Predictive Network Representation Learning for Link Prediction

Zhitao Wang, Chengyao Chen and Wenjie Li
Department of Computing, The Hong Kong Polytechnic University, Hong Kong
{csztwang,cscchen,cswjli}@comp.polyu.edu.hk

ABSTRACT

In this paper, we propose a predictive network representation learning (PNRL) model to solve the structural link prediction problem. The proposed model defines two learning objectives, i.e., observed structure preservation and hidden link prediction. To integrate the two objectives in a unified model, we develop an effective sampling strategy to select certain edges in a given network as assumed hidden links and regard the rest network structure as observed when training the model. By jointly optimizing the two objectives, the model can not only enhance the predictive ability of node representations but also learn additional link prediction knowledge in the representation space. Experiments on four real-world datasets demonstrate the superiority of the proposed model over the other popular and state-of-the-art approaches.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Learning latent representations**;

KEYWORDS

Network Representation Learning; Link Prediction

ACM Reference format:
Zhitao Wang, Chengyao Chen and Wenjie Li. 2017. Predictive Network Representation Learning for Link Prediction. In *Proceedings of SIGIR '17, Shinjuku, Tokyo, Japan, August 07-11, 2017,* 4 pages.
DOI: http://dx.doi.org/10.1145/3077136.3080692

1 INTRODUCTION

Many social, physical and information systems in the world exist as networks. Predicting missing or promising candidate links on these networks is of great importance. For example, new friend recommendation by inferring potential friendships enhances user experience in on-line social media services and interaction discovery by identifying the latent links on protein-protein network saves large amount of human effort on blindly checking. Therefore, link prediction has drawn much attention from researchers [1, 7, 10]. Existing researches are often categorized as temporal link prediction which predicts potential new links on an evolving network, and structural link prediction which infers missing links on a static network. In this paper, we focus on *structural link prediction.* Given the partially observed structure of a network, the goal of it is to discover the unobserved hidden links.

How to represent network nodes with a set of carefully designed features is critical for link prediction. In the previous work, most node features are manually devised based on the graph topology. However, the hand-crafted features can only represent limited information of nodes yet require large amount of computation or manual effort. Recently, Network Representation Learning (NRL) models [2, 6, 9] are proposed to learn the latent representations of nodes, which can embed the rich structural information into the latent space. Most of these NRL models are learned in an unsupervised manner. They are hence more concerned with the descriptive ability of the representations. Such a learning paradigm limits their predictive ability on inferring hidden links due to the lack of the supervision on learning the prediction knowledge.

It is nontrivial to explore the learning model that can enhance the predictive ability of network representation that is specific for structural link prediction. To achieve this goal, we propose a predictive network representation learning model. The idea of this model is to learn node representations with two objectives. One is the ability to preserve the structural information of the observed structure. The other is the ability to discover the unobserved hidden links. The model is trained to learn both the predictive representations and the link prediction knowledge simultaneously. The two learning objectives influence each other interactively and progressively. The learned link prediction knowledge gives feedback to representation learning such that the learned representations tend to be more predictive. Meanwhile, the prediction knowledge is updated to fit the learned representations more. In this sense, the proposed model is a representation-based link prediction model. To optimize these two objectives jointly, we develop an effective algorithm with a well-designed hidden link sampling strategy to automatically remove certain observed links such that the assumed hidden links can be introduced into the learning process. The experimental results on various real-world datasets demonstrate that our model is more predictive than the state-of-the-art NRL models and other popular approaches.

2 METHOD

Formally, we denote a partially observed network as $G = (V, E)$, where $V = \{v_i\}_{i=1}^{|V|}$ is the node set and $E = E^+ \cup E^- \cup E^?$ represents the node-pair set with different link status in the network. Note that we focus on undirected networks in this paper. For each node-pair $(u, v) \in E^+$, the link status is *known present*; For each node-pair $(u, v) \in E^-$, the link status is *known absent*; while for $E^?$, the link status is *unknown.* Given a such network G, the goal of structural link prediction is to infer link status of node-pairs in $E^?$. Network representation learning aims at embedding each node $v_i \in V$ into a low-dimensional representation $\mathbf{x}_i \in \mathbb{R}^n$, which can be treated as features for prediction task.

