PRRE: Personalized Relation Ranking Embedding for Attributed Networks

Sheng Zhou[1,2,3], Hongxia Yang[3], Xin Wang[4], Jiajun Bu[1,2,3],
Martin Ester[5], Pinggang Yu[1,2,3], Jianwei Zhang[3], Can Wang[1,2,3]*
[1]College of Computer Science, Zhejiang University, China
[2]Zhejiang Provincial Key Laboratory of Service Robot, China
[3]Alibaba-Zhejiang University Joint Institute of Frontier Technologies, China
[4] Alibaba Group, China [5] Tsinghua University, China [6] Simon Fraser University, Canada
{zhen_zhang,bjj,zhousheng_zju,pgyu,wcan}@zju.edu.cn,ester@cs.sfu.ca,xin_wang@tsinghua.edu.cn,{yang.yhx,
zhangjianwei.zjw}@alibaba-inc.com,

ABSTRACT

Attributed network embedding focuses on learning low-dimensional latent representations of nodes which can well preserve the original topological and node attributed proximity at the same time. Existing works usually assume that nodes with similar topology or similar attributes should also be close in the embedding space. This assumption ignores the phenomenon of *partial correlation* between network topological and node attributed similarities i.e. nodes with similar topology may be dissimilar in their attributes and vice versa. Partial correlation between the two information sources should be considered especially when there exist fraudulent edges (i.e., information from one source is vague) or unbalanced data distributions (i.e, topology structure similarity and node attribute similarity have different distributions). However, it is very challenging to consider the partial correlation between topology and attributes due to the heterogeneity of these two information sources. In this paper, we take partial correlation between topology and attributes into account and propose the **P**ersonalized **R**elation **R**anking **E**mbedding (**PRRE**) method for attributed networks which is capable of exploiting the partial correlation between node topology and attributes. The proposed PRRE model utilizes two thresholds to define different node relations and employs the Expectation-Maximization (EM) algorithm to learn these thresholds as well as other embedding parameters. Extensive experiments results on multiple real-world datasets show that the proposed PRRE model significantly outperforms the state-of-the-art methods in terms of various evaluation metrics.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics; • **Networks** → Network reliability;

KEYWORDS

Attributed network embedding, partial correlation, relation ranking

ACM Reference Format:
Sheng Zhou, Hongxia Yang, Xin Wang, Jiajun Bu, Martin Ester, Pinggang Yu, Jianwei Zhang, Can Wang. 2018. PRRE: Personalized Relation Ranking Embedding for Attributed Networks. In *The 27th ACM International Conference on Information and Knowledge Management (CIKM '18), October 22–26, 2018, Torino, Italy.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3269206.3271741

1 INTRODUCTION

Network embedding has become an important research topic aiming to learn a low-dimensional latent representation of each node in a network for various learning tasks including link prediction, node classification and clustering [8, 25, 29, 35]. As more and more information becomes available, nodes in real-world networks are often associated with attributed features: people on social networks are often annotated by personal profiles including education, organization and location, etc. Papers on bibliographic networks such as DBLP usually belong to a journal/conference and have a publishing year, research topic, etc. Such networks with extra attribute information are known as attributed networks [26]. Recently, mining attributed networks has attracted lots of research interests including community detection [11, 40], network alignment [43] and network embedding [8, 25, 29]. Compared with vanilla network embedding, embedding in attributed networks focuses on capturing node relationships in terms of both topology and node attributes [14, 17, 24].

Existing attributed network embedding methods either learn node embedding for network topology and node attributes in an independent way [14, 42] or pass feature vectors into matrix factorization/deep neural networks to predict the network topology [15, 24, 36]. Some works also define a parametric function of the feature vectors to mine the network topology [17]. A critical yet common issue faced by these methods is that node attributes and network topology may be only weakly correlated to each other — nodes sharing similar attributes may actually lie far away from each other in the network and vice versa. We call this phenomenon *partial correlation* between network topology and node attributes. For instance, people living in the same block may share similar attributes such as addresses, schools and IP addresses etc, while there may be no direct connections between them in reality and they may even be not aware of each other. Take a bibliographic

[*]Corresponding author: wcan@zju.edu.cn

CIKM '18, October 22–26, 2018, Torino, Italy

ACM ISBN 978-1-4503-6014-2/18/10.
https://doi.org/10.1145/3269206.3271741

network as another example, two papers with citation relationship (one cites another) may focus on different topics and share only a small number of common attributes. Another problem for partial correlation is sparsity and noisiness of data. Real-world social networks are always large and sparse, and the observed relationships in an online social network may only reveal partial social circles of users.

In this paper, we present an attributed network embedding method which takes partial correlation between topology and attributes into account. Before discussing the details of the model, several challenges need to be addressed:

(1) Due to partial correlation, node proximity of the topology may be different from node proximity of attributes. Preserving node proximity for these two information sources may produce totally different embedding results.

(2) Some existing attributed network embedding methods simply employ a linear combination of the two information sources or treat topology and attributes independently, which may result in information loss especially when the network is sparse or its node attributes are noisy.

To tackle the above challenges, we propose the PRRE method which is capable of recognizing the partial correlation between topology and attributes. More concretely, we divide the relations in attributed networks into three groups:

(1) *Positive relations* are node pairs that are close to each other in terms of both topology and attributes.

(2) *Ambiguous relations* are node pairs that are close to each other in terms of topology but far away from each other in terms of attributes, or close to each other in terms of attributes but far in terms of topology structure.

(3) *Negative relations* are node pairs that are far away from each other in terms of both topology and attributes.

The Bayesian Personalized Ranking (BPR) framework [27] is a pairwise ranking method for recommender systems. The core idea of BPR is to learn a personalized ranking of items for each user, and the basic assumption is that a user prefers an observed item over all non-observed items. Inspired by the BPR framework, we develop a personalized ranking function for network embedding. Instead of directly preserving the proximities between nodes, the proposed PRRE model aims at fitting the ranking of above relations among the nodes such that the partial correlation between topology and node attributes can be handled. We utilize two thresholds that best separate the ‘far away from’ and ‘close to’ relations to categorize different relationships among nodes and employ the EM algorithm to learn these thresholds as well as other embedding parameters iteratively.

To summarize, we make the following contributions.

(1) We observe the partial correlation phenomenon between topology and attributes which affects the actual proximity among nodes in the embedding space.

(2) We propose the **P**ersonalized **R**elation **R**anking **E**mbedding (**PRRE**) model to embed attributed networks considering the phenomenon of partial correlation.

(3) We propose an EM-style iterative algorithm to simultaneously learn the best embedding parameters and the optimal thresholds separating different types of relations in a given network.

