Network completion via deep metric learning

Qiang Wei 1,2 *

1 School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China.
2 National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu, 610041, China.
* Corresponding author’s e-mail: weiqiang@std.uestc.edu.cn

Abstract. Completing networks is often a necessary step when dealing with problems arising from applications in incomplete network data mining. This paper investigates the network completion problem with node attributes. We proposed a new method called DeepMetricNC by exploiting the correlation between node attributes and the underlying network structure. In DeepMetricNC, the correlation is modeled as a nonlinear mapping from node attributes to the probability of edge existence. To obtain the mapping, deep metric learning is applied with batch training and random negative sampling. DeepMetricNC has linear training time complexity and can adapt to large-scale network completion tasks. Experiments of real networks show that DeepMetricNC completes network structures better than other methods, and is more suitable when the portion of the observed part is small.

1. Introduction
Networks are commonly used to represent the patterns of connections between the components of complex systems [1]. Network structures play an important role in many network applications such as node classification, link prediction, etc. [2]. Complete knowledge of the structure is important to most network analysis applications. However, it is often difficult to gather the complete structure of complex networks in practice. Some networks can only be accessed through sampling or crawling (e.g., Internet). In other cases, network structure measurement is costly (e.g., biological networks) [3]. Given that the collected network data is often incomplete with missing nodes and edges, completing the unobserved part of the network is one of the key problems. This problem is known as network completion (NC) [2]. Several studies have been introduced to solve network completion on network topology itself [2, 7], and others use node attributes for network completion. In addition to the known partial observed structure, node attribute is often available, e.g. profile information in social networks [8] and content information in citation networks [9]. Node attribute can provide additional information for the underlying structure. How to make better use of node attribute is a problem that needs to be solved. Existing works exploit node attributes by the factorization of the similarity matrix which is a linear transformation [5,6]. However, the correlation between node attributes and network structure demonstrates non-linear characteristics [10], which leads to existing methods limited in solving real-world network completion problems.

In this paper, we propose a network completion method by exploiting node attributes via deep metric learning. We transform network completion into a deep metric learning problem, in detail, we model the correlation between node attributes and network structure as a nonlinear mapping with learnable
parameters, and use the observed part of networks to train the learned parameters, then the learned mapping is used to complete the unobserved part. Experimental results show that our method performs significantly better than the state-of-art methods.

2. Related work

2.1. Network completion
Network completion methods infer the rest of the network with missing nodes or edges based on a partial observed sample of the network. At present, there are two types of network completion methods, one is only concerned with the topology itself, and the other is to exploit node attribute information. Kim and Leskovec [2] treat the whole network structure as a Kronecker model, and use the observed part of the network to fit the model, they cast the problem in the Expectation Maximization framework and design a Gibbs sampling approach to estimate the model parameters as well as the inference for the unobserved network. Hanneke and Xing [7] analyse a general formalization of learning from random subsamples, and derive confidence bounds on the number of differences between the true and learned topologies. Tran et al. [4] infer the missing parts of a network based on a deep generative graph model which consecutively adds one node to maximize the probability in generation step. The above works only use the network topology itself, in order to further utilize the node attribute information, some works focus on the correlation between the attribute similarity matrix and the underlying structure. Forsati et al. [5] utilize the node attributes similarity to transduct the knowledge to the unobserved nodes by matrix completion/factorization on the observed adjacency matrix and the complete attribute similarity. Rafailidis and Crestani [6] further utilize the cluster based similarities of the node attributes, they design a joint objective function to factorize the observed adjacency matrix with the cluster based similarities. However, none of the above studies have fully explored the correlation between node attributes and network structure, either not using node attributes, or using them in a linear manner.

2.2. Deep metric learning
Metric learning aims to find an optimal distance similarity metric for specific learning tasks among samples [10]. In recent years, deep metric learning has attracted researchers’ attention in many different areas. Deep metric learning utilizes deep neural networks to obtain nonlinear embedding representation of samples instead of traditional linear methods such as convex programming, SVM, etc. Siamese networks [11] and Triplet networks [15] inspired most of the recent studies in the literature. Siamese Networks find an embedding space such that similar samples have small distances and vice versa. Several variants [12-14] has been developed based on Siamese Networks. Triplet networks inspired by Siamese network provide a higher discrimination power while using both in-class and inter-class relations [16], they commonly contain three objects, which are formed positive, negative, and anchor samples. Besides deep neural network structure, sample selection and loss function both play a very important role in deep metric learning (see [10] for further discussion). Although deep metric learning has been applied in many fields, such as face recognition, semantic textual similarity and speaker verification, it has not been applied in network completion.