SIGIR '17, Shinjuku, Tokyo, Japan
© 2017 ACM. 978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080692

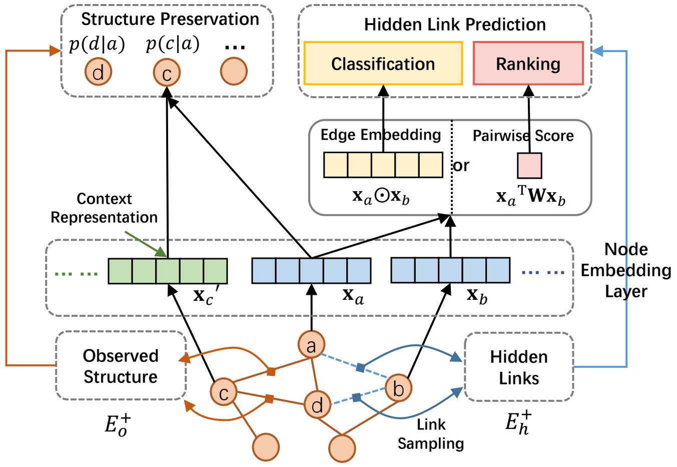

Figure 1: PNRL Model Overview

To solve structural link prediction problem, we propose the Predictive Network Representation Learning (PNRL) model. The main idea can be illustrated as Fig.1. The input network is firstly sampled as two components, i.e., observed structure (E_o^+) and hidden link set (E_h^+). The PNRL model defines a structural information preservation objective for the observed component and a representation-based hidden link prediction objective for the hidden component. By jointly optimizing the two objectives with shared embedding layer in the unified model, the learned node representations tend to be more predictive for link prediction. Additionally, this representation learning model can be directly used for link prediction based on the learned representations and prediction "knowledge" (classification hyperplane or ranking weight matrix).

2.1 Network Representation Learning for Observed Structure Preservation

The learning objective of observed structure is to preserve structural proximities among nodes in latent space. The neighbor set of a node has proven an important index for inferring pair-wise structural proximity. Therefore, node representations should be constrained within their neighbors. This is similar to the Skip-Gram word embedding model [6], where word representations can predict frequent surrounding words (context). In our scenario, neighbors of a node can be treated as frequent "context" and the representation learning objective is to maximum the likelihoods that each node generates its neighbors for observed edges in E_o^+. Therefore, each node v_i should have two representations, i.e., node representation $\mathbf{x}_i$ when it is target node and context representation $\mathbf{x}_i'$ when it is treated as "context" for other nodes.

Based on the above idea, we define the network representation learning objective for E_o^+ as minimizing the following negative-log loss function:

$$\mathcal{L}_o = -\sum_{(i,j)\in E_o^+} \log p(v_j|v_i) = -\sum_{(i,j)\in E_o^+} \log \frac{\exp(\mathbf{x}_j'^T \mathbf{x}_i)}{\sum_{h=1}^{|V|} \exp(\mathbf{x}_h'^T \mathbf{x}_i)} \tag{1}$$

where $p(v_j|v_i)$ represents the probability of "context" v_j generated by node v_i for $(i,j) \in E_o^+$. For the sake of improving computation efficiency, we apply a approximation method called Negative Sampling (NEG)[6], then the above loss function can be formulated as follow:

$$\mathcal{L}_o = -\sum_{(i,j)\in E_o^+} (\log \sigma(\mathbf{x}_j'^T \cdot \mathbf{x}_i) + \sum_k^K \mathbb{E}_{v_k \sim P_n(v_i)}[\log \sigma(-\mathbf{x}_k'^T \cdot \mathbf{x}_i)]) \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function, K is the number of negative samples and $P_n(v_i)$ is the noise distribution. For each $(i,j) \in E_o^+$, K negative samples are randomly drawn from $P_n(v_i) \sim (d_{v_i})^{3/4}$, where d_{v_i} is the degree of v_i in G.

