



Link Prediction Based on Smooth Evolution of Network Embedding

Hao Dong, Yue Kou^(✉), Derong Shen, and Tiezheng Nie

Northeastern University, Shenyang 110004, China
2431752775@qq.com, {kouyue, shenderong,
nietiezheng}@cse.neu.edu.cn

Abstract. The problem of link prediction in dynamic heterogeneous information networks has been widely studied in recent years. The technique of network embedding has been proved extremely useful for link prediction. However, the existing methods lack the close combination between deep-level features and temporal features of networks, which affects the accuracy of prediction and makes it difficult to adapt to the dynamic networks. In this paper, a Smooth Evolution model for Network Embedding (called SENE) is proposed, which considers both deep-level features and temporal features to obtain the embedded representations of the network structure, and uses the transformer mechanism to effectively obtain the smooth evolution of network embedding. Also an SENE-based link prediction algorithm is proposed, which can effectively guarantee the accuracy of link prediction. The feasibility and effectiveness of the proposed key technologies are verified by experiments.

Keywords: Link prediction · Network embedding · Smooth evolution · Transformer

1 Introduction

With the increasing popularity of the Internet, a large number of users use the Internet every day to understand the changing development and entertainment of the world. While using the Internet, users also leave a lot of data on it. And these different types of data can be combined into different networks. For example: the information from WeChat can be abstracted into a social network, treating each user as a node, and the friend relationship between the users as an edge. Compared to homogeneous networks, heterogeneous networks contain more information, so data mining for heterogeneous networks is more valuable. But, due to the inherent complexity of the heterogeneous network, how to better integrate the features contained in the heterogeneous network is still a challenge.

At the same time, link prediction [1], has attracted increasing attention in the research community, due to its importance in many real-world application. Link prediction is to analyze the known network topology and construct a prediction method to find the probability of edges between node pairs that do not yet have formed in the network. It can be divided into static link prediction and dynamic link prediction according to the time when the edge appears. Static link prediction is to predict the

topology of an unobserved network based on the observed partial network topology of the current network. Dynamic link prediction [2] is to predict the network structure at the next moment based on the observed changes in the network topology. Because the real world networks such as e-commerce networks, social networks, and user travel networks often change dynamically, dynamic link prediction has more practical value.

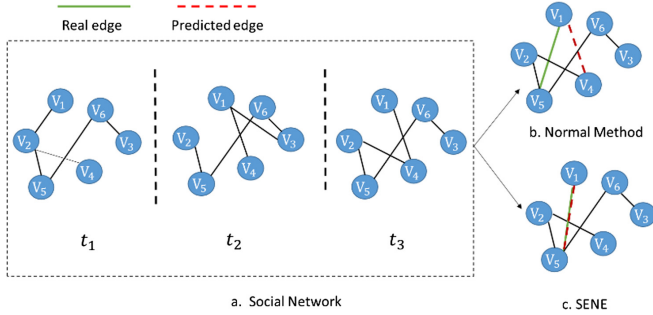


Fig. 1. An example of link prediction in social network

Let us consider the following motivating scenarios.

Scenario 1: Yesterday, employee e_1 sent an email to employee e_2 , but today employee e_1 sent an email to employee e_3 . This is, the real social networks are not static and will change over time. Generally, the current interaction information of employees is more dependent on recent behavior. Therefore, we need find a way to capture the temporal feature in the network.

Scenario 2: Figure 1(a) depicts a social network which includes a set of users (v_1-v_6). Each node represents a person and the edge between nodes represents the interaction information between people. t_1, t_2, t_3 represents the topological structure of the network in three adjacent time slices respectively. Next, the connection of v_1 in the next time slice will be predicted.

From **Scenario 2**, we can see the general link prediction method tend to predict between the edges that formed in previous time slice. As shown in Fig. 1 (b), it is incorrectly predicted that the edge between v_1 and v_4 is more likely to form at the next moment. Considering the network evolution information, we can predict the edge that never formed before. As shown in Fig. 1 (c), after considering the evolution information, the connection of v_1 in the next time slice can be correctly predicted.