3. Method
We begin by introducing the network completion problem of inferring unobserved part of networks with available node attributes. We then propose DeepMetricNC, an algorithm based on correlation of attribute space and network structure via deep metric learning.

3.1. Problem formulation
Let $G(V, E, X)$ be an undirected and unweighted network structure, where $V$ is the node set and $N=|V|$ is the total number of nodes, $E$ is the edge set, $X \in \mathbb{R}^{N \times d}$ is the attribute matrix, $V$ is composed of two disjoint set $V_o$ and $V_u$ corresponding the observed part and the unobserved part, that is $V=V_o \cup V_u$ and
\( V_o \cap V_u = \emptyset \) hold. We assume that the observed part \( G_o(V_o, E_o) \) is completed, that is all the edges in \( V_o \) is observed. In the unobserved part \( G_u = G - G_o \), edges in \( V_u \) and edges between \( V_o \) and \( V_u \) need to be completed. We further assume that \( X \) is not missing. Therefore, our problem is defined as follows: 

**Problem definition.** Given node attributes matrix \( X \) and partial observed network structure \( G_o \), our goal is to complete the unobserved part \( G_u \).

### 3.2. DeepMetricNC algorithm

We assume that the node attributes are related to the underlying network structure, furthermore, we assume that whether there exists an edge between two nodes or not is related to the attributes of the two nodes. This assumption is based on the fact that node attributes are related to the underlying structure [17].

Before diving into DeepMetricNC, we first discuss the shortcomings of the current methods in the use of node attributes. Exist methods [5, 6] achieve network completion by establishing the correlation between the similarity matrix \( S \) of node attributes and the underlying structure. These methods have three shortcomings: firstly, it is difficult to obtain a proper \( S \), the corresponding similarity matrix \( S \) is not unique for a given attributes matrix \( X \), and it is not easy to choose a good one for network completion; secondly, the linear assumption is made from \( S \) to \( G \), which is difficult to satisfy in real networks; finally, these methods suffer from high time complexity, thus cannot be used in large-scale networks.

To overcome the shortcomings of existing methods, we propose DeepMetricNC algorithm. In particular, DeepMetricNC directly establishes the relationship between \( X \) and \( G \) via deep metric learning. We treat each node as a sample, and if there exists an edge between two nodes, they are considered similar, otherwise, they are not similar. We then cast the network completion problem into deep metric learning framework. Node attributes and edges in \( G_o \) are used for training, then the learned mapping function from attributes to edges is used for completion in \( G_u \).

Given two input node attributes \((X_i, X_j)\), we aim to seek a nonnegative similarity metric function \( S_{ij} = f(g(X_i), g(X_j)) \) that represents the pairwise relationship between node \( i \) and \( j \). That is, we first learn an embedding \( g \) from node attributes \( X \) into an embedding subspace, then use distance function \( f \) determine whether node \( i, j \) is close or not. We implement \( g \) via a multilayer neural network with ReLU as an activation function, which is parameterized by a weight vector \( P \in \mathbb{R}^{d \times p} \), and we take \( f \) as the simple inner product function with sigmoid. Figure 1 illustrates the architecture of DeepMetricNC algorithm. We embed the original node attributes matrix \( X \) into a nonlinear space \( Y \), then a pair wise similarity matrix \( S \) is computed and normalized to \([0,1]\). Formally, we learn metric similarity matrix \( S \) as

\[
S_{ij} = f(X_i, X_j) = \sigma(\text{ReLU}(PX_i)\text{ReLU}(PX_j)^T)
\]

where \( \sigma(\cdot) \) is the logistic sigmoid function.

![DeepMetricNC architecture.](image)

We optimize the optimal weight vector \( P \) by minimizing the following cross-entropy loss function

\[
L = -\sum_{(i,j) \in E_o} \ln S_{ij} - \sum_{(i,j) \notin E_o} \ln(1-S_{ij})
\]
In the implementation, we adopt batch training and random negative sampling. Due to the sparse nature of the graph structure, there is only a small part of the edges between the node pairs. During the training process, we randomly select negative node pairs with number equal to $|E_o|$ in each epoch. Negative sampling is necessary for learning a good metric for node pairs with no edge between them. As shown in [18-20], more informative sample selection strategy is capable of increasing both the success of the parameter learning and the training speed, we leave this as our future work. The training time complexity of DeepMetricNC is $O(|E_o|p^2 + |V_o|dp)$ for each epoch [24], which is linear to network size. The linear training complexity makes DeepMetricNC suitable for large-scale network completion tasks.