2.2 Network Representation Learning for Hidden Link Prediction

The objective of this component is to improve predictive ability of node representations for link prediction and directly learn the link prediction model (either classification or ranking model) based on such representations.

Hidden Link Classification. Hidden link prediction can be regarded as a binary classification problem, i.e., classifying linked node-pairs in E_h^+ and unlinked node-pairs in E^- based on feature set of node-pair. In our model, the latent representation space naturally forms the feature space and we can derive features (embeddings) of node-pairs by using composition approach: $\mathbf{d}_{mn} = f(\mathbf{x}_m, \mathbf{x}_n)$. Following edge feature learning approaches used in previous work [2], we define the composition function as $f(\mathbf{x}_m, \mathbf{x}_n) = \mathbf{x}_m \odot \mathbf{x}_n$, where $\odot$ is Hadamard product. To formulate the hidden link classification objective, we introduce a classical max-margin classifier L2-SVM, which minimizes the following loss function:

$$\mathcal{L}_h^c = C \sum_{(m,n)\in E_h^+ \cup E^-} \max(1 - y_{mn} \boldsymbol{\alpha}^T \mathbf{d}_{mn}, 0)^2 + \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{\alpha} \tag{3}$$

where C is the regularization parameter and $\boldsymbol{\alpha}$ is the hyperplane that the classifier seeks for. y_{mn} is the link status label of node-pair (m,n), if $(m,n) \in E_h^+$, $y_{mn} = 1$, otherwise $y_{mn} = 0$. Network representation learning on this objective will not only make representations tend to be discriminative but also learn the "knowledge" $\boldsymbol{\alpha}$ for link classification in the representation space.

Hidden Link Ranking. Link prediction can also be considered as a ranking problem. The idea is that node-pairs with hidden links should have higher rank than those without hidden links. From per-node view, the ranking score of node v_n who has hidden link with node v_m should be greater than v_l who has no hidden link with v_m. The above idea can be formulated as following constraint:

$$r_{m,n} > r_{m,l}, \quad if\ (m,n) \in E_h^+\ and\ (m,l) \in E^-$$

We employ the bilinear product on node representations to derive the ranking score of node-pairs as $r_{m,n} = \mathbf{x}_m^T \mathbf{W} \mathbf{x}_n$, where $\mathbf{W}$ is a weight matrix that will be learned. For the sake of defining the ranking objective based on the above constraint, we apply the classical squared hinge loss function as follow:

$$\mathcal{L}_h^r = \sum_{(m,n)\in E_h^+, (m,l)\in E^-} \max(1 - \mathbf{x}_m^T \mathbf{W}(\mathbf{x}_n - \mathbf{x}_l), 0)^2 + \frac{\lambda_{\mathbf{W}}}{2} \|\mathbf{W}\|_2^2 \tag{4}$$

where $\lambda_{\mathbf{W}}$ is a regularization parameter. In this learning objective, node representations tend to fit the ranking criterion and additional link ranking "knowledge" $\mathbf{W}$ will be learned.

