Similarity-Aware Network Embedding with Self-Paced Learning

Chao Huang∗
University of Notre Dame
chuang7@nd.edu

Baoxu Shi∗
LinkedIn
bshi@nd.edu

Xuchao Zhang
Virginia Tech
xuczhang@vt.edu

Xian Wu
University of Notre Dame
xwu9@nd.edu

Nitesh V. Chawla
University of Notre Dame
nchawla@nd.edu

ABSTRACT

Network embedding, which aims to learn low-dimensional vector representations for nodes in a network, has shown promising performance for many real-world applications, such as node classification and clustering. While various embedding methods have been developed for network data, they are limited in their assumption that nodes are correlated with their neighboring nodes with the same similarity degree. As such, these methods can be sub-optimal for embedding network data. In this paper, we propose a new method named SANE, short for Similarity-Aware Network Embedding, to learn node representations by explicitly considering different similarity degrees between connected nodes in a network. In particular, we develop a new framework based on self-paced learning by accounting for both the explicit relations (i.e., observed links) and implicit relations (i.e., unobserved node similarities) in network representation learning. To justify our proposed model, we perform experiments on two real-world network data. Experiments results show that SANE outperforms state-of-the-art embedding models on the tasks of node classification and node clustering.

CCS CONCEPTS

• Theory of computation → Social networks; • Computing methodologies → Knowledge representation and reasoning; Learning latent representations.

KEYWORDS

Network embedding; Deep Neural Network; Self-paced Learning

1 INTRODUCTION

Mining large-scale information networks (e.g., social networks, academic networks) has benefited many real-world applications, such as friend recommendation and user classification in online social platforms [11]. In those applications, identifying effective features play a crucial role, however, involves huge amounts of human efforts and massive handcrafted feature engineering based on domain-specific knowledge. To tackle this issue, a line of research on network embedding [4, 8, 12], which aims to learn low-dimensional vector representations of nodes, has attracted a lot of attention. These methods have been shown to be effective in various network mining tasks (e.g., node classification and clustering).

In particular, existing network embedding techniques aim to learn node representations by preserving the proximities between nodes based on their structural properties. To simplify the design, these approaches assumed the similarities between all pairs of connected nodes in a network with the same degree. However, this assumption does not hold in reality where nodes may have various degrees of similarity with their neighbors. Take the network between users on Twitter as an example, the followers of a movie star could come from: i) his/her friends who are most familiar with this movie star; ii) the people who share similar interests to the movie star, such as his/her colleagues; iii) or the ordinary people who are only interested in the movie star, but do not share similar interests and are personally unrelated. Such difference in node (user) similarity is expected to have a clear effect on network representation learning. This leads to the question investigated in this paper: can we explore different degrees of node similarity to learn more robust node representations for a network?

To solve the above question, we develop a new embedding framework called Similarity-Aware Network Embedding (SANE), aiming to learn robust node representations by considering different degrees of node similarity in a network. Since the similarity between different pairs of nodes may be different, each relationship between the target node and its neighbors should be treated differently during the representation learning process. In this work, we develop a self-paced learning algorithm to automatically learn the weights of node correlations in learning node representations. Finally, we perform extensive experiments on two real-world network datasets to evaluate the performance of our representation learning framework. Experimental results on both multi-label classification and node clustering show that SANE outperforms state-of-the-art network embedding methods. In summary, we highlight the contributions of this paper as follows:
2 METHODOLOGY
2.1 SANE Model with Self-paced Learning

There exist several ways to infer the latent similarity degrees between different nodes, such as Expectation Maximization (EM) algorithm [2]. However, it heavily relies on the defined margin likelihood distribution and is very time consuming, which is not scalable to large-scale network data [6]. To address this issue, we propose to utilize the Self-Paced Learning (SPL) [6] to estimate the latent similarity degrees between nodes with no prior knowledge.

