Dynamic Network Embedding By Time-Relaxed Temporal Random Walk

Yifan Song[1], Darong Lai[1,2], Zhihong Chong[1,2], and Zeyuan Pan[1]

[1] School of Computer Science and Engineering, Southeast University, Nanjing, China
[2] MOE Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing, China
syfufoasa@gmail.com, daronglai@seu.edu.cn, chongzhihong@seu.edu.cn, yggdrasilyuan@gmail.com

Abstract. Network embedding, also called network representation learning, aims at mapping high dimensional network information into low dimensional vectors. Previous studies mainly focus on static networks. In recent years, dynamic network embedding attracts much attention and methods specific to dynamic network are emerging. However, previous dynamic network embedding methods, such as CTDNE, still have drawbacks when using random walk to generate node sequences. Temporal random walk strictly requires that the time value of next edge be larger (i.e. later visiting) than that of the previous visited edge, which often leads to insufficient information obtained by random walk. In this article, a novel model named **T**ime-Rela**x**ed **T**emporal Random **Walk**(**TxTWalk**) for dynamic network embedding is proposed. Firstly, a time-relaxed function is designed, which enables random walk to select the next edge in a time interval, not strictly larger than the time of previously visited edge. It can make the walking sequences obtained by **TxTWalk** contain a wider range of temporal information. Then the node sequences are put into the skip-gram model for training to generate embedding of nodes on dynamic networks. Experimental validations on various networks demonstrate that **TxTWalk** is more effective than other state-of-the-art methods in link prediction.

Keywords: network embedding · network representation learning · link prediction · random walk

1 Introduction

Networks arise naturally in applications that contain a great deal of useful information about the world around. In recent years, network embedding, also called network representation learning, which associates nodes with low-dimensional numerical vectors has become an effective way to process and use these information.

Existing studies mainly focus on static network by preserving as much as possible the structure of a network, such as **Deepwalk**[15], **LINE**[20] and **node2vec**[8]. Nonetheless, real-world networks always change over time, which provide a large amount of temporal information. For example, figure 1 shows a co-author network, whose nodes indicate authors and edges represent the collaboration between authors. The timestamp of an edge means the time of collaboration. In this network, the larger the timestamp of an edge is, the more similar the two connected nodes are (more recent collaboration). Therefore, node 6 and 8 in figure 1 will be closer to node 5 than node 4 after considering the timestamps of the edges, while in static network neglecting timestamps node 4, 6 and 8 are "equivalently close" to node 5.

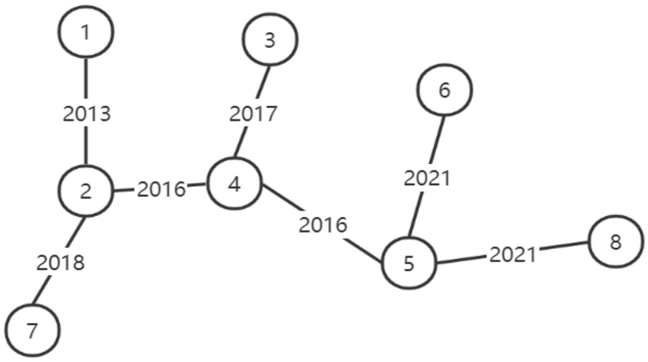

Fig. 1. An example of temporal network. Each edge is annotated with a timestamp denoting when the edge was created.

In order to generate better network embedding in dynamic networks, dynamic network embedding not only aims at preserving network structure but also takes into account temporal information[10]. Among previous dynamic network studies, one typical algorithm is **CTDNE**[13]. **CTDNE** takes into account the time information of each edge, that is the timestamp of an edge, in the random walk sampling of nodes, and requires all sampled edges in a random walk to be within a specific time interval. At the same time, the timestamp of each edge on the random walk path should be greater than that of the previous edge.

CTDNE[13] as well as other random walk based dynamic network representation learning methods perform well, however, they still have drawbacks when using random walk to generate node sequences. Temporal random walk strictly requires that the timestamp of next edge be larger (i.e. later visiting) than that of the previous visited edge. For instance, as shown in figure 1, when the edge between node 2 and 4 is selected, the next edge can not be the one connecting node 4 to node 5 since its timestamp is not larger than that of the edge between node 2 and 4. As a result, this type of methods may lead to insufficient information obtained by random walk when the timestamp of the initial edge is large.

For example, in the co-author network, as shown in figure 1, author 4 may first publish a paper with author 5 in 2016, and then published a paper with author 3 in 2017. It is obvious that node 3 should have a high degree of similarity to node 5. But in **CTDNE**, it is impossible to get a random walk from author 3 to author 5 because the timestamp of the edge from node 3 to node 4 is larger than the edge from node 4 to node 5. This situation is obviously unreasonable, and how to get temporal random walk to avoid this phenomenon becomes important.