Algorithm 1: PNRL Model Learning Algorithm

```
Input  : Training network $G = (V, E_t)$, $E_t = E^+ \cup E^-$;
Output : Node embedding matrix $\mathbf{X}$, context embedding
         matrix $\mathbf{X}'$, hyperplane $\boldsymbol{\alpha}$ or weight matrix $\mathbf{W}$
1  Initialize $\mathbf{X}, \mathbf{X}', \boldsymbol{\alpha}$ or $\mathbf{W}$ (uniformly random), $E_r^+ \leftarrow E^+$;
2  while $E_r^+ \neq \emptyset$ do
3      $E_h^+ \leftarrow$ randomly sample $\beta|E^+|$ edges from $E_r^+$;
4      $E_h^- \leftarrow$ randomly sample $\beta|E^+|$ edges from $E^-$;
5      $E_r^+ \leftarrow E_r^+ - E_h^+$, $E_o^+ \leftarrow E^+ - E_h^+$;
6      repeat
7          for each node $v_i$ do
8              Optimization on $\mathcal{L}_o$:
9              Randomly sample $v_j$ that $(i, j) \in E_o^+$;
10             Randomly draw $K$ negative nodes from $P_n(v_i)$;
               $\mathbf{x}_i \leftarrow \mathbf{x}_i - e_1\eta\frac{\partial\mathcal{L}_o}{\partial\mathbf{x}_i}$, $\mathbf{x}'_j \leftarrow \mathbf{x}'_j - e_1\eta\frac{\partial\mathcal{L}_o}{\partial\mathbf{x}'_j}$;
11             for each $v_k$ in $K$ negative samples do
12                 $\mathbf{x}'_k \leftarrow \mathbf{x}'_k - e_1\eta\frac{\partial\mathcal{L}_o}{\partial\mathbf{x}'_k}$ ;
13             Optimization on $\mathcal{L}_h$:
14             if $\mathcal{L}_h = \mathcal{L}_h^c$ (PNRL-C) then
15                 Sample $v_l$ that $(i, l) \in E_h^+ \cup E_h^-$ ;
16                 $\mathbf{x}_i \leftarrow \mathbf{x}_i - e_2\frac{\partial\mathcal{L}_h^c}{\partial\mathbf{x}_i}$, $\mathbf{x}_l \leftarrow \mathbf{x}_l - e_2\frac{\partial\mathcal{L}_h^c}{\partial\mathbf{x}_l}$,
                   $\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} - e_2\frac{\partial\mathcal{L}_h^c}{\partial\boldsymbol{\alpha}}$ ;
17             if $\mathcal{L}_h = \mathcal{L}_h^r$ (PNRL-R) then
18                 Sample $v_m, v_n$ that $(i, m) \in E_h^+$ and $(i, n) \in E^-$;
19                 $\mathbf{x}_i \leftarrow \mathbf{x}_i - e_2\frac{\partial\mathcal{L}_h^r}{\partial\mathbf{x}_i}$, $\mathbf{x}_m \leftarrow \mathbf{x}_m - e_2\frac{\partial\mathcal{L}_h^r}{\partial\mathbf{x}_m}$,
                   $\mathbf{x}_n \leftarrow \mathbf{x}_n - e_2\frac{\partial\mathcal{L}_h^r}{\partial\mathbf{x}_n}$, $\mathbf{W} \leftarrow \mathbf{W} - e_2\frac{\partial\mathcal{L}_h^c}{\partial\mathbf{W}}$ ;
20     until Convergence;
```

2.3 Unified Model and Joint Learning

By incorporating the above learning objectives into the unified framework (Fig.1), we can derive our Predictive Network Representation Learning model with the following optimization objective:

$$\min \mathcal{L} = \min(\eta\mathcal{L}_o + \mathcal{L}_h) \tag{5}$$

where η is a weight parameter for balancing the importance of the two learning objectives. $\mathcal{L}_h$ can be either $\mathcal{L}_h^c$ and $\mathcal{L}_h^r$, hence our model has two variants: **PNRL-C** (link classification) and **PNRL-R** (link ranking). In order to optimize the above problem efficiently, we propose an algorithm framework represented as Algorithm 1.

- Hidden Link Sampling (line 2-5). Before parameters learning, the algorithm firstly samples the hidden link set E_h^+ (with the ratio β) and the observed set E_o^+ from input graph. This sampling process will be repeated until all edges have been trained as hidden links in the whole learning process. Additionally, we employ the under-sampling strategy in this process for PRNL-C to handle the inevitable imbalance problem resulted by $|E^-| \gg |E_h^+|$. The under-sampling draws a subset $E_h^- \subseteq E^-$ such that $|E_h^-| = |E_h^+|$ (line 4). As for PRNL-R, it can naturally overcome imbalance due to the property of pair-wise ranking loss.
- Parameter Learning (line 6-20). We adopt the classical stochastic gradient decent (SGD) method for parameter learning and design a per-node learning strategy to guarantee a balanced update on node representations. In a single learning iteration, we update representations and corresponding parameters for each node by optimizing the two objectives sequentially. Representations are firstly updated on $\mathcal{L}_o$ with learning rate e_1, where a linked pair and k negative pairs of current node v_i are sampled from E_o^+. Then updating is conducted to optimize $\mathcal{L}_h$ with learning rate e_2. In this step, a node-pair of v_i is sampled from $E_h^+ \cup E_h^-$ for PNRL-C, while for PNRL-R two node-pairs of v_i are drawn from E_h^+ and E^- separately. If the convergence condition is satisfied or a max number of learning iteration is reached for current E_o^+ and E_h^+, the algorithm will stop current learning process and conduct a new hidden link sampling.