(4) We carry out extensive experiments on various real-world datasets. The results show that the PRRE model significantly outperforms state-of-the-art methods in terms of both node classification, link prediction and network visualization.

2 RELATED WORK

Some earlier works such as Local Linear Embedding (LLE) [28], ISOMAP [3] and Laplacian Eigenmap [4] generate low-dimensional manifolds which can model the nonlinear geometry of data. These approaches are part of dimensionality reduction techniques and can be regarded as the pioneer of network embedding.

Recent works focus more on embedding an existing network into a low-dimensional vector space to facilitate a better understanding of semantic relationships among nodes. Among them, DeepWalk [25] employs a truncated random walk to generate node sequences, which is treated as sentences in language models and fed to the Skip-gram model to learn the embeddings. Node2Vec [8] further advances the random walk based embedding algorithms by adding flexibility in exploiting neighborhoods. LINE [29] is proposed for large scale network, which preserves both first-order and second-order proximities to learn network representations. GraRep [6] can be regarded as an extension of LINE which considers high-order proximity. SDNE [34] incorporates graph structure into deep auto-encoder to preserve the highly non-linear first order and second order proximities.

All the above mentioned approaches, however, are limited to dealing with non-attributed networks. Attributed network analysis is then put forward due to the fact that numerous networks are often associated with abundant content describing attributes of each node. Some existing algorithms have investigated the possibility of jointly embedding these two information sources into a unified space to improve the performance of network embedding[44]. For example, TADW [39] extends DeepWalk by assuming that each node is associated with rich texts and incorporate text features into the matrix factorization framework. AANE [10] is a distributed embedding approach that jointly learns node representations by decomposing attribute affinity matrix and penalizing the embedding difference between connected nodes with network lasso regularization. SNE [17] generates embeddings by leveraging an end-to-end neural network model to capture the complex interrelations between network structure and node attribute information. GraphSAGE [9] is an inductive framework that leverages node feature information and network topology to efficiently generate node embeddings for previously unseen data. While associating network topology and node attribute, none of the above methods considers the partial correlation between two information sources.

There have also been semi-supervised approaches for attributed network embedding. TriDNR [24] uses information from three parties including node structure, node content, and node labels to jointly learn node representations. Planetoid [41] develops both transductive and inductive methods to jointly predict the class label and neighborhood context in the graph. SEANO [16] takes the input form the aggregation of input sample attributes and its average neighborhood attributes to mitigate the negative effect of outliers in the representation learning procedure. However, in real world data sets, labels of nodes are often unknown which limit the application of these methods.

3 THE PROPOSED PRRE METHOD

3.1 Preliminaries

Notations Let $G = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$ be an attributed network, where $\mathcal{V} = \{v_1, v_2 ... v_n\}$ denotes the set of n nodes in the network and $\mathcal{E}$ denotes the set of edges between nodes. $\mathbf{A} \in R^{n \times m}$ represents the attribute information of nodes, where m is the dimension of feature vectors and $\mathbf{A}_i$ is the feature vector corresponding to node v_i.

Definition 1. ***Attributed network embedding*** *Given an attributed network $G = \{\mathcal{V}, \mathcal{E}, \mathbf{A}\}$, attributed network embedding aims at learning a low-dimension representation $h_i \in R^d$ for each node v_i so that the learned representations can preserve both the network topology and node attribute proximities. d is the dimension of the learned representation and $d \ll m$.*

In the original network, the relationships among the nodes are represented by edges in network topology or distances between feature vectors. In the embedding space, the relationships are captured only by the distances between nodes in the latent vector space. The main challenge of attributed network embedding is to capture the two proximities into a unified vector space.

3.2 Personalized Relation Ranking in Attributed Networks

In attributed networks, proximities between nodes are related to two information sources: network topology and node attributes. For each information source, proximity can be measured by a similarity function: for attributes, cosine similarity of attribute vectors is widely used and other popular measures include the kernels, euclidean distance and manhattan distances [2]. For network topology, first order proximity is measured by the edges between nodes, second order proximity is determined by the neighbors of nodes through Jaccard's coefficient [22] or Adamic-Adar similarity [1], and high order proximity can be captured by random walk based methods such as Rooted PageRank [7] and PPMI [13]. We will discuss and experiment with the selection of these similarity measures in section 4.6.

Given the proximities between nodes, existing methods aim at preserving such proximities so that given a node v_i, nodes with higher proximities are closer in the embedding vector space. Inspired by the BPR model which models user's preference on observed items and non-observed items, proximities in networks can also be understood as personalized preferences: nodes with higher proximities are preferred with higher ranking. We use a personalized relation ranking function to define the preservation of proximities.

Definition 2. ***Personalized Relation Ranking****: Given a node v_i and another two nodes v_j and v_k in the network, if v_j has higher proximity with v_i compared to v_k, we define that the ranking of relation (v_i, v_j) is higher than relation (v_i, v_k) and denote $j \geq_i k$ to represent such ranking of relations.*

For all the triplets in the network, relation ranking has to meet the properties of a total order [27]:

1. **Totality:** $\forall v_i, v_j, v_k \in \mathcal{V} : j \neq k \Rightarrow j \geq_i k \vee k \geq_i j$
2. **Antisymmetry:** $\forall v_i, v_j, v_k \in \mathcal{V} : j \geq_i k \wedge k \geq_i j \Rightarrow j = k$
3. **Transitivity:** $\forall v_i, v_j, v_k, v_l \in \mathcal{V} : j \geq_i k \wedge k \geq_i l \Rightarrow j \geq_i l$

Network embedding that aims at preserving the proximities between nodes is then viewed as fitting the relation ranking function: for each triplet $\{v_i, v_j, v_k\}$ that satisfies the relation ranking $j \geq_i k$, in the embedding space, node v_j are closer to node v_i compared with node v_k. For two nodes $v_i, v_j \in \mathcal{V}$, their closeness in the embedding space is defined as:

$$\sigma_{ij} = \sigma(h_i^T h_j) = \frac{1}{1 + e^{-h_i^T h_j}}, \tag{1}$$

where h_i, h_j are the embedding vectors of nodes v_i, v_j, $\sigma(x)$ is the logistic sigmoid function which smooths the dot product of embedding vectors.

Fitting the relation ranking is equal to maximizing the following objective function:

$$\prod_{\{i,j,k\} \in \mathcal{V}} P(j \geq_i k). \tag{2}$$

We expect network embeddings to fit the relation ranking function for all the triplets $\{v_i, v_j, v_k\}$ in the network. However, capturing all the relation ranking has the following weakness:

(1) **Time Complexity** It is time consuming to fit all the relation rankings in the network, since there exist n^3 triplets in the network. Also, many real world networks are sparse and proximities based on network topology in such data is also sparse which means fitting all relation ranking will not provide useful information.

