cdot \mathbb{E}_{v_N\sim P_C}[\log\sigma(-\overrightarrow{W^u_{node}}\cdot\overrightarrow{W^{v_N}_{context}})] \\ & + \lambda\cdot|\overrightarrow{W^u_{node}} - \sum_{i=1}^{K} R_{ui}\overrightarrow{W^{N^u_i}_{node}}|^2 \ \} . \end{aligned} \tag{1}$$

Here, $\#(u, v)$ denotes the number of times the pair (u, v) appears in C, $\overrightarrow{W^u_{node}}$ denotes the embedding representation of u in W_{node}, k is the number of negative samples, v_N is the sampled context based on the empirical unigram distribution $P_C(v)$, and σ is the sigmoid function. The first two terms of Eq.1 are same with the objective function in node2vec, which is trying to maximize the dot-product between the vectors of frequently occurring node pairs and minimize it for random ones. λ is a positive hyper parameter that balances the contribution of attribute information. N^u_i is the i^{th} $(i = 1, ..., K)$ nearest neighbor of node u on the attribute space A, and R_{ui} represents the linear coefficient of N^u_i for reconstructing node u using its K-nearest neighbors, which is obtained by minimizing

$$\varepsilon(R) = \sum_{u\in V}\left|\overrightarrow{A_u} - \sum_{i=1}^{K} R_{ui}\overrightarrow{A_{ui}}\right|^2 , \tag{2}$$

where K is the number of nearest neighbors that are used for reconstruction, and $\overrightarrow{A_{ui}}(i = 1, ..., K)$ represents the attribute vector of i^{th} K-nearest neighbors of the node u on the attribute space. Therefore, by enforcing the linear relationship, which is extracted within each node and its locality on the attribute space, to be preserved in the joint embeddings, *SANE* successfully fuses topology information and attribute information. In addition, the embedding representations generated by word2vec (the core algorithm of node2vec) has been proven to support arithmetic calculation [14], the objective function we proposed is actually trying to extend the number of nodes related to arithmetic calculation from 4 (element-wise addition/subtraction: $a - b = c - d$) to $K + 1$. Please note that, the linear mapping is only enforced on the node vectors W_{node}, rather than on W_{node} and $W_{context}$ together, which is mainly for lower time complexity.

To optimize Eq.1, we use stochastic gradient updates in the same way with node2vec. Alg.1 gives the pseudo code for the proposed method.

Algorithm 1: SANE: Scalable attribute-aware network embedding

```
Input: $G = (V, E, A)$
corpus ← Build corpus from node2vec procedure based on
    topology information;
neighbors, weights ← Find K-nearest neighbors and calculate
    $R$ according to Eq.2 based on attribute information;
Optimize $L$ according to Eq.1;
```

3 EXPERIMENTS

3.1 Datasets

Here we test the proposed method on three datasets. 1.**BlogCatalog** is a real-world social media dataset[1]; 2.**Cora** is a dataset based on citations between scientific papers[2]; 3.**PPI** is a graph built based on the protein-protein interaction (PPI)[3]. Table 1 gives the detailed information of these datasets.

Table 1: Detailed information of the datasets.

Name	# Nodes	# Edges	# Attributes	# Labels
Cora	2708	10858	1433	7
BlogCatalog	5196	171743	8189	6
PPI	56944	818716	50	121

[1] https://github.com/xhuang31/LANE

[2] https://linqs.soe.ucsc.edu/data

[3] https://downloads.thebiogrid.org/BioGRID

3.2 Results

Table 2 reports the F1-scores on *SANE*, along with five baseline algorithms: Local Linear Embedding (LLE) [15], node2vec [6], Accelerated Attributed Network Embedding (AANE) [7], Label Informed Attributed Network Embedding (LANE) [8] and MultiView [10] on the task of node classification.

Table 2: Classification performance (F1 score) of different methods on different datasets results.