3 EXPERIMENTS

3.1 Experimental Setup

Data. We evaluate the performance of proposed methods on four different types of datasets, all of which are widely used for link prediction evaluation. The statistic of these datasets are summarized in Table.1. **Facebook** [4] is a sample of online social network, where nodes represent users and edges represent friendships. **Email** [3] is an email communication network. Nodes in this network are email addresses and each edge represents that at least one email was sent. **PowerGrid** [10] is a sampled infrastructure network of high-voltage power grid, where nodes are power stations and links are high-voltage transmission lines. **Condmat** [7] is a scientific collaboration network. Authors of papers on condensed matter physics are treated as nodes and edges represent their co-author relationships.

Table 1: The Statistics of Experimental Data

Dataset	Nodes	Edges	Avg. Degree
Facebook	4,024	87,887	43.68
Email	1,133	5,451	9.62
PowerGrid	4,941	6,594	2.67
Condmat	16,264	47,594	5.85

Baselines. We evaluate our methods against both popular link prediction algorithms and state-of-the-art network representation learning methods. As for link prediction algorithms, we mainly consider popular heuristic methods with three handcraft indexes, i.e., Adamic-Adar Score (**AA**) [1], Jaccard's Coefficient (**JC**) and Preferential Attachment (**PA**). Besides, we compare performance with a matrix factorization based algorithm (**MF**) [5], which also learns node latent representations. All these algorithms can achieve state-of-the-art performance in link prediction. In terms of network representation learning models, three state-of-the-art models are evaluated in the experiments. Both **Deepwalk** [8] and **node2vec** [2] employ random walk to sample structural information and node representations are learned to preserve pairwise similarities of close nodes on these walks. **Line** [9] aims at preserving first-order and second-order proximities in concatenated embeddings. These models were reported to have better performance than above heuristic link prediction methods [2].

Table 2: AUC Scores for Link Prediction

Methods	Facebook	Email	PowerGrid	CondMat
JC	0.925 ± 0.001	0.821 ± 0.008	0.600 ± 0.005	0.904 ± 0.007
AA	0.927 ± 0.001	0.827 ± 0.007	0.600 ± 0.006	0.903 ± 0.006
PA	0.733 ± 0.002	0.662 ± 0.019	0.555 ± 0.009	0.817 ± 0.006
MF	0.967 ± 0.009	0.814 ± 0.028	0.627 ± 0.027	0.936 ± 0.017
Deepwalk	0.953 ± 0.004	0.719 ± 0.036	0.608 ± 0.030	0.934 ± 0.005
Line	0.928 ± 0.006	0.721 ± 0.042	0.603 ± 0.026	0.917 ± 0.004
Node2Vec	0.951 ± 0.003	0.742 ± 0.046	0.628 ± 0.017	0.943 ± 0.008
PNRL-C	**0.994 ± 0.001****	0.870 ± 0.025*	0.664 ± 0.014**	0.966 ± 0.004**
PNRL-R	0.990 ± 0.001**	**0.882 ± 0.010*****	**0.668 ± 0.009****	**0.968 ± 0.002****

(*, **, *** indicate significantly better than best baseline at 0.01, 0.001, 0.0001 level in paired t-test.)