4. Experiments

4.1. Datasets

To verify the effectiveness and benefit of the proposed DeepMetricNC on network completion tasks, we test it on three popular datasets for paper-citation networks, Citeseer, Cora and Pubmed [21]. The details of these datasets are introduced below.

**Cora.** The Cora dataset consists of 2708 scientific publications as node and 5429 cited relations as links. The citation network consists of 5429 links. Each node in the dataset has a 1433 dimension feature descriptor indicating the presence or absence of a word from a dictionary, and all the nodes are falling into seven classes.

**Citeseer.** The Citeseer dataset consists of 3327 nodes and 4732 edges. Each node has a 3703 dimension feature descriptor indicating the presence or absence of a word from a dictionary. The nodes are falling into six classes.

**Pubmed.** The Pubmed dataset consists of 19717 nodes and 44338 edges. Each node has a 500 dimension feature descriptor indicating a term frequency-inverse document frequency (TF-IDF) vector, and all the nodes are classified into three classes.

We only concern the inference of edges in our work, so we represent the node classes by a one-hot vector and concatenate them with node attributes.

4.2. Experimental setting

We artificially split a dataset into two parts, corresponding to $G_o$ and $G_u$. We start from a randomly selected node for $G_o$, and join the remaining nodes one by one in a breadth-first way until $|E_o|/|E|$ reaches a predetermined ratio. The reported results are averaged over 50 runs with different data splits. In our experiments, we set the number of MLP layers in DeepMetricNC to 1 without bias. The embedding size $d$ of $Y$ is set to 32 for all the three datasets. We train DeepMetricNC for a maximum of 2000 epochs using an ADAM algorithm [22] with learning rate 0.01, and stop training if the validation loss does not decrease for 200 consecutive epochs. All the network weights $P$ are initialized using Glorot initialization [23]. Code is made publicly available at https://github.com/weiqianglg/DeepMetricNC.

4.3. Baseline methods

We introduce two different classes of methods and compare them with DeepMetricNC to assess the quality of the proposed method for network completion problem. First, we consider an attribute-based approach MC-DT [5]. MC-DT decouples the completion from transduction to exploit the similarity information of node attributes. MC-DT consists of two decoupled stages: completion and transduction. In particular, it first completes the partially observed network by matrix completion methods, then the recovered network along with available similarity matrix from node attributes is utilized to transduce the correlation to fully unobserved nodes. In our settings, there are no missing edges in the observed network, so we just simply ignore the completion stage in MC-DT. The similarity matrix $S$ for MC-DT is obtained by cosine similarity in our experiments.
We also consider a direct attribute-based approach SVC [25]. SVC directly takes two node attributes as input, and predict the existence of corresponding edge. We adopt Linear SVC in our experiments.

4.4. Results and analyses
We evaluate each algorithm on its ability to infer the missing edges of the network. Since each method (DeepMetricNC, MC-DT and SVC) is able to compute the probability of each edge candidate, we quantify the performance of inference by the area under ROC curve (AUC) and the average precision (AP). The calculation of AUC and AP are balance between positive and negative edges, that is we randomly select \(|E-E_o|\) negative edges to estimate AUC and AP in \(G_e\).

Besides AUC and AP, we also adopt Mean Absolute Error (MAE) metric for both positive and negative unobserved edges, negative edges are also randomly sampled with number equal to \(|E-E_o|\).

Figure 2 and Figure 3 show the results of AUC and AP for the three real networks. DeepMetricNC gives a significant boost in performance compared to MC-DT and SVC, e.g., above 60% higher than MC-DT and 50% higher than SVC. Compared to MC-DT, DeepMetricNC directly exploits the attribute information of the nodes without making any assumptions about the underlying structure, while the MC-DT introduces errors when calculating the attribute similar matrix, furthermore, MC-DT cannot complete the calculation on Pubmed dataset due to the high time complexity. And Compared to SVC, on one hand, it cannot be calculated incrementally, which leads to bad fit for negative edges, on the other hand, it does not fully exploit the nonlinear relationship between attributes and structure. In addition, it is worth mention that DeepMetricNC obtains acceptable performance when the proportion of the observation network is small, e.g., above 70% for both AUC and AP when the observed portion is only 20%.

![Figure 2. AUC for the three real networks.](image1)

![Figure 3. AP for the three real networks.](image2)

We further evaluated the three algorithms in MAE for positive and negative edges, as shown in Figure 4 and Figure 5.
MC-DT has higher positive MAE but lower negative MAE, indicating that MC-DT tends to complete the unobserved structure to be empty (no edges). And DeepMetricNC learns positive edges function better than MC-DT and SVC, but there is still room for improvement in the learning of negative edges.