Inspired by the learning principle of humans: start with easier concepts and then gradually takes more complex examples into consideration [1], SPL, a learning algorithm, is proposed to mimic such training strategies to facilitate the learning process. In particular, SPL is proposed to quantify the easiness of each sample/data point as a hidden variable/weight. Given the training samples as \( D = (x_1, y_1), ..., (x_n, y_n) \), where \( x_i \in \mathbb{R}^m \) denotes the \( i \)-th observed sample, and \( y_i \) represents \( i \)-th label, SPL learns the objectives by jointly learning model parameters and the easiness of each training data point (i.e., hidden variable):

\[
\arg\max_{\theta, \psi} \sum_i -w_i \cdot L(y_i, g(x_i, \theta)) + f(w_i, t) \tag{1}
\]

where \( L(y_i, g(x_i, \theta)) \) denotes the loss function which calculates the cost between the real label \( y_i \) and the estimated label \( g(x_i, \theta) \). \( w_i \) indicates the easiness of the observed \( i \)-th training sample \( (w_i > 0) \). \( f(w_i, t) \) is the self-paced function which controls the pace to increase the difficulty level. The parameter \( t \) is the iteration index which is termed as "age" of the SPL model to control the learning pace. In particular, when \( t \) is small, the "easy" samples with small losses are considered. As \( t \) grows, more samples with larger losses will be gradually added into the training set.

In our work, we assume that a training node pair is proximate if their similarity is high. Motivated by SPL algorithm, we propose to first learn embedding vectors of nodes with higher similarity degree in order to avoid interfering by irrelevant neighbor nodes, then target at nodes with complex relationships between the already learned nodes. Incorporating the similarity degree into the SPL process is challenging due to dependencies existing between the similarity degrees of different node pairs. To incorporate the similarity degree of each training pair, the skip-gram based loss function could be defined as:

\[
\arg\max_{\theta} \sum_{u \in V} \sum_{c \in C(u)} w_{u,v} \log(\Pr(c|u; \theta)) + f(u, t), \tag{2}
\]

where \( w_{u,v} \) represents the similarity degree of the training pair \( (u, v) \). However, the calculation on the similarities between center node \( u \) and other nodes in \( \Pr(c|u; \theta) = \frac{\exp(X_{c,u})}{\sum_{v \in V} \exp(X_{v,u})} \) is computational expensive[4]. Therefore, in our framework, we adopt the negative sampling technique to modify the conditional probability \( \Pr(c|u; \theta) \) as follows:

\[
\Pr(c|u; \theta) \propto \log(\sigma(X_{c,u})) + \sum_j \log(\sigma(-X_{c,u} + X_{\theta} X_{c,j}^T)) \tag{3}
\]

By incorporating both the self-paced based skip-gram loss function and negative sampling strategy, we further give our loss functions as:

\[
L = \sum_{u \in X} \sum_{c \in C(u)} w_{u,v} \log(\sigma(X_{c,u} X_{\theta}^T))
+ \sum_j \log(\sigma(-X_{\theta} X_{c,j}^T)) + \lambda_1 \lambda_2 \|w_{e_i,u}\|^2.
\]

where \( \lambda_1 \) and \( \lambda_2 \) represents the similarity degree learning strength and difficulty increasing rate, respectively.

2.2 Model Optimization

In this work, we apply stochastic gradient decent algorithm to derive the solutions by maximizing objective function Eq. 4. Specifically, we partition the variables into two disjoint sets. In each iteration, our algorithm optimize one set of variables and keep another set of variables fixed. In the learning process, when we fix similarity degree \( \psi \), the skip-gram learning algorithm is employed to obtain the node embedding vectors which maximize the likelihood of preserving both the structural and similarity information in the network. As the number of iterations increases, more node training pairs with low similarity degree are fed into the learning process. Formally, we derive the gradients as follows:

\[
\frac{\partial L}{\partial w_{\theta}} = w_{\theta} \log(\sigma(X_{c,u} X_{\theta}^T)) + \lambda_1 \lambda_2 \|w_{\theta}\|^2,
\]

\[
\frac{\partial L}{\partial w_{\theta}} = \sum_j \log(\sigma(-X_{\theta} X_{c,j}^T)) + \lambda_1 \lambda_2 \|X_{\theta}\|^2,
\]

\[
\frac{\partial L}{\partial w_{\theta}} = \sum_j \log(\sigma(-X_{\theta} X_{c,j}^T)) + \lambda_1 \lambda_2 \|X_{\theta}\|^2,
\]

\[
\frac{\partial L}{\partial w_{\theta}} = \sum_j \log(\sigma(-X_{\theta} X_{c,j}^T)) - \sum_j \log(\sigma(X_{\theta} X_{c,j}^T)) X_{\theta} X_{c,j} X_{j,u}.
\]