(2) **Noise for Embedding** Fitting all the relation ranking may also bring noise to the embedding. For example, the topology based similarities are usually small in large scale networks. As a result, a large amount of proximities between nodes is small or even zero. Fitting such a relation ranking is too strict and may overfit to noise.

Some existing methods have noticed this problem and negative sampling (NS) is one of the popular solutions. In NS, only partial negative samples are used. In the PRRE model, instead of considering all the ranking pairs, we first use thresholds to divide the relations into groups and then sample relations from each group for ranking. Given node v_i, v_j, if their similarity is larger than θ, it is defined as *positive relation* and otherwise, it is defined as *negative relation* and we name it as discrete relation ranking. It is worth noting that different thresholds will separate the relations into different groups, which will further affect the relation ranking and the embedding results. Thus the threshold should be learned from the data instead of pre-defined. With discrete relation ranking, we are able to learn a latent representation so that for each node v_i, nodes in groups with higher ranking are closer to v_i than nodes in groups with lower ranking in the embedding space.

Quality of threshold As discussed above, threshold selection is critical in the embedding process. We use the degree of separation [37, 38] between positive relations and negative relations imposed by threshold θ as the quantitative measure for the network topology as well as node attributes:

$$\begin{aligned} g(\theta_T) &= (\overline{S_T^P} - \theta_T)(\theta_T - \overline{S_T^N}), \\ g(\theta_A) &= (\overline{S_A^P} - \theta_A)(\theta_A - \overline{S_A^N}), \\ g(\theta_T, \theta_A) &= (\overline{S_T^P} - \theta_T)(\theta_T - \overline{S_T^N}) + (\overline{S_A^P} - \theta_A)(\theta_A - \overline{S_A^N}), \end{aligned} \tag{3}$$

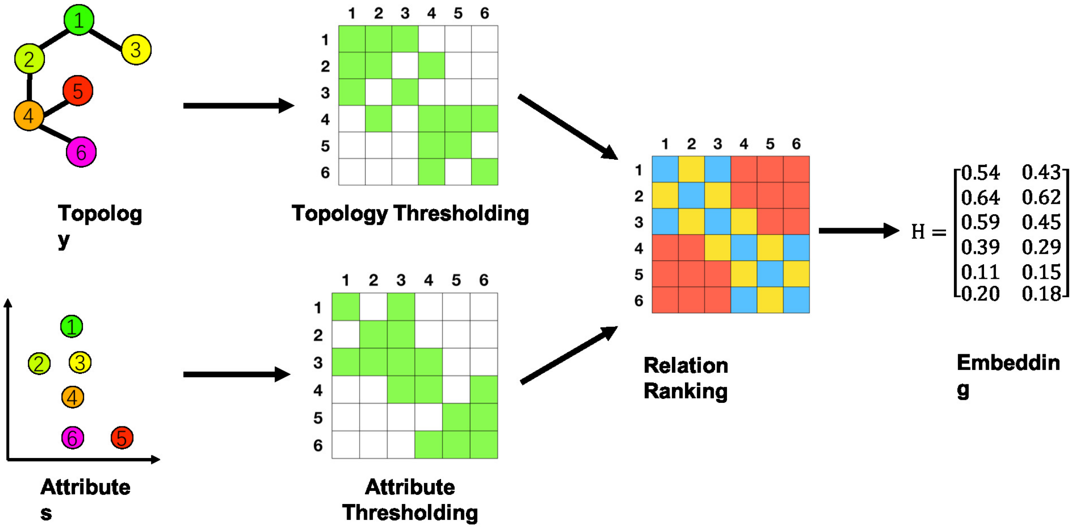


Figure 1: Personalized Relation Ranking Embedding in attributed networks. Green blocks and white blocks represent positive and negative relations based on single information source. Blue blocks, yellow blocks and red blocks represent positive, ambiguous and negative relations based on two information sources.

where θ_T, θ_A are the thresholds corresponding to network topology and node attributes, $\overline{S_T^S}, \overline{S_A^S}$ are the average similarities of all pairs with positive relations based on network topology and node attributes, $\overline{S_T^N}, \overline{S_A^N}$ are the average similarities of all pairs with negative relations based on network topology and node attribute.

Data likelihood To incorporate the threshold into the objective function so that the model is able to learn it in a principled manner, we use an exponential formulation for the likelihood, which can be written as:

$$\prod_{\{i,j,k\}\in\mathcal{V}} P(j \geq_i k|\theta, H) = \left[\frac{\sigma_{ij} - \sigma_{ik} + 1}{2}\right]^{\frac{1}{1+g(\theta)}}, \tag{4}$$

where the formulation of $(\sigma_{ij} - \sigma_{ik} + 1)/2$ keeps values in the range of $[0, 1]$. The exponential form is used to keep the monotonicity of both the quality of threshold and the relation ranking. We add denominator in the power for smoothness. Maximizing this likelihood can be reached by fitting relation ranking and improving the quality of threshold.

3.3 PRRE model

In this subsection, we first introduce the application of relation ranking embedding for single information source and then take the partial correlation into consideration.

From the network topology aspect, we first employ graph based similarity measure to calculate the similarity between feature vectors. Given node v_i and topology similarity threshold θ_T, if similarity between node v_i and v_j is larger than threshold θ_T, the relation between them is defined as **positive in network**, denoted as P_T; else the relation is defined as **negative in network**, denoted as N_T:

$$R_T(v_i, v_j) = \begin{cases} P_T, & S_T(v_i, v_j) > \theta_T, \\ N_T, & else, \end{cases} \tag{5}$$

where R_T is the network topology only relation and S_T is the network topology based similarity. PRRE for network topology (PRRE-N) aims at maximizing the likelihood:

$$\mathcal{J}(H, \theta_T) = \prod_{(i,j,k)} P(j \geq_i k|\theta_T, H). \tag{6}$$

From the node attribute aspect, we first employ cosine similarity to measure the similarity between feature vectors. Given node v_i and attribute similarity threshold θ_A, if similarity between node v_i and v_j is larger than threshold θ_A, the relation between them is defined as **positive in attribute**, denoted as P_A, else the relation is defined as **negative in attribute**, denoted as N_A:

$$R_A(v_i, v_j) = \begin{cases} P_A & S_A(v_i, v_j) > \theta_A, \\ N_A & else, \end{cases} \tag{7}$$

where R_A is the attribute only relation and S_A is the attribute based similarity. PRRE for node attributes (PRRE-A) aims at maximizing the likelihood:

$$\mathcal{J}(H, \theta_A) = \prod_{(i,j,k)} P(j \geq_i k|\theta_A, H). \tag{8}$$