The current link prediction methods for heterogeneous information networks [3] face some challenges: First, how to consider the inherent features and temporal features of the network itself when embedding the network? Secondly, how to effectively capture the evolution information embedded in the networks?

For the above problems, we propose a link prediction method based on smooth evolution of network embedding. The main contributions are as follows:

- (1) A Smooth Evolution model for Network Embedding (called SENE) is proposed. Different from traditional network embedding models, SENE not only considers the topology feature and context feature of networks, but also makes full use of the temporal feature to capture the smooth evolution of network embedding.
- (2) A SENE-based link prediction algorithm is proposed, which uses the smooth evolution information of network embedding to predict the network embedding at the next moment, further to predict the network structure at the next moment. It can effectively guarantee the accuracy of link prediction.
- (3) The feasibility and effectiveness of the key technologies proposed in this paper are verified through experiments.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 presents SENE model. Section 4 proposes a SENE-based link prediction algorithm. Section 5 shows the experimental results and Sect. 6 concludes.

2 Related Work

Various approaches for link prediction have been studied over the years. First, we briefly review the techniques for them. Then we analyse how our work differs from them. Current link prediction methods can be divided into network topology-based link prediction and network embedding-based link prediction [4].

Topology-based link prediction [5] starts from the topology of the network and calculates the similarity between the nodes in the network. Generally based on common neighbors, Jaccard, path similarity, etc. For example, Mitzenmache et al. [6] proposed a domain-based link prediction method. The essence is that two nodes have more common neighbors, and it is more likely that there is an edge. Brin et al. [7] proposed a method based on path similarity, using random walk to obtain the path of the node, and then calculating the path similarity. Nodes with high similarity are more likely to have edge connections.

Network embedding-based link prediction [8] mainly uses a low-dimensional vector to represent the nodes, and then calculates the similarity between the node representations. For example, Perozzi et al. [9] proposed a network embedding algorithm Deepwalk, which introduces deep learn into network representation learning, uses random walk to obtain sequence information, and then uses the Skip-gram model for network embedding. Grover et al. [10] proposed the node2vec algorithm to further develop the Deepwalk algorithm, and reached a good balance between the depth and breadth of random walk. Subsequently, Dong et al. [11] proposed a network embedding algorithm *matepath2vec* for heterogeneous information networks, using random walk based on a given element path to capture rich context information in heterogeneous information networks. Tang et al. [12] proposed another embedding algorithm LINE, and optimized it with negative sampling, and achieved a good balance between the accuracy and time complexity of the algorithm. Chen et al. [13] proposed PME, a link prediction algorithm for heterogeneous information networks, to decompose a heterogeneous information network into multiple networks, and then embedding each network separately to complete the link prediction task.

The differences between our work and existing work are as follows:

First, the link prediction method proposed in this paper differs from the above-mentioned techniques in that the traditional link prediction technology mainly acts on the prediction on homogeneous information networks, although some papers have studied the link prediction technology on heterogeneous information networks. But usually only consider the characteristics of the network itself, and ignore the timing information.

Second, this paper propose a SENE-based Link Prediction Algorithm. When embedding nodes, it also considers the deep-level features and temporal features of the network, and finally obtains the evolution information of network embedding by using the transformer mechanism.

3 SENE Model

3.1 Problem Definition

This section first introduces several related concepts, and then formalizes the definition of link prediction for dynamic heterogeneous information networks.

Definition 1 (heterogeneous information network): A heterogeneous information network is an information network with multiple types of objects and/or multiple types of links, formally defined as $G = (V; E; T; R)$. Among them, V is the union of different types of vertices, E is the union of different types of edges, T denotes the node type set, and R denotes the edge type set.