In this article, a novel model called **T**ime-Rela**x**ed **T**emporal Random **Walk** (**TxTWalk**) has been designed for dynamic network representation learning. Firstly, a time relaxed function is designed, which enables the random walk to select the next edge in a time interval, not strictly larger than the timestamp of previous visited edge. It can make the walking sequences obtained by **TxTWalk** contain a wider range of temporal information. Then the node sequences are put into the skip-gram model for training to generate embedding of the nodes of dynamic networks.

In summary, the main contributions in this article are summarized as follows:

1)A new temporal random walk strategy is proposed by using time-relaxed function, which relaxes strict time restriction of temporal random walk.

2)A novel dynamic network embedding framework **TxTWalk** is proposed, which uses time-relaxed temporal random walk to sample node sequences and skip-gram to learn node representations of a network.

3)Experiments on various dynamic networks and the comparisons to state-of-the-art baseline network embedding methods are conducted to verify the effectiveness of **TxTWalk**.

2 Related Work

Most of the early classical methods to obtain network embedding are based on graph factorization[2–4]. With the introduction of word embedding technology, some methods based on skip-gram model are used to solve the problem of network embedding. Deepwalk[15] bridges the gap between network embedding and word embedding by treating nodes as words and generates short random walks as sentences. Skip-gram[12] can then be applied to these random walk sequences to obtain network embedding. LINE[20] defines the first-order and the second-order similarity for nodes. node2vec[8] introduces a biased random walk by adjusting two hyper parameters of the random walk.

For the past few years, researchers have focused on studying more intricate networks, such as heterogeneous network and attributed networks. HIN2Vec[6] defines metapaths in heterogeneous networks and proposes a model that use logistic classification to learn the embedding. GraphSAGE[9] uses neural network to generate embedding, which is useful for graphs that have rich node attribute information.

However, the great mass of study looked at dynamic networks[10], they study the dynamic network by dividing it into multiple snapshots. For example, DANE[11] considers situations in which attribute matrices of the networks evolve over time and learns the embedding by neural network. **CTDNE**[13] differs from other works that it uses continues-time dynamic network and can be served as a basis to popularize other deep learning methods based on random walk. Various other network embedding methods for dynamic networks are emerging, and details can be seen from the excellent surveys on dynamic network embedding[21, 1].

3 Notations and Problem Definitions

In this section, the related notations and definitions of dynamic network embedding problem are given.

Definition 1. *(Continuous-Time Dynamic Network).A Continuous-Time Dynamic Network is a network with each edge being associated with a real value timestamp, which can be represented in the following way:*

$$G = \{V, E_T, T\}$$

where V is the set of nodes in the network, E_T is set of edges between nodes in V, and T is the set of all times of edge emergence. Each edge in E_T of the network is:

$$e_{u,v} = (u, v, t) \in E_T$$

where u and v are the two ending nodes of the connected edge $e_{u,v}$, respectively, and $t \in T$ is the timestamp of that edge.

Definition 2. *(Temporal Random Walk).A temporal walk [13] from v_1 to v_k in G is a sequence of nodes $(v_1, v_2, \cdots, v_k)$ such that $(v_i, v_{i+1}) \in E_t$ for $1 \leq i < k$, as well as $T(v_i, v_{i+1}) \leq T(v_{i+1}, v_{i+2})$ for $1 \leq i < (k-1)$. When v_{i+1} is randomly selected after v_i in the node sequence $(v_1, v_2, \cdots, v_k)$, it is a temporal random walk. For two arbitrary nodes $u, v \in V$, we say that u is temporally (randomly) connected to v if there exists a temporal (random) walk from u to v.*

The link prediction in continuous-time dynamic network is to predict the existence of a link after time t on the basis of the edges generated until time t. Formally, the link prediction problem is

Definition 3. *(Link Prediction). A graph $G = (V, E_T^o, E_T^p, T)$ and two nodes $u, v \in V$ are given. Function Φ learning on the basis of E_T^o maps $\{u, v\}$ from E_T^p to 1 or 0:*

$$\Phi((u, v)) = \begin{cases} 1, \text{ then } e_{u,v} \in E_T^p \\ 0, \text{ otherwise} \end{cases}$$

where E_T^o is the set of the edges observed at time t, and E_T^p contains edges to be predicted after that time.

4 Time-Relaxed Temporal Random Walk(TxTWalk)

Given a temporal random walk (Definition 2), the temporal neighbourhood of node v is defined:

$$\Gamma_t(v) = \{(w, t') \mid e_{v,w} = (v, w, t') \in E_T \ \wedge \ t' \geq t\} \tag{1}$$

Here $\Gamma_t(v)$ is the set of node v's temporal neighbours after time t. $e_{v,w}$ represents the edge that starts at v and ends at w, generated at time t'. Temporal random walk only allows the selection of the next node within the temporal neighborhood. The rationale behind such neighborhood, as adopted in CTDNE[13], is that this definition can ensure the random walk respects the edge generation time and works logically in networks transmitting flows. A typical example is, in an email network, a person cannot forward an unreceived message to someone else in this network.