The skip-gram model with negative sampling (SGNS) [21] is popular node embedding method, it models the conditional neighbor distribution of nodes and preserve the proximity between nodes. The objective function of SGNS can be written as:

$$\mathcal{J}_{SGNS} = log\sigma(h_j^T h_i) + \sum_{i=1}^{k} E_{j' \in P_n(i)}[log\sigma(-h_{j'}^T h_i)], \tag{9}$$

where node v_j is in the window of node v_i and node $v_{j'}$ is generated by negative sampling. Compared with PRRE-N, it is interesting to observe that they have the similar mathematical formulation. In fact, SGNS can be viewed as a special case of PRRE-N by assuming that

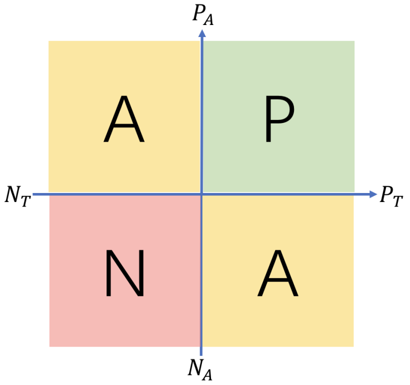


Figure 2: Relationships in attributed networks based on attribute and network topology

node in the window of v_i has the positive relation in network and node from negative sampling has the negative relation in network.

Given the definitions of relations based on topology and attributes, we have positive and negative relations on both sources of information. However the relatedness does not always correlated to each other as discussed in section 1: node pairs that positive in attribute may be negative in network and vice versa. We use Fig. 2 to illustrate such a phenomenon, for each node $v_i \in \mathcal{V}$, relations between v_i and the rest nodes can be put into such a two-dimension space. The horizontal coordinate refers to the strength of network topology proximity and the vertical coordinate refers to the strength of node attribute proximity, the x and y axis is defined by threshold θ_T and θ_A respectively. The thresholds separate the relation space into four parts and we further define the relation in attributed networks.

Definition 3. ***Positive(P), Ambiguous(A), Negative(N) relations**: for node v_i in attributed network G, if relations between node v_i and v_j are positive in both the network topology and node attributes aspect, the relation is defined as **Positive(P) relation**; if relations are negative in both aspects, the relation is defined as **Negative(N) relation**. if relation on only one information aspect is positive and another is negative, the relation is defined as **Ambiguous(A) relation**.*

$$R(v_i, v_j) = \begin{cases} P & R_T = P_T \ , \ R_A = P_A, \\ A & R_T = P_T/N_T, R_A = N_A/P_A, \\ N & R_T = N_T \ , \ R_A = N_A. \end{cases} \tag{10}$$

Based on the definition of above relations in attributed networks, for node $v_i \in \mathcal{V}$, relations between other nodes and v_i are expected to have the following personalized ranking: Positive(P)>Ambiguous(A)>Negative(N). The reason is straight forward that for node $v_i \in \mathcal{V}$, nodes with both positive relation in network topology and node attributes are supposed to be more similar with node v_i compared with nodes with one positive relation and one negative relation in network topology and node attribute. Nodes with both negative relation in network topology and node attributes are expected to have the lowest ranking. Preserving such relation ranking in the embedding space can be more flexible when the network and attribute information are partial correlated.

Loss function Concerning the relationships defined in attributed networks, a good embedding in such network should satisfy the relation ranking. The likelihood function is written as:

$$\begin{aligned} \mathcal{J}(H, \theta_T, \theta_A) = \prod_{i \in \mathcal{V}} \Bigg(\prod_{p \in P} \prod_{a \in A} P(p \geq_i a | \theta_A, \theta_T, H) \\ \prod_{a \in A \prod_{n \in N}} P(a \geq_i n | \theta_A, \theta_T, H) \Bigg). \end{aligned} \tag{11}$$

where P is the set of nodes with positive relations, A is the set of nodes with ambiguous relations and N is the set of nodes with negative relations.

3.3.1 *Joint learning of embedding and parameters.*

Our goal is to learn representations for nodes of the attributed networks and parameters that satisfy the relation rankings. The likelihood function to maximize is:

$$\begin{aligned} &\mathcal{J}(H, \theta_T, \theta_A) = ln\mathcal{J}(H, \theta_T, \theta_A) - \lambda_h \sum_{i \in \mathcal{V}} ||h_i||^2 \\ &= \sum_{i \in \mathcal{V}} \Bigg(\sum_{p \in P} \sum_{a \in A} \frac{1}{1 + g(\theta_T, \theta_A)} ln \frac{\sigma_{ip} - \sigma_{ia} + 1}{2} + \\ &\sum_{a \in A} \sum_{n \in N} \frac{1}{1 + g(\theta_T, \theta_A)} ln[\frac{\sigma_{ia} - \sigma_{in} + 1}{2}] \Bigg) - \lambda_h \sum_{i \in \mathcal{V}} ||h_i||^2 \end{aligned} \tag{12}$$

Figure 1 illustrates the framework of PRRE in attributed networks. Input network topology and node attribute information, the similarity measure and corresponding threshold are employed to define the positive and negative relation in single information source. Green blocks and white blocks represent positive and negative relation based on single information source. Then the single thresholding result is combined into the final relation: positive relation represented by blue blocks, ambiguous relation represents by yellow blocks and negative relation represented by red blocks. Finally, these relations guide the network embedding to best satisfy the relation ranking.

3.4 Updating algorithm

We employ the EM algorithm as well as mini-batch gradient ascent to learn the embedding vectors H and the thresholds θ_A, θ_T. In the EM algorithm, the hidden parameters are the similarity thresholds θ_A, θ_T which should be learned from the data.

E-step In each iteration t, given the thresholds θ_A, θ_T, we first calculate average similarity of node pairs and separate all the relations into positive, ambiguous and negative groups. Then, we draw a node v_i uniformly at random from $\mathcal{V}$ and draw a node from each relation group of node v_i. When the relation group is empty, we just skip it and the model only need to satisfy the partial ranking. All samples are drawn independently until the batch size s is reached. Gradients are computed using the following partial derivative formulas.