Given a heterogeneous information network $G = (V; E; T; R)$, where $T = \{t_1, t_2, \dots, t_n\}$, representing n consecutive time intervals. At the same time, $G_t = (V_t, E_t, T, R)$ is used to represent the network topology at time t , where V_t is a subset of V , representing the set of nodes at time t , and E_t is a subset of E at time t , representing the set of edges at time t . At the same time, We assumed that the total number of nodes at each moment does not change, so G_t can represented by $G_t = (V_t, E_t, T, R)$. In total, the heterogeneous information network G can represented by $G = G_1 \cup G_2 \cup \dots \cup G_n$ or $G = \{G_t \mid t = 1, 2, \dots, n\}$. Based on this we give the definition of dynamic link prediction.

Definition 2 (Dynamic Link Prediction): Given the network topology at time slice $0 - T$, predicting the network connection at time $T + 1$, that is, predicting the network connection at the next time based on network history information as follows:

$$G = \{G \mid G_1, G_2, \dots, G_t\} \rightarrow G_{t+1}$$

3.2 Model Overview

In order to better capture the deep-level and temporal features of heterogeneous information networks, we proposes Smooth Evolution model of Network Embedding—SENE. This section first introduces how to consider the deep-level and temporal feature when embedding the network, and then introduces the method of using the transformer mechanism to obtain network evolution information.

We build the SENE model based on the heterogeneous network G . The model can be divided into two phases: network embedding phase and smooth evolution phase (see Fig. 2).

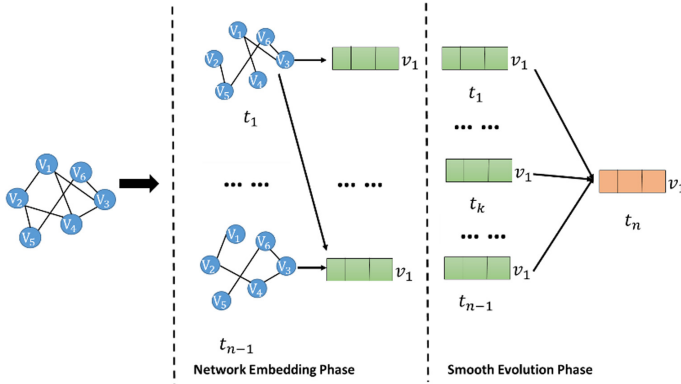


Fig. 2. Overview of SENE

Network Embedding Phase: The network is divided into sub-networks of different time slices at certain time intervals, and then each sub-network is embedded separately. When embedding, not only the topology information and context information of the node itself, but also the historical behavior information of the node must be considered to facilitate the capture of the evolution of network embedding.

Smooth Evolution Phase: In the previous network embedding stage, the network embedding that time slices from 0 to $n-1$ can already be obtained. By exploring the underlying rules in the evolution information of the network embedding from 0 to $n-1$ time slices, the network embedding at N th time slice can be obtained. Then, we use the transformer mechanism to capture network evolution information.

3.3 The First Phase: Network Embedding

In order to get a good embedding, we use Tang’s LINE method to embedding the network, and make some modifications to the LINE algorithm to make it more suitable for our model. The LINE method is show as follows:

$$p_1(v_i, v_j) = \frac{1}{1 + \exp(-u_i \cdot u_j)} \tag{1}$$

$$O = - \sum_{(i,j) \in E} \log p_1(v_i, v_j) \tag{2}$$

Where u_i is low dimensional vector representation of node v_i , $p_1(\dots)$ is a distribution in the vector space of $V * V$, O is the objective function.

Since the LINE does not distinguish the different of links, LINE is only applicable to homogeneous information networks. Therefore, we decompose a heterogeneous information network into multiple homogeneous information networks. At the same time, in order to ensure the consistency of embedding, we add the objective function of each homogeneous network to obtain the learning objective function O as follows:

$$O = \sum_{r \in R} \sum_{(i,j) \in E} \sum_{(i,k) \notin E} [s(v_i, v_j) - s(v_i, v_k)] \quad (3)$$

Where $s(\cdot, \cdot)$ represents similarity of node pairs, in this paper we use Euclidean distance to measure the similarity of node pairs.