	Datasets		
Algorithms	Cora	BlogCatalog	PPI
LLE	0.33	0.34	–
node2vec△	0.79	0.65	0.69
AANE	0.78	0.91	–
LANE▽	0.84	0.92	–
MultiView	0.82	0.83	–
SANE	0.84	0.92	0.77
SANE settings (λ, K)	(0.8, 127)	(8.0, 112)	(0.8, 81)
Gain of node2vec [%]	**6.3**	**41.5**	**11.6**

Notes △: We use default node2vec hyper parameters: $p = q = 1$; ▽: LANE needs a fraction of label information as well; −: Impractical to calculate due to scalability flaws of the algorithm; The number of embedding dimension of these methods are all fixed to $d = 96$;

From the results, it is evident that the joint embedding representations generated by *SANE* outperforms the single embedding from topology space only (node2vec) and attribute space only (LLE). For Cora and BlogCatalog dataset, *SANE* is able to achieve a level of performance comparable or better than other methods, even have the same performance with LANE (which needs label information as well in the training phase). Future work will focus on how to integrate label information into the framework of *SANE*.

For PPI dataset, as it needs more than $40G = (4 \times 50K)^2 bytes$ memory space to complete the matrix decomposition, LLE, AANE, LANE, and MultiView suffer from the scalability problem. Thus, it is impractical to get results from PPI dataset in a meaningful time (e.g., in several hours). On the contrary, as *SANE* uses the local view on both attribute and topological space, the empirical results show that it takes within 20 minutes to learn the joint embedding representations for PPI dataset.

3.3 Complexity analysis

According to Alg.1, *SANE* comes with three parts: 1). Building Corpus; 2). Building K-nearest neighbors and weights; 3). Learning joint embedding representations. Here we give a detailed analysis for each part.

(1) "Building Corpus:" As node2vec use a second order random walk to capture the interconnections between the neighbors of every node, the space complexity of building corpus is $O(\alpha^2 N)$, where N is the number of nodes, α is the average degree of the graph, and is usually small for the real-world networks. On the other hand, as the transition probabilities for the second order random walk can be precomputed, then by using alias sampling, the time complexity of building corpus is $O(L)$, L is the length of the random walk.
(2) "Building K-nearest neighbors and weights:" Here a ball-tree is constructed for finding K-nearest neighbors, therefore the time complexity and space complexity are $O(Dlog(K)Nlog(N))$ and $O(KN)$, respectively. Calculation of the weight matrix construction needs solving a $K \times K$ linear equation, thus the time complexity and space complexity are both $O(DNK^3)$, where D is the dimension of attribute vector, K is the number of nearest neighbors.
(3) "Learning joint embedding representations:" *SANE* needs to use two matrices W_{in} and W_{out} to store the embedding representations of nodes and contexts, respectively. Hence, the space complexity is $O(2Nd)$, where d is the embedding dimension. In addition, as *SANE* updates only a constant number of nodes at a time, the time complexity is $O(N)$.

Therefore, comparing with matrix factorization with quadratic time and space complexity, *SANE* is more suitable for scalable learning.

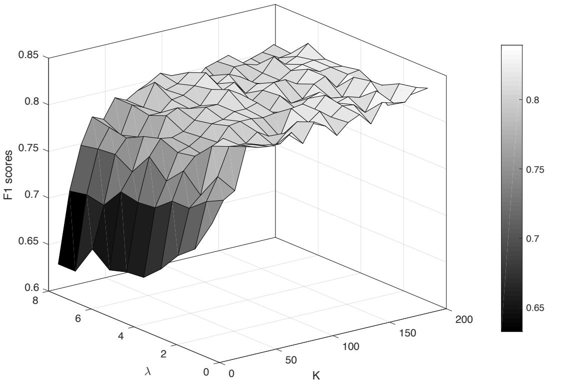

Figure 1: Micro-F1 score on different (λ, K) for Cora dataset. The λ axis denotes the hyperparameter in Eq.1 ranging from 0.4 to 8.0, K axis denotes the number of nearest neighbors ranging from 1 to 200, and y axis denotes the Micro-F1 score.

3.4 Parameter Sensitivity

As λ and K are two important hyperparameters to learn the joint representations, here we use Cora dataset as an example to demonstrate the performance with λ and K. Figure 1 reports the result with respect to $\lambda \in [0.4, 8]$, and $K \in [1, 200]$, on a fixed embedding size $d = 96$.

It is clear to see an upward trend influence of (λ, K) on performance. Among them, K has a more significant impact on performance than λ. To be specific, when K increases from 1 to 200, the classification performance in terms of Micro-F1 grows rapidly; After that, performance turns to grow slowly. This phenomenon makes sense as the more neighbors are included, the more accurate they can describe the target node. On the other hand, too many neighbors always contain too many redundant information, which has a minor contribution for learning the joint representations.