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial h_i} = &\frac{1}{1 + g(\theta_T, \theta_A)} \Bigg[\sum_{(i,p,a)} \frac{\sigma_{ip}(1 - \sigma_{ip}) h_p - \sigma_{ia}(1 - \sigma_{ia}) h_a}{\sigma_{ip} - \sigma_{ia} + 1} \\ &+ \sum_{(i,a,n)} \frac{\sigma_{ia}(1 - \sigma_{ia}) h_a - \sigma_{in}(1 - \sigma_{in}) h_n}{\sigma_{ia} - \sigma_{in} + 1} \Bigg] - \lambda_h h_i, \end{aligned} \tag{13}$$

$$\frac{\partial \mathcal{J}}{\partial h_p} = \frac{1}{1+g(\theta_T,\theta_A)} \sum_{(i,p,a)} \frac{\sigma_{ip}(1-\sigma_{ip})h_i}{\sigma_{ip}-\sigma_{ia}+1} - \lambda_h h_p, \tag{14}$$

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial h_a} =& \frac{1}{1+g(\theta_T,\theta_A)} \Bigg[\sum_{(i,p,a)} \frac{-\sigma_{ia}(1-\sigma_{ia})h_i}{\sigma_{ip}-\sigma_{ia}+1} + \\ & \sum_{(i,a,n)} \frac{\sigma_{ia}(1-\sigma_{ia})h_i}{\sigma_{ia}-\sigma_{in}+1} \Bigg] - \lambda_h h_a,\end{aligned} \tag{15}$$

$$\frac{\partial \mathcal{J}}{\partial h_n} = \frac{1}{1+g(\theta_T,\theta_A)} \sum_{(i,a,n)} \frac{-\sigma_{in}(1-\sigma_{in})h_i}{\sigma_{ia}-\sigma_{in}+1} - \lambda_h h_n, \tag{16}$$

M-step After updating the embedding vectors, we update the threshold θ_T, θ_A, the derivate can be computed as follows:

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \theta_T} =& -\frac{\overline{S_T^P} + \overline{S_T^N} - 2\theta_T}{[1+g(\theta)]^2} \Bigg[\sum_{(i,p,a)} ln \frac{\sigma_{ip}-\sigma_{ia}+1}{2} \\ & + \sum_{(i,a,n)} ln \frac{\sigma_{ia}-\sigma_{in}+1}{2} \Bigg] - \lambda_\theta \theta_T,\end{aligned} \tag{17}$$

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \theta_A} =& -\frac{\overline{S_A^P} + \overline{S_A^N} - 2\theta_A}{[1+g(\theta)]^2} \Bigg[\sum_{(i,p,a)} ln \frac{\sigma_{ip}-\sigma_{ia}+1}{2} \\ & + \sum_{(i,a,n)} ln \frac{\sigma_{ia}-\sigma_{in}+1}{2} \Bigg] - \lambda_\theta \theta_A.\end{aligned} \tag{18}$$

Gradient Ascent Update To maximize the likelihood, we use standard gradient ascent to update the parameters:

$$x^{(t+1)} = x^{(t)} + \eta \frac{\partial \mathcal{J}}{\partial x^{(t)}}, \tag{19}$$

where x refers to any parameter to be updated, t is the iteration number and η denotes the updating step size. The algorithm terminates when the absolute difference between the losses in two consecutive iterations is less than 10^{-5}. The pseudocode of the learning algorithm is presented in Algorithm 1.

3.4.1 Computational Complexity Analysis. In E-step, we update embedding **H** for all nodes in the network. The frequency of update is related to the triplets in the batch, as a result, the complexity of E-step is **O(nds)** where n is the number of nodes, d is the dimension of embedding vector and s is number of sampled triplets of each node. In M-step, we update threshold θ_T, θ_A. The complexity of M-step is **O(nds)** where s is the size of batch. Overall, the time complexity of PRRE is **tO(nds)** where t is the number of iterations.

4 EXPERIMENTAL EVALUATION

In this section, we report the experimental results on real world datasets to answer the following questions:

(1) Does partial correlation between topology and node attributes exist in real-world attributed networks?

(2) Can PRRE learn better node representations compared to state-of-the-art network embedding methods?

(3) How does similarity measures affect PRRE model and how to select the similarity for different datasets?

Algorithm 1 PRRE for attributed networks

```
Input: $G = \{\mathcal{V}, E, A\}, d$
Output: $H \in R^{n \times d}$
Compute similarity matrix $S_A \in R^{n \times n}$ and $S_T \in R^{n \times n}$ using selected similarity measure
Initialize $\theta_A, \theta_T$ and $H \sim U(0,1)$
while t < max_iter and $\triangle \mathcal{J} < \epsilon$ do
    Sample batch $B$ with size $s$,
    for $v_i \in \mathcal{V}$ do
        Compute Positive/Ambiguous/Negative pairs using current thresholds $\theta_T, \theta_A$
    end for
    Compute the gradients for $h_i, h_p, h_a, h_n$ by Equation [13], [14], [15], [16].
    Update H by Equation [19]
    Compute the gradients for $\theta_T, \theta_A$ by Equation [17],[18].
    Update $\theta_T, \theta_A$ by Equation [19]
end while
```

4.1 Datasets

We conduct node classification, link prediction and network visualization experiments on several real-world datasets and the details of datasets are described as follows with dataset statistics summarized in Table 1:

Citation Graphs Cora, Citeseer and Pubmed are three available public datasets consisting of bibliographic data that have been widely used in evaluating network embedding [18, 20, 32]. Nodes represent the published paper and edges represent that if paper cites or is cited by other papers. Each published paper is associated with a word vector, in which the 0/1-valued elements represent the absence/presence of the corresponding word from the dictionary. Labels indicate the research categories that each paper belongs to.

Social Networks BlogCatalog is an online blogger community, where users follow each other and form a network. Nodes represent the users in the community and edges represent the following relationship between users.GraphSAGE Users are allowed to generate keywords as a short description of their blogs which are severed as node attributes. Labels indicate the predefined categories where users register their blogs to.

4.2 Baselines

We compare the PRRE model with several state-of-the-art network embedding methods and the variants of PRRE, which can be divided into the following 4 groups:

(1) **Attributes Only**: This group of algorithms only consider node attributes and transform the attribute vectors to low dimensional representations. We compare with these methods to reveal the effectiveness of attributes in node classification and choose SVM and AutoEncoder [33] as baseline methods in this group.

(2) **Network Only**: This group of algorithms leverage network topology information but ignores node attributes. Representative works include DeepWalk [25] and Node2Vec [8], which use truncated random walks to generate node sequences and employ skip-gram model for node representation learning. LINE [29] is one of the state-of-the-art embedding algorithms for large-scale networks.

	# node	# edge	# feature	# label
Citeseer	3264	4714	3668	6
Cora	2708	5429	1433	7
Pubmed	19717	44338	500	3
Blogcatalog	5196	171743	8189	6

Table 1: Datasets statistics

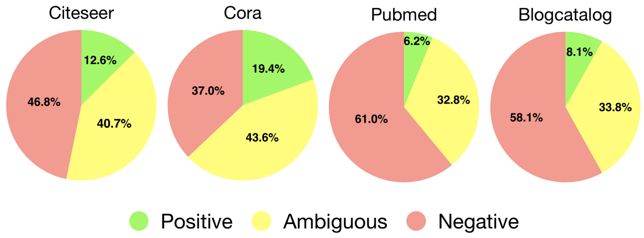


Figure 3: Distribution of types of relations in real-world datasets.