In addition, in order to distinguish between different links, we assign different weights w to different type of links when embedding nodes, and use w_r represents the weight of edge type r . The attention [14] mechanism can achieve a corresponding weight for each different links type. Therefore, we use the attention mechanism for weight learning. The learning objective function after adding weight is shown in (4):

$$O = \sum_{r \in R} w_r \sum_{(i,j) \in E} \sum_{(i,k) \notin E} [s(v_i, v_j) - s(v_i, v_k)] \quad (4)$$

The calculation of the minimum value of Eq. (4) takes a huge time cost, and the number of non-linked pairs is extremely large compared to the linked pairs. Therefore, we adopt a negative sampling strategy to optimize the model. The general negative sampling method is to sample non-links pairs with the same probability, and does not take into account temporal feature, so we proposes a negative sampling strategy based on temporal feature.

If the i and k nodes are connected in the previous time slice, but not connected in the current time slice, the possibility of the i and k nodes connecting at the next moment is still very large, so the embedding between the nodes should be relatively similar. Based on the above analysis, we should sample nodes that have never been connected or nodes that have not been connected for a long time before.

At the same time, in order to simplify the problem, we only consider the distance between the time slice last connected between two nodes and the current time slice as follows:

$$P(j) = \frac{t - t_{ij}}{\sum_{k \notin E_t} (t - t_{ik})} \quad (5)$$

Where $P(j)$ represents the probability that node j is sampled, and t_{ij} represents the last time slice connected between node i and node j , E_t represents the union of edge in time slice t .

At the same time, in order to ensure that the vectors embedded in different time slices belong to the same vector space, we will embed the nodes of the $t-1$ time slice as the initial value of the t time slice node embedding, and set the initial value of the 0 time slice to be a random value.

3.4 The Second Phase: Smooth Evolution

The network embedding of each time slice has been obtained through the above. Next, we introduce how to capture the evolution information of the network embedding through the network embedding of each time slice.

The change of the network topology with time is generally smooth and contains some rules. At the same time, because the network embedding we obtain implies the topology feature of the network, the evolution of the network embedding over time should also be smooth and have some rules. In order to better capture this hidden rule, we use the transformer [15] mechanism proposed by Google to capture the smooth evolution information of network embedding.

Transformer consists of a 6-layer coding layer and a 6-layer decoding layer, where each coding layer is composed of multi-head attention and a single-layer fully connected network. This article uses Multi-head attention to get the next time slice network Embedded, the specific model is shown in Fig. 3.

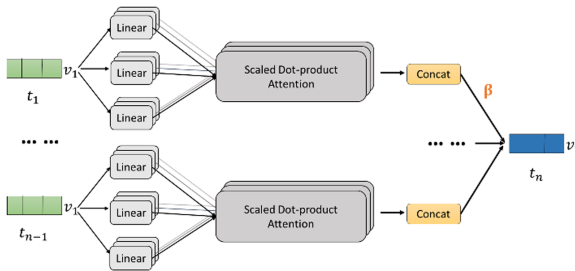


Fig. 3. Smooth Evolution Model

The use of the model in Fig. 3 to capture embedded network evolution information has the following advantages: (1) The number of operations required by the model to calculate the association between two time slices is independent of the distance between the two time slices, so it can capture long distances network evolution information. (2) The model can be calculated in parallel during training. (3) Multi-head attention can capture the evolution information embedded in the network from multiple dimensions, making the results more accurate.

4 SENE-Based Link Prediction Algorithm

The link prediction based on SENE includes the following steps (as shown in Algorithm 1):

Step 1: Dataset division. Divide the network into n different sub-networks according to a certain time interval, respectively: G_1, G_2, \dots, G_n , select the $0-n-1$ time slices as the training set, and the n th time slice network as the test set.