5. Conclusion

In this paper, DeepMetricNC is proposed for the network implementation problem with node attributes. Compared with MC-DT and SVC algorithms, DeepMetricNC directly models the correlation between node attributes and underlying network structures as a nonlinear function, thereby transforming the network implementation problem into a deep metric learning framework. And we combine batch training and random negative sampling in the implementation. Real networks experiments show that DeepMetricNC has better AUC and AP performance on real networks with more than 50% improvement, and it is also suitable to a smaller proportion of observed part and can adapt to large-scale networks. In the future, we will further improve the negative sampling strategy for better learning on negative edges.

References

[1] Newman, M. E. (2012). Communities, modules and large-scale structure in networks. Nature Physics, 8(1): 25-31.
[2] Kim, M., Leskovec, J. (2011). The network completion problem: Inferring missing nodes and edges in networks. In: SIAM International Conference on Data Mining, Mesa. pp. 47-58.
[3] Ahmed, N. K., Neville, J., Kompella, R. (2013). Network sampling: From static to streaming graphs. ACM Transactions on Knowledge Discovery from Data (TKDD), 8(2): 1-56.
[4] Tran, C., Shin, W. Y., Spitz, A., Gertz, M. (2019). DeepNC: Deep Generative Network Completion. arXiv preprint arXiv:1907.07381.
[5] Forsati, R., Barjaste, I., Ross, D., Esfahanian, A. H., Radha, H. (2016). Network completion by leveraging similarity of nodes. Social Network Analysis and Mining, 6(1): 102-124.
[6] Rafailidis, D., Crestani, F. (2016). Network completion via joint node clustering and similarity learning. In: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), San Francisco. pp. 63-68.

[7] Hanneke, S., & Xing, E. P. (2009). Network completion and survey sampling. In: Artificial Intelligence and Statistics, Florida. pp. 209-215.

[8] Verbeek, K., Suri, S. (2014). Metric embedding, hyperbolic space, and social networks. In: Thirtieth annual symposium on Computational geometry, Kyoto. pp. 501-510.

[9] Kipf, T. N., Welling, M. (2016). Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907.

[10] Kaya, M., Bilge, H. Ş. (2019). Deep metric learning: A survey. Symmetry, 11(9): 1066-1092.

[11] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R. (1994). Signature verification using a"siamese" time delay neural network. In: Advances in neural information processing systems, Denver. pp. 737-744.

[12] Chopra, S., Hadsell, R., LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego. pp. 539-546.

[13] Hadsell, R., Chopra, S., LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), New York. pp. 1735-1742.

[14] Mueller, J., Thyagarajan, A. (2016). Siamese recurrent architectures for learning sentence similarity. In: Thirtieth AAAI conference on artificial intelligence, Phoenix. pp. 2786-2792.

[15] Hoffèr, E., Ailon, N. (2015). Deep metric learning using triplet network. In: International Workshop on Similarity-Based Pattern Recognition, Copenhagen. pp. 84-92.

[16] Ge, W. (2018). Deep metric learning with hierarchical triplet loss. In: European Conference on Computer Vision (ECCV), Munich. pp. 269-285.

[17] Lin-Yuan, L. Ü. (2010). Link prediction on complex networks. Journal of University of Electronic Science and Technology of China, 39(5): 651-661.

[18] Wu, C. Y., Mannmatha, R., Smola, A. J., Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. In: IEEE International Conference on Computer Vision, Venice. pp. 2840-2848.

[19] Cui, Y., Zhou, F., Lin, Y., Belongie, S. (2016). Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In: IEEE conference on computer vision and pattern recognition, Venice. pp. 1153-1162.

[20] Movshovitz-Attias, Y., Toshev, A., Leung, T. K., Ioffe, S., Singh, S. (2017). No-fuss distance metric learning using proxies. In: IEEE International Conference on Computer Vision, Venice. pp. 360-368.

[21] Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., Eliassi-Rad, T. (2008). Collective classification in network data. AI magazine, 29(3): 93-93.

[22] Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[23] Glorot, X., Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In: Thirteenth international conference on artificial intelligence and statistics, Chia Laguna Resort. pp. 249-256.

[24] Mizutani, E., Dreyfus, S. E. (2001). On complexity analysis of supervised MLP-learning for algorithmic comparisons. In: International Joint Conference on Neural Networks. Proceedings, Washington DC. pp. 347-352.

[25] Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., Lin, C. J. (2008). LIBLINEAR: A library for large linear classification. Journal of machine learning research, 9(1): 1871-1874.