It preserves both first and second-order proximities between the nodes. GraRep [6] extends to high-order proximity and uses the SVD to train the model. It also directly concatenates the representations of first and high orders.

(3) **Attributes and Networks**: This group of algorithms try to preserve both node attributes and network topology proximities and are most closely related to our work. We choose the state-of-the-art methods as baseline methods, including AANE[10], SNE[17], GraphSAGE[9] and TADW[39]. The details of these methods have been described in section 2.

(4) **PRRE and variants:** This group of algorithms include PRRE and its variants: PRRE-A and PRRE-N. PRRE-A and PRRE-N are variants of PRRE that only use node attributes and network topology and their purpose is to preserve the personalized relation ranking based on attributes or topology only. We compare PRRE-A and PRRE-N with attributes only and network only baselines to show that personalized relation ranking framework can capture the relationships between nodes. Finally, all these variants show the advantage of preserving positive, ambiguous and negative relation ranking for network embedding.

For all baselines, we used the open source codes released by the original authors. The parameters for baselines are tuned to be optimal by greedy search. The code of our proposed method can be found in [1].

4.3 Exploring Relations in Attributed Network

In this subsection, we study the partial correlated relations in real-world attributed networks. We use the average similarity of all node pairs as the Relations in attributed networks have been defined in section 3. For attributes similarity, we calculate cosine similarity on the feature vectors. For topology similarity, we select RPR similarity to capture first, second and high order proximities in network topology. Figure 3 illustrates the distributions of positive, ambiguous and negative relations in four real-world datasets. From the figure, we get the following observations:

(1) There exist partial correlated (ambiguous) relations between nodes in attributed network, which refers to the yellow sector in each fan diagram. An even more interesting finding is that the percentage of ambiguous relations is larger than the percentage positive relations. This confirms that in attribute network embedding, missing the partial correlated relations may lose much information.

(2) In most of the datasets, negative relations have the biggest percentage. This is reasonable since in real-world networks, a node shares only a limit number of attributes with other nodes and most pairs of nodes are far from each other in network. Especially, in pubmed and blogcatalog datasets, over half of the relations are negative since these two networks are large and sparse.

4.4 Node classification

Node classification has been widely used in literature to evaluate network embeddings [25, 30]. In this subsection, we report the experimental results of node classification on labeled datasets. Following the experimental procedure of existing works [6, 8], for each method, we first train models on the training sets to obtain node representations. Then we randomly sample 30% labeled nodes to train a SVM classifier and the rest of the nodes are used to test the model. We repeat this process 10 times, and report the average performance in terms of both Macro-F1 and Micro-F1 [12]. The detailed results are shown in Table 2. To summarize, we have the following observations:

(1) Among these competitors, methods using both node attribute information and network topology information have better performances compared to methods using single information source. This shows that in attributed network embedding, preserving proximity from both information sources can learn better latent representations of nodes. Another interesting observation is that *attribute only* methods gain better results compared with *network only* methods in most of the datasets. This can be explained since network topology alone provides very limited information (compared to node attributes) especially when the network is large and sparse.

(2) Among methods using single information source, our relation ranking methods PRRE-A, PRRE-N achieves the best performance. This result proves the framework of personalized relation ranking can capture the relationships among nodes on both graph topology and node attribute. In the attribute only group, we observe that SVM performs better than Autoencoder in all datasets, which suggests that dimension reduction may lose useful attribute information. In network only group, we observe that GraRep outperforms the other baselines, likely since GreRep preserves high order proximities in network.

(3) Among methods using both node attribute information and graph topology, PRRE achieves better performance than the baseline methods. This confirms the usefulness of considering partially correlated relations in attributed network. Also, comparing PRRE and its variants PRRE-A, PRRE-N, we also observe significantly improvements, which indicates that under the framework of relation ranking, considering partial correlation phenomenon can significantly improve the performance of network embedding.

4.5 Link Prediction

In this subsection, we evaluate the network embedding results via link prediction. Following the same experimental procedure in many exiting works [8, 31, 34], we randomly hold out part of the

[1] https://github.com/zhoushengisnoob/PRRE

Group	Algorithm	Citeseer		Cora		Pubmed		Blogcatalog	
		Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
Atribute Only	SVM	0.7068	0.6353	0.6959	0.6746	0.8074	0.7987	0.8185	0.8174
	AutoEncoder	0.6946	0.6299	0.6787	0.6518	0.7873	0.7769	0.7989	0.7954
Network Only	DeepWalk	0.6811	0.6236	0.7770	0.7621	0.7439	0.7149	6459	0.6395
	Node2Vec	0.6909	0.6407	0.7925	0.7721	0.7585	0.7233	0.6507	0.6451
	LINE	0.6127	0.5467	0.7275	0.7078	0.7278	0.7032	0.6498	0.6421
	GraRep	0.7142	0.6452	0.7954	0.7722	0.7681	0.7208	0.7399	0.7359
Attribute + Network	SNE	0.7356	0.6678	0.8017	0.7834	0.8221	0.8529	0.8740	0.8721
	GraphSAGE	0.7119	0.6377	0.7917	0.7782	0.8141	0.8045	0.8289	0.8204
	AANE	0.7014	0.6245	0.7201	0.6844	0.8202	0.8136	0.8560	0.8558
	TADW	0.7435	0.6862	0.8304	0.8164	0.8337	0.8197	0.8908	0.8895
RRNE variant	PRRE-A	0.7081	0.6357	0.7436	0.7113	0.8026	0.8023	0.8271	0.8239
	PRRE-N	0.7230	0.6493	0.8166	0.7920	0.8054	0.7912	0.7042	0.6761
	PRRE	**0.7637**	**0.6935**	**0.8521**	**0.8283**	**0.8406**	**0.8235**	**0.9189**	**0.9171**

Table 2: Classification performance in different datasets. We use bold font to highlight the winners.

existing links as positive instances in test set and randomly sample the same amount of non-existing links as negative instances. The residual network is used to train the network embedding models. After obtaining the latent representations of each node, we evaluate link prediction in the labeled edge dataset. Specifically, we rank both positive and negative instances according to the cosine similarity function. To judge the ranking quality, we employ the AUC[2] to evaluate the ranking list and a higher value indicates a better performance.