Step 2: Embedding the network of the $0-n-1$ time slices. According to Eq. (4), the $0-n-1$ time slices of the network are embedding, and the negative sampling strategy based on temporal feature is used to optimize the network embedding.

Step 3: Obtain the network embedding of the n th time slice. Use the transformer mechanism to capture the evolution information of network embedding, and finally capture the network embedding of the n th time slice. The network embedding of each node in the $0-n-1$ time slices is input into the evolution model introduced earlier, and the resulting output is the network embedding of the n th time slice.

Step 4: Generate prediction results. First, for each node n that to be predicted in the graph G_n , the similarity between the remaining nodes and node n is calculated. Then, according to the similarity, the top- k nodes are selected as the prediction result of n nodes. Finally, the prediction results of each node in G_n are aggregated and returned.

Algorithm 1. SENE-based link prediction algorithm

```

Input:G
output:{Ni,..., N|V|| Ni•V ∩|Ni|=k ∩ i=1~|Gn|}
1 Divide G into G1,G2...Gn;
2 For g∈{Gi| i>0 && i<=n-1}
3   REPEAT:
4     loss_g = compute_loss(g); //compute loss of g, and to loss_g
5     updata(g,loss_g); //updata embedding of g by loss_g
6   UNTIL Convergence
7 End for
8 Foreach v ∈ G
9   compute vn by vi~vn-1;
10End for
11Foreach n in Gn
12  Sim=ComputeSim(n,Gn);
13 Ni=sort(Sim,k);
14End for
    
```

5 Experiments

5.1 Dataset

We extracts part of the data from the Bibsonomy [16] dataset as an experimental dataset. This dataset contains the tags information that users did to the publications during 2009. There are three types of nodes in the network: user nodes, publication nodes and tag nodes. The statistic of the dataset is shown in Table 1.

Table 1. Bibsonomy network statistics

Type of Edges(A-B)	Number of Node A	Number of Node B	Avg. Degree of A	Avg. Degree of B	Number of Edge
User-Publication	1426	299269	422.8	2.1	601400
User-Tag	1426	116639	341.4	4.2	486787

When dividing the data set, we divide the data set in months. At the same time, we select the data from January to November as the training dataset, and select the data from December as the test dataset. And then, we capture the embedding evolution from January to November to predict the network embedding in December.

5.2 Evaluation Metrics

In this paper, Precision, Recall and AUC [17] curves are used to measure the link prediction results.

In definition, precision is the proportion of real positive examples in the forecast results to the whole forecast results. Recall is the proportion of positive examples in the prediction results to the real results. For this experiment, the real positive example is the exist edges in the test dataset.

AUC curve is defined as the area surrounding the coordinate axis under ROC curve. In practice, the approximate calculation can be made by formula (6). Where, k is the number of comparisons, k' is the number that the similarity of the selected edge in the positive example is greater than similarity that in the negative example, and k'' is the number that the similarity of the selected edge in the positive example is less than or equal to the similarity of negative example.

$$AUC = \frac{k' + 0.5 * k''}{k} \tag{6}$$

5.3 Performance Evaluation

First, from the perspective of temporal, we compare different link methods based on temporal feature, including:

FT: Link prediction method based on frequency time(FT) [18], if two nodes are connected more frequently in the past time slice, the possibility of connection in the next time slice is higher;

GT: Link prediction method based on generation time (GT) [19], if the node pairs are over connected in the closer time slice, the more likely the node pairs are to be connected;

SENE: SENE-based Link Prediction Algorithm proposed by this paper.