3 EVALUATION
3.1 Experimental Setup

3.1.1 Data. We perform experiments to evaluate our SANE framework on the following two real-world network datasets:

- **Twitter Social Circle Data**: This dataset is collected from Twitter to contain the friend relationships (edges) between users (nodes). Each user is associated with 84 labels representing his/her profile information.
- **Facebook Social Circle Data**: Similar as the above Twitter data, this dataset contains the friendships (edges) between users...
We first present the evaluation results on the node classification task. We vary the training size from 5% to 90% using a stratified split and report the average performance.

SANE achieves the best performance in terms of Macro-F1 and Micro-F1 with different size of training set, which further demonstrate the efficacy of our SANE which is capable of learning significantly better node embeddings than existing state-of-the-art methods in multi-label classification task. In summary, the advantage of SANE lies in its proper consideration of different degrees of node similarity which are usually unavailable in network data.

3.3 Node Clustering

In our experiments, we also investigate how the latent representations learned by embedding methods can help the node clustering task on the aforementioned two datasets. The embeddings learned from each method is considered as input to k-means algorithm and the clustering results is evaluated in terms of Normalized Mutual Information (NMI) [4]. All clustering experiments are conducted 10 times and the average performance is reported in Table 4. In this table, we can observe that SANE outperforms other competitive baselines in node clustering task on both Facebook and Twitter datasets. In summary, SANE could generate more appropriate embeddings for nodes in the network as compared to baselines, which suggest its ability to jointly capture the underlying structural and semantic relationships between nodes during the process of network representation learning.

3.4 Parameter Study

In this subsection, we analyze the influences of key hyperparameters on the performance of our SANE model. Due to space limit, we only present the evaluation results of multi-label classification task on Facebook dataset in Figure 1. From the evaluation results, we can observe that the proposed SANE is not strictly sensitive to different hyperparameter settings. We could observe that increasing the embedding dimension slightly improve the performance till d reaches 128. We attribute the improvement to the stronger representation ability with larger hidden state dimensionality. Additionally, we can notice that the performance becomes stable as long as the path length l is above 120. The number of walks per node r is positively correlated with the classification accuracy. The performance is improved slightly as window size k_w increases and then saturates when k_w \geq 10. We further observe that both similarity degree learning strength \lambda_1 and difficulty increasing rate \lambda_2 have low impact on the model performance.

4 CONCLUSION

This paper presents a similarity-ware embedding method for network embedding with a self-paced learning framework. The proposed approach can jointly incorporate the network structure and semantic relationship information between nodes into the network representation learning process. We evaluate the performance of our proposed approach on two real-world networks in both multi-label classification and node clustering tasks. Experimental results show that our SANE significantly outperforms competitive baselines by generating better embeddings for nodes in a network.

ACKNOWLEDGMENTS

This work is supported in part by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research