We only compare methods that use both topology and attributes information here and baselines use single information source are not included. The reason is that there is no decrease trend for attribute only baselines with different training percent and network only methods are observed poor performance on the residual network. Figure 4 presents the link prediction results on four datasets and we have the following observations:

(1) With decreasing percentage of training edges, the performances of all methods decreases. This is reasonable since with fewer training edges, networks become sparser and the topology is less reliable. Also, different methods have different speed of decreasing which tells the robustness of the methods when network topology becomes noisy. We observe that the proposed proposed method PRRE and baseline method TADW decrease slower than other baseline methods.

(2) Among methods that use both network topology and node attribute, PRRE achieves the best performance. This demonstrates that considering partial correlated relations including the uncorrelated relations can capture more information among nodes. Network embedding that satisfy such personalized relation ranking can better represent the nodes in the latent space.

4.6 Comparison of similarity measures

There exist many candidate similarity measures based on network topology and different measures further influence the ranking of relations. To study the difference between them, we evaluate network only PRRE-N with different similarity measures on link prediction. According to previous work [13, 22, 23], the selected similarity measure can be formulated as:

(1) Rooted Page Rank (RPR) [7] describes the probability of a random walk from start nodes locating at end node in the steady state.

$$S_{RPR} = (I - \alpha P)^{-1} \cdot (1-\alpha)I, \tag{20}$$

where S is the similarity matrix, I is the identity matrix, $\alpha \in [0, 1)$ is the probability to randomly walk to a neighbor and P is the probability transition matrix satisfying that $\sum_{i=1}^{N} P_{ij} = 1$.

(2) Adamic-Adar similarity (AA) [1] is a variant of common neighbors that assign each neighbor a weight based on the degree.

$$S_{AA} = I^{-1} \cdot A \cdot D \cdot A, \tag{21}$$

where D is the diagonal matrix and A is the adjacent matrix.

(3) Jaccard's coefficient (Jaccard) [22] is a similarity based on the neighborhood of nodes.

$$S_{Jaccard}(i,j) = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}, \tag{22}$$

where N_i is the set of neighbors of node v_i.

(4) Positive Pointwise Mutual Information (PPMI) [13] is related to the statistic of graph context generated by random walk.

$$S_{PPMI}(i,j) = max(0, log\frac{\#(i,j) \cdot |C|}{\#(i) \cdot \#(j)}), \tag{23}$$

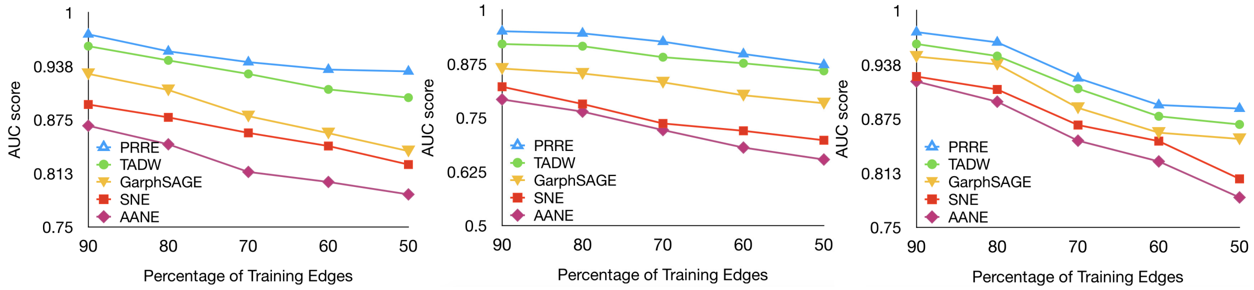
where $\#(i,j)$ is the number of occurrence in the graph context, $\#_i$ is the number of occurrence of node v_i and C is the collections of observed node pairs.

Figure 5 presents the performance of different similarity measures on link prediction. According to the experiment results, we have the following observations:

(1) The basic observation is that RPR and PPMI, Jaccard and AA have the similar results in four datasets. This is reasonable since RPR and PPMI are random walk based similarity measure that capture high order proximities, Jaccard and AA are neighborhood based similarity measures that capture first and second order proximities.

(2) In the blogcatalog dataset, the network is denser than in other datasets and the neighborhood based similarity achieves better performance. However, in other three datasets, the networks are sparse, capturing only first and second order proximities may lose information, and PPMI and RPR gain better performance than neighborhood based similarities.

[2] https://en.wikipedia.org/wiki/Receiver_operating_characteristic



(a) Citeseer (b) Cora (c) Pubmed

Figure 4: AUC score of link prediction on citeseer, cora and pubmed.

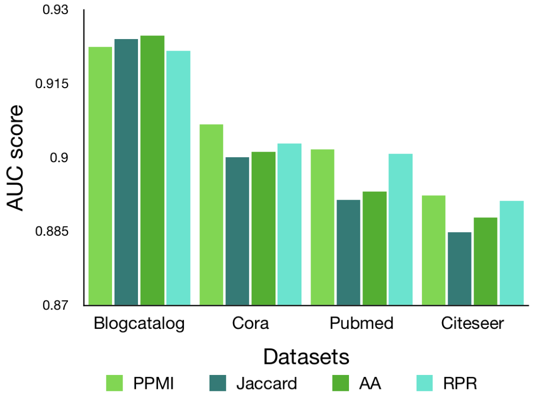
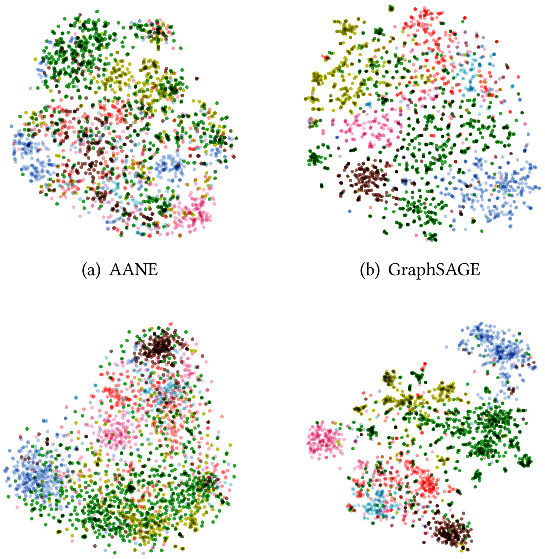


Figure 5: Comparisons of similarity measures in link prediction task on four datasets.



(a) AANE (b) GraphSAGE

(c) SNE (d) PRRE

Figure 6: Network visualization results on cora dataset. Nodes are mapped into the 2-D space using the t-SNE package with learned embeddings. Color indicates the class label. Best viewed on screen.