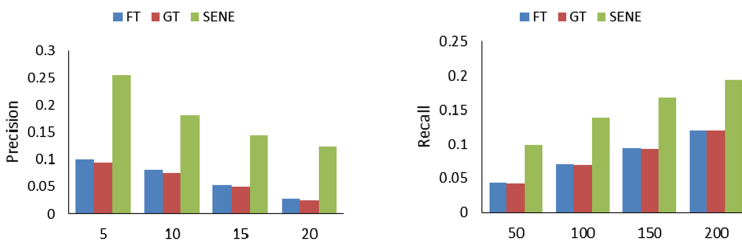


Fig. 4. Performance evaluation of temporal feature-based link prediction methods

The experimental results are shown in Fig. 4. It can be seen from Fig. 4 that the accuracy and recall rate of the model proposed in this paper are much higher than that of the comparison method no matter how much k value is taken. The reasons may be as follows: FT and GT only consider the temporal feature of the network, but ignore the topological structure feature and context feature of the network itself. Therefore, the precision and AUC values are not high. The SENE combines the deep-level features of the network with the temporal feature, which can better predict the upcoming connections. AUC values are shown in Table 2.

Table 2. The values of AUC of temporal feature-based link prediction methods

Method	AUC
FT	0.625
GT	0.587
SENE	0.971

Secondly, from the perspective of network embedding, we compare different network embedding methods, including:

Node2vec: Node2vec [10] is a graph embedding model proposed by Grover et al. Its core idea is to generate random walk sequences, and then use word vector model to represent and learn the nodes in random walk sequences. It forms a good balance between depth and breadth when generating random walk sequences;

Metapath2vec: Metapath2vec [11] is a node embedding algorithm for heterogeneous information networks proposed by Dong et al. It uses the random walk algorithm based on meta-path to construct the domain nodes of each node, and then uses skip-gram model to complete the node embedding;

LINE: LINE [12] is a graph embedding algorithm proposed by Tang et al. The goal is to embed large-scale information network into low dimensional space, and use edge sampling algorithm to solve the problem of gradient descent.

SENE: SENE-based Link Prediction Algorithm proposed by this paper.

Before the experiment, we explain some parameters of the network embedding method. In the experiment, both the baseline and the method proposed in this paper need to set some parameters. In this paper, we choose the embedded dimension as 16, and set the learning rate as 0.001. For node2vec, metapath2vec and LINE, we choose 100 training times, and the training times of SENE is 50. Then, all the data from January to November will be used as the training set of the comparison method, and the data from December will be used as the test set.

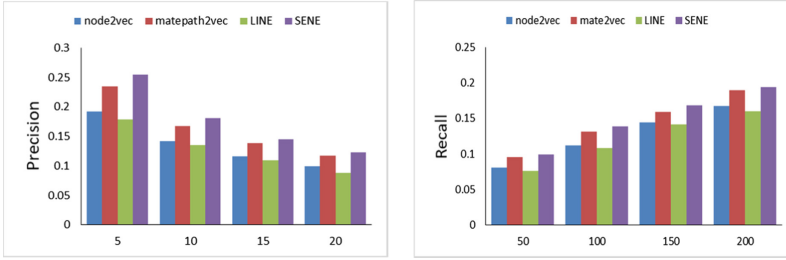


Fig. 5. Performance evaluation of network embedding-based link prediction methods

The experimental results are shown in Fig. 5, which shows that the SENE based link prediction method has the best performance. Node2vec, metapath2vec and LINE considered the topological structure feature and context feature, but ignored the temporal feature of the network. In SENE, the inherent features and temporal feature of the network are considered. The AUC value of the experiment is shown in Table 3. It can be seen from the table that even compared with the best performing method, the method proposed in this paper has nearly 5% improvement.

Table 3. The values of AUC of network embedding-based link prediction methods

Method	AUC
Node2vec	0.826
Metapath2vec	0.924
LINE	0.813
SENE	0.971

6 Conclusion

In view of the shortcomings of the existing link prediction technology, this paper proposed a dynamic link prediction method on heterogeneous information networks. In order to predict the edge in the next slice more accurately, this paper fully considers the deep-level and temporal features of heterogeneous information networks, and proposes the Smooth Evolution model for Network Embedding (SENE). In the model, context feature and topology feature are added at the same time, and a negative sampling strategy based on temporal feature is used to optimize the embedding, and transformer mechanism is used to obtain the smooth evolution of the network embedding, making full use of the rich features of heterogeneous information network. In addition, we propose a SENE-based Link Prediction Algorithm, by calculating the similarity of embedding between nodes to evaluate the possibility of edge formed, effectively ensuring the accuracy of link prediction. In the next step, we will study the capture high-order similarity and algorithm optimization strategy.