<table>
<thead>
<tr>
<th>Table 1: Dataset Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
</tr>
<tr>
<td>Twitter Social Circle</td>
</tr>
<tr>
<td>Facebook Social Circle</td>
</tr>
</tbody>
</table>
Table 2: Multi-label node classification results on Facebook data.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Model</th>
<th>Training Set %</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-F1</td>
<td>MF</td>
<td>0.6800</td>
<td>0.7224</td>
<td>0.7647</td>
<td>0.7792</td>
<td>0.7894</td>
<td>0.7951</td>
<td>0.7981</td>
<td>0.8021</td>
<td>0.8130</td>
<td>0.8148</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Walklets</td>
<td>0.6701</td>
<td>0.7058</td>
<td>0.7344</td>
<td>0.7491</td>
<td>0.7567</td>
<td>0.7629</td>
<td>0.7655</td>
<td>0.7681</td>
<td>0.7720</td>
<td>0.7735</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.6566</td>
<td>0.6928</td>
<td>0.7298</td>
<td>0.7444</td>
<td>0.7602</td>
<td>0.7644</td>
<td>0.7689</td>
<td>0.7742</td>
<td>0.7833</td>
<td>0.7856</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LINE</td>
<td>0.6578</td>
<td>0.6973</td>
<td>0.7324</td>
<td>0.7475</td>
<td>0.7633</td>
<td>0.7682</td>
<td>0.7734</td>
<td>0.7820</td>
<td>0.7889</td>
<td>0.7912</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DeepWalk/Node2Vec</td>
<td>0.7298</td>
<td>0.7661</td>
<td>0.8008</td>
<td>0.8171</td>
<td>0.8261</td>
<td>0.8313</td>
<td>0.8359</td>
<td>0.8428</td>
<td>0.8501</td>
<td>0.8515</td>
<td></td>
</tr>
<tr>
<td>SANE</td>
<td>0.7478</td>
<td>0.8005</td>
<td>0.8435</td>
<td>0.8667</td>
<td>0.8830</td>
<td>0.8941</td>
<td>0.9065</td>
<td>0.9083</td>
<td>0.9159</td>
<td>0.9202</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Multi-label node classification results on Twitter data.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Model</th>
<th>Training Set %</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
<th>60%</th>
<th>70%</th>
<th>80%</th>
<th>90%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Micro-F1</td>
<td>MF</td>
<td>0.4341</td>
<td>0.4155</td>
<td>0.5226</td>
<td>0.5633</td>
<td>0.5936</td>
<td>0.6167</td>
<td>0.6292</td>
<td>0.6425</td>
<td>0.6594</td>
<td>0.6158</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Walklets</td>
<td>0.3591</td>
<td>0.4280</td>
<td>0.4947</td>
<td>0.5253</td>
<td>0.5444</td>
<td>0.5610</td>
<td>0.5656</td>
<td>0.5746</td>
<td>0.5759</td>
<td>0.5481</td>
<td></td>
</tr>
<tr>
<td></td>
<td>BPR</td>
<td>0.3302</td>
<td>0.3866</td>
<td>0.4678</td>
<td>0.4983</td>
<td>0.5323</td>
<td>0.5448</td>
<td>0.5542</td>
<td>0.5643</td>
<td>0.5745</td>
<td>0.5437</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LINE</td>
<td>0.3706</td>
<td>0.4462</td>
<td>0.5292</td>
<td>0.5693</td>
<td>0.6046</td>
<td>0.6239</td>
<td>0.6356</td>
<td>0.6495</td>
<td>0.6574</td>
<td>0.6141</td>
<td></td>
</tr>
<tr>
<td></td>
<td>DeepWalk/Node2Vec</td>
<td>0.4527</td>
<td>0.5412</td>
<td>0.6224</td>
<td>0.6593</td>
<td>0.6829</td>
<td>0.7033</td>
<td>0.7181</td>
<td>0.7285</td>
<td>0.7276</td>
<td>0.6810</td>
<td></td>
</tr>
<tr>
<td>SANE</td>
<td>0.3079</td>
<td>0.6220</td>
<td>0.7125</td>
<td>0.7574</td>
<td>0.7826</td>
<td>0.8124</td>
<td>0.8258</td>
<td>0.8363</td>
<td>0.8345</td>
<td>0.8376</td>
<td>0.7861</td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: Hyper-parameter sensitivity evaluation of SANE in multi-label classification.

Table 4: Node clustering results on both Facebook and Twitter data in terms of Normalized Mutual Information (NMI).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Facebook Dataset</th>
<th>Twitter Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>MF</td>
<td>0.7871</td>
<td>0.7168</td>
</tr>
<tr>
<td>Walklets</td>
<td>0.7912</td>
<td>0.8570</td>
</tr>
<tr>
<td>BPR</td>
<td>0.7943</td>
<td>0.5628</td>
</tr>
<tr>
<td>LINE</td>
<td>0.7291</td>
<td>0.7500</td>
</tr>
<tr>
<td>DeepWalk/Node2Vec</td>
<td>0.7854</td>
<td>0.8621</td>
</tr>
<tr>
<td>SANE</td>
<td>0.7958</td>
<td>0.8892</td>
</tr>
</tbody>
</table>

REFERENCES


[11] Yizhou Sun, Jiawei Han, and etc. 2012. When will it happen?: relationship prediction in heterogeneous information networks. In WSDM. ACM, 663–672.