4.7 Network Visualization

Network visualization is a key application of network embedding which supports tasks such as data exploration and understanding. Following the experimental setting of existing works [5, 24, 29], we first learn a low dimensional representation for each node and then map those representations into the 2-D space with t-SNE [19]. Figure 6 illustrates the network visualization results on cora dataset. Each dot denotes a node in the network and each color represent denotes label of a class. A good embedding method is expected to make nodes with same label close to each other in the embedding space.

As observed in Fig 6, three baselines that use both attributes and network topology do not perform as well as PRRE and different colors of nodes are not separated well. The visualization results of PRRE are quite clear since most of nodes with same label (color) are close to each other and nodes with different labels(colors) are far from each other. This further verifies the effectiveness of the proposed PRRE method.

5 CONCLUSIONS

In this paper, we have proposed the PRRE model for embedding attributed networks. Based on the observation of partial correlation between topology and attributes, we adopt a personalized relation ranking framework to exploit the partial correlation. Experimental results of node classification, link prediction and network visualization on four real-world datasets demonstrate the superior performance of PRRE compared to several state-of-the-art network embedding methods.

This paper suggests several potential future directions of research. First, thresholding the similarity can reduce complexity cost but may also bring noise to the ranking, a potential solution is to develop a strategy to sample the ranking pairs. Another possible direction is to learn the confidence for topology and node attributes so that when one information source is not reliable, we are able to reduce its influence. Last, it would be interesting to explore a semi-supervised methods so that it can make use of labeled data.

6 ACKNOWLEDGEMENTS

This research is supported by Alibaba-Zhejiang University Joint Institute of Frontier Technologies and Zhejiang Provincial Key Research and Development Plan (No. 2017C01012), China Postdoctoral

Science Foundation (No. BX201700136), National Key Technology R&D Program of China (No. 2012BAI34B01, 2014BAK15B02).

REFERENCES

[1] Lada A Adamic and Eytan Adar. 2003. Friends and neighbors on the web. *Social networks* 25, 3 (2003), 211–230.

[2] Philippe Apparicio, Mohamed Abdelmajid, Mylène Riva, and Richard Shearmur. 2008. Comparing alternative approaches to measuring the geographical accessibility of urban health services: Distance types and aggregation-error issues. *International journal of health geographics* 7, 1 (2008), 7.

[3] Mukund Balasubramanian and Eric L Schwartz. 2002. The isomap algorithm and topological stability. *Science* 295, 5552 (2002), 7–7.

[4] Mikhail Belkin and Partha Niyogi. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*. 585–591.

[5] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*. https://openreview.net/forum?id=r1ZdKJ-0W

[6] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 891–900.

[7] Dániel Fogaras, Balázs Rácz, Károly Csalogány, and Tamás Sarlós. 2005. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics* 2, 3 (2005), 333–358.

[8] 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.

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1025–1035.

[10] 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.

[11] Caiyan Jia, Yafang Li, Matthew B Carson, Xiaoyang Wang, and Jian Yu. 2017. Node Attribute-enhanced Community Detection in Complex Networks. *Scientific Reports* 7, 1 (2017), 2626.

[12] Ling Jian, Jundong Li, Kai Shu, and Huan Liu. 2016. Multi-Label Informed Feature Selection.. In *IJCAI*. 1627–1633.

[13] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*. 2177–2185.

[14] 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.

[15] Jiongqian Liang, Peter Jacobs, and Srinivasan Parthasarathy. 2017. SEANO: semi-supervised embedding in attributed networks with outliers. *arXiv preprint arXiv:1703.08100* (2017).

[16] Jiongqian Liang, Peter Jacobs, Jiankai Sun, and Srinivasan Parthasarathy. 2017. Semi-supervised Embedding in Attributed Networks with Outliers. *arXiv preprint arXiv:1703.08100* (2017).

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

[18] Ben London and Lise Getoor. 2014. Collective Classification of Network Data. *Data Classification: Algorithms and Applications* 399 (2014).

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

[20] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3, 2 (2000), 127–163.

[21] 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.

[22] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of Jaccard coefficient for keywords similarity. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Vol. 1.

[23] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. 2016. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1105–1114.

[24] Shirui Pan, Jia Wu, Xingquan Zhu, Chengqi Zhang, and Yang Wang. 2016. Triparty deep network representation. *Network* 11, 9 (2016), 12.

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

[26] Joseph J Pfeiffer III, Sebastian Moreno, Timothy La Fond, Jennifer Neville, and Brian Gallagher. 2014. Attributed graph models: Modeling network structure with correlated attributes. In *Proceedings of the 23rd international conference on World wide web*. ACM, 831–842.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 452–461.

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

[29] 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*. International World Wide Web Conferences Steering Committee, 1067–1077.

[30] Lei Tang and Huan Liu. 2009. Relational learning via latent social dimensions. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 817–826.

[31] Cunchao Tu, Han Liu, Zhiyuan Liu, and Maosong Sun. 2017. Cane: Context-aware network embedding for relation modeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vol. 1. 1722–1731.

[32] Cunchao Tu, Weicheng Zhang, Zhiyuan Liu, and Maosong Sun. 2016. Max-Margin DeepWalk: Discriminative Learning of Network Representation.. In *IJCAI*. 3889–3895.

[33] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11, Dec (2010), 3371–3408.

[34] 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.

[35] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding.. In *AAAI*. 203–209.

[36] Xin Wang, Roger Donaldson, Christopher Nell, Peter Gorniak, Martin Ester, and Jiajun Bu. 2016. Recommending Groups to Users Using User-Group Engagement and Time-Dependent Matrix Factorization.. In *AAAI*. 1331–1337.

[37] Xin Wang, Steven C.H. Hoi, Martin Ester, Jiajun Bu, and Chun Chen. 2017. Learning Personalized Preference of Strong and Weak Ties for Social Recommendation. In *Proceedings of the 26th International Conference on World Wide Web (WWW '17)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, 1601–1610. https://doi.org/10.1145/3038912.3052556

[38] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 5–14.

[39] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network Representation Learning with Rich Text Information.. In *IJCAI*. 2111–2117.

[40] Jaewon Yang, Julian McAuley, and Jure Leskovec. 2013. Community detection in networks with node attributes. In *Data Mining (ICDM), 2013 IEEE 13th international conference on*. IEEE, 1151–1156.

[41] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861* (2016).

[42] Shuhan Yuan, Xintao Wu, and Yang Xiang. 2017. Sne: signed network embedding. In *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 183–195.

[43] Si Zhang and Hanghang Tong. 2016. Final: Fast attributed network alignment. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1345–1354.

[44] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. 2018. ANRL: Attributed Network Representation Learning via Deep Neural Networks.. In *IJCAI*. 3155–3161.