Acknowledgment. This work is supported by the National Key R&D Program of China (2018YFB1003404) and the National Natural Science Foundation of China (61672142).

References

1. Chen, C., et al.: Unsupervised Adversarial Graph Alignment with Graph Embedding (2019). ArXiv, abs/1907.00544
2. Mutinda, F.W., Nakashima, A., Takeuchi, K., Sasaki, Y., Onizuka, M.: Time series link prediction using NMF. *IEEE International Conference on Big Data and Smart Computing (BigComp)* **2019**, 1–8 (2019). <https://doi.org/10.1109/BIGCOMP.2019.8679502>
3. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: PathSim: meta path-based Top-K similarity search in heterogeneous information networks. *Proc. VLDB Endow.* **4**, 992–1003 (2011)
4. Martínez, V., Galiano, F.B., Cubero, J.C.: A survey of link prediction in complex networks. *ACM Comput. Surv.* **49**, 69:1–69:33 (2016). <https://doi.org/10.1145/3012704>
5. Liu, Y., Shen, D., Kou, Y., Nie, T.: Link prediction based on node embedding and personalized time interval in temporal multi-relational network. In: Ni, W., Wang, X., Song, W., Li, Y. (eds.) *WISA 2019*. LNCS, vol. 11817, pp. 404–417. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-30952-7_40
6. Mitzenmacher, M.: A brief history of generative models for power law and lognormal distributions. *Internet Math.* **1**, 226–251 (2003). <https://doi.org/10.1080/15427951.2004.10129088>
7. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Comput. Netw.* **30**, 107–117 (1998). [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X)
8. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016). <https://doi.org/10.1145/2939672.2939753>
9. Perozzi, B., Al-Rfou, R., Skiena, S.: DeepWalk: online learning of social representations. *KDD 2014* (2014). <https://doi.org/10.1145/2623330.2623732>
10. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016). <https://doi.org/10.1145/2939672.2939754>
11. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: scalable representation learning for heterogeneous networks. In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017). <https://doi.org/10.1145/3097983.3098036>
12. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: LINE: Large-scale Information Network Embedding (2015). ArXiv, abs/1503.03578
13. Chen, H., Yin, H., Wang, W., Wang, H., Nguyen, Q.V., Li, X.: PME: projected metric embedding on heterogeneous networks for link prediction. In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018). <https://doi.org/10.1145/3219819.3219986>
14. Luong, T., Pham, H., Manning, C.D.: Effective Approaches to Attention-based Neural Machine Translation (2015). ArXiv: abs/1508.04025
15. Vaswani, A., et al.: Attention is All you Need (2017). ArXiv, abs/1706.03762
16. Benz, D., et al.: The social bookmark and publication management system bibsonomy. *VLDB J.* **19**, 849–875 (2010). <https://doi.org/10.1007/s00778-010-0208-4>

17. Norton, M., Uryasev, S.P.: Maximization of AUC and Buffered AUC in binary classification. *Mathematical Programming*, **174**, 575–612 (2019). <https://doi.org/10.1007/s10107-018-1312-2>
18. Divakaran, A., Mohan, A.: Temporal link prediction: a survey. *New Gener. Comput.* **38**(1), 213–258 (2019). <https://doi.org/10.1007/s00354-019-00065-z>
19. Li, D., Shen, D., Kou, Y., Lin, M., Nie, T., Yu, G.: Research on a link-prediction method based on a hierarchical hybrid-feature graph. *Sci. Sin. Inform.* **50**, 221–238 (2020). <https://doi.org/10.1360/N112018-00223>