Setting. Given above four networks, we draw a fixed proportion of existed edges for training all models, and treat the rest of edges as test set. We set a fixed training split ratios for all datasets, i.e., 80% for training and 20% for testing. The prediction quality is measured by a standard metric, the area under the receiver operating characteristic curve (AUC). The calculation of AUC requires prediction scores (probabilities) of each node-pairs in test set. Our models and link prediction baselines can output prediction scores directly. As for network representation learning baselines, we extract edge features by utilizing the same composition approach on learned representations and train a L2-SVM classifier with under-sampling to get prediction scores.

To guarantee fairness, the dimension size for all latent representation learning models is set as 64. Other relevant parameter settings are also kept similar. For DeepWalk and node2vec, window size is 5, walk length is 40 and walks per node is 10; the number of NEG is set as 5 for both Line and our models; regularization parameter in PNRL-C is $C = 1$ as that in the L2-SVM trained for other representation learning methods. In terms of specific parameters in our models, we fix $\eta = 1$, $\lambda_{\mathbf{W}} = 0.01$, set learning rates as $e_1 = 0.001$, $e_2 = 0.001$ (with linearly decrease), sampling ratio for hidden link creation as $\beta = 0.2$ in all experiments.

3.2 Evaluation Results

As reported in Table 2, we summarize the average evaluation results after 5-cross-validation for each method on each dataset. Besides, we conduct paired t-test to verify the significance of result difference between our models and best performing baselines. The prediction performance is affected by both data sparsity and network type.

Averagely, network representation learning (NRL) baselines outperform heuristic link prediction baselines on most datasets. However, NRL baselines only have similar or even worse performance compared with MF. Besides, in spite of employing under-sampling, NRL baselines only achieve second worst performance on Email dataset and gain similar AUC scores with heuristic methods on PowerGrid dataset. This demonstrates that the predictive ability of NRL baselines is still limited.

Compared with NRL baselines, the proposed PNRL models succeed to improve the predictive ability of network representations with 2.7%-18.9% higher performance. In particular, PNRL-C achieves 5.7% and 17.3% improvements over the best performing NRL baseline on PowerGird and Email respectively, and PNRL-R gains 6.4% and 18.9% higher AUC scores on these two datasets. Meanwhile, our methods are superior to the best heuristic baseline and MF method with 7.1%-11.3% and 2.8%-9.6% improvements respectively. Overall, the proposed PNRL models significantly outperform all the baselines on the four datasets.

Additionally, PNRL-R performs better than PNRL-C on Email, PowerGrid and CondMat datasets. It indicates that the ranking objective better overcomes imbalance than classification objective with under-sampling on sparse networks.

4 CONCLUSION

In this paper, we propose the predictive network representation learning model for solving structural link prediction problem. This model defines two objectives for representation learning, i.e., observed structure preservation and hidden link prediction. By jointly optimizing the two objectives with shared node embeddings, the learned representations are more predictive and additional link prediction knowledge are learned. Experiments on the different types of classical datasets witness the superior performance of the proposed methods over the other state-of-the art approaches. In the future work, the prior knowledge of link formation (e.g., social theory) is expected to be incorporated in sampling strategy, which will give better supervision for predictive representation learning.

ACKNOWLEDGMENTS

The work in this paper was supported by Research Grants Council of Hong Kong (PolyU 5202/12E, PolyU 152094/14E), National Natural Science Foundation of China (61272291) and The Hong Kong Polytechnic University (4-BCB5, G-YBJP).

REFERENCES

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

[2] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD'16*. ACM, 855–864.

[3] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt, and Alex Arenas. 2003. Self-similar community structure in a network of human interactions. *Physical review E* 68, 6 (2003), 065103.

[4] Jure Leskovec and Julian J Mcauley. 2012. Learning to discover social circles in ego networks. In *NIPS'12*. 539–547.

[5] Aditya Krishna Menon and Charles Elkan. 2011. Link prediction via matrix factorization. In *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 437–452.

[6] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS'13*. 3111–3119.

[7] Mark EJ Newman. 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences* 98, 2 (2001), 404–409.

[8] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD'14*. ACM, 701–710.

[9] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW'15*. ACM.

[10] Duncan J. Watts and Steven H. Strogatz. 1998. Collective dynamics of "small-world" networks. *nature* 393, 6684 (1998), 440–442.