4 CONCLUSION

Jointly incorporating the network topology and node attributes into network embedding is promising but challenging. Therefore, we propose a novel and scalable framework *SANE*, which learns unified embedding representations by enforcing the alignment of locally linear mapping from the two spaces. Without the eigendecomposition of a large matrix, *SANE* can easily support large-scale attributed network embedding. Several experiments on real-world datasets demonstrate that *SANE* is indeed an effective and scalable algorithm.

REFERENCES

[1] Pin-Yu Chen, Baichuan Zhang, and Mohammad Al Hasan. 2017. Incremental eigenpair computation for graph Laplacian matrices: theory and applications. *Social Network Analysis and Mining* 8, 1 (2017), 4.

[2] Pin-Yu Chen, Baichuan Zhang, Mohammad Al Hasan, and Alfred O Hero. 2016. Incremental Method for Spectral Clustering of Increasing Orders. In *KDD Workshop on Mining and Learning with Graphs*.

[3] Sutanay Choudhury, Khushbu Agarwal, Sumit Purohit, Baichuan Zhang, Meg Pirrung, William Smith, and Mathew Thomas. 2017. NOUS: Construction and Querying of Dynamic Knowledge Graphs. In *Proceedings of 33rd IEEE International Conference on Data Engineering*. 1563–1565.

[4] Vachik Dave, Baichuan Zhang, Pin-Yu Chen, and Mohammad Al Hasan. 2018. Neural-Brane: Neural Bayesian Personalized Ranking for Attributed Network Embedding. *arXiv preprint arXiv:1804.08774*.

[5] Vachik S. Dave, Mohammad Al Hasan, Baichuan Zhang, and Chandan K. Reddy. 2018. Predicting interval time for reciprocal link creation using survival analysis. *Social Network Analysis and Mining* 8, 1 (2018), 16.

[6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 855–864.

[7] Xiao Huang, Jundong Li, and Xia Hu. 2017. Accelerated attributed network embedding. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 633–641.

[8] Xiao Huang, Jundong Li, and Xia Hu. 2017. Label informed attributed network embedding. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 731–739.

[9] Przemyslaw Kazienko and Tomasz Kajdanowicz. 2012. Label-dependent node classification in the network. *Neurocomputing* 75, 1 (2012), 199–209.

[10] Abhishek Kumar, Piyush Rai, and Hal Daume. 2011. Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems*. 1413–1421.

[11] Jundong Li, Harsh Dani, Xia Hu, Jiliang Tang, Yi Chang, and Huan Liu. 2017. Attributed network embedding for learning in a dynamic environment. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 387–396.

[12] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2017. Attributed social network embedding. *arXiv preprint arXiv:1705.04969* (2017).

[13] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[15] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *science* 290, 5500 (2000), 2323–2326.

[16] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 1067–1077.

[17] Daixin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1225–1234.

[18] Xiaowei Xu, Nurcan Yuruk, Zhidan Feng, and Thomas AJ Schweiger. 2007. Scan: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 824–833.

[19] Baichuan Zhang. 2017. *Towards Name Disambiguation: Relational, Streaming, and Privacy-Preserving Text Data*. Ph.D. Dissertation. Purdue University, West Lafayette.

[20] Baichuan Zhang, Sutanay Choudhury, Mohammad Al Hasan, Xia Ning, Khushbu Agarwal, Sumit Purohit, and Paola Gabriela Pesntez Cabrera. 2016. Trust from the past: Bayesian Personalized Ranking based Link Prediction in Knowledge Graphs. In *SDM Workshop on Mining Networks and Graphs (MNG 2016)*.

[21] Baichuan Zhang, Murat Dundar, and Mohammad Al Hasan. 2017. Bayesian Non-Exhaustive Classification for Active Online Name Disambiguation. *arXiv preprint arXiv:1702.02287* (2017).

[22] Baichuan Zhang, Noman Mohammed, Vachik S. Dave, and Mohammad Al Hasan. 2017. Feature Selection for Classification under Anonymity Constraint. *Transactions on Data Privacy* 10, 1 (2017), 1–25.