However, in a Users-Goods network as shown in figure 2, if the initial edge formed in a temporal random walk is from User 1 to Goods 1, then random walk will not be able to obtain the information of other nodes in figure 2, for the timestamp of the next edge in a walk needs to be strictly larger than that of the previous edge. But in our common sense, since User 1 and User 2 have bought the same goods, Goods 1 purchased by User 1 should be similar to Goods 3, which has been purchased by User 2.

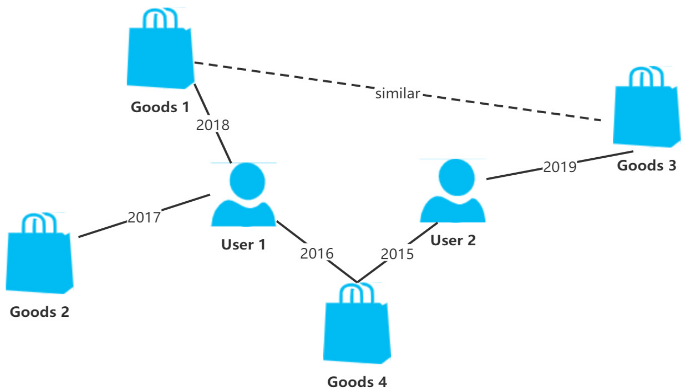

Fig. 2. An example of Users-Goods network.

To address the above-mentioned issue, **T**ime-Relax**e**d **T**emporal Random **Walk**(**TxTWalk**) has been proposed for continuous-time dynamic network embedding, which is inspired by the idea of image repairing[7]. In image repairing problem, to repair a given image, the input for training a neural network model are the original image together with all motion frames one second before the original image. Motivated by this, it is meaningful to use the edges whose timestamp is earlier than that of the current edge. Therefore, the timestamp restriction of selecting the next edge in temporal random walk can be relaxed by the following function:

$$\Gamma_t(v) = \{(w, t') \mid e = (v, w, t') \in E_T \ \wedge \ t' \geq \Theta(w, t)\} \tag{2}$$

$$\Theta(w, t) = t * max(\gamma, min(\sigma, \frac{t_{max} - t + 1}{t - t_{min} + 1})) \tag{3}$$

Here, $\Theta(w, t)$ is called time-relaxed function, and σ and γ are two parameters that can be manually adjusted based on the temporal information of a network. When the initial time(the timestamp of the first edge in random walk) is large and the time span in T is small, σ should be set to be large(but still smaller than 1) and γ be small(larger than 0). These two parameters are used to control maximum and minimum degree of relaxation. In relaxed function $\Theta(w, t)$, t_{max} and t_{min} represent the maximum and minimum timestamps of all edges starting with node w, respectively. $\frac{t_{max}-t+1}{t-t_{min}+1}$ ensures random walk can relax the time restriction of selecting the next edge when the the timestamp of currently selected edge is large. It is not difficult to show that CTDNE[13] is in fact a special case of **TxTWalk** when $\sigma = 0$ and $\gamma = 1$.

Algorithm 1: Random Walk Sampling

Input: A continuous-time dynamic network $G = \{V, E_T, T\}$, initial edge (s, r), number of context windows X, maximum walk length l.
Output: Random Walk $S_t = \{v_i, v_{i+1}, ..., v_{i+l}\}$

$S_t \leftarrow (s, r)$;
$i \leftarrow r$;
for $k = 1 \rightarrow min(l, X)$ **do**
    $\Theta(i, t) = t * min(\sigma, \frac{t_{max}-t+1}{t-t_{min}+1})$;
    $\Gamma_t(i) = \{(w, t') \mid e = (i, w, t') \in E_T \ \wedge \ t' \geq \Theta(w, t)\}$;
    if $|\Gamma_t(i)| > 0$; **then**
        $(j, t') \leftarrow$ *select a random element in* $\Gamma_t(i)$ *using equation* 4;
        Append j to S_t;
        $t \leftarrow t'$;
        $i \leftarrow j$;
    else
        terminate random walk
    end
end
return S_t;

During random walk sampling, the probability of each edge being selected is defined:

$$Pi(e) = \frac{exp(t_e - \Theta(w, t))}{\sum_{e' \in E_T} exp(t_{e'} - \Theta(w, t))} \tag{4}$$

Here Pi represents the probability that the edge e is selected. According to this equation, edges whose timestamp are larger can be more easily selected.

As a result, multiple random walks can be sampled in a dynamic network by equations 2 and 4.

The main advantage of **TxTWalk** over traditional static network embedding methods such as Deepwalk[15] is that it can effectively utilize the temporal information contained in dynamic networks. Compared with the dynamic network representation learning methods as CTDNE[13], **TxTWalk** has an advantage to avoid the insufficiency of random walk sampling.

Finally, the skip-gram model[12] is used to learn network embedding. Given a node sequence, the optimization of training skip-gram model is defined as follows:

$$minimize \ \ -logPr(\{v_{i-w},...,v_{i-1},v_{i+1},...,v_{i+w}\} \mid \Phi(v_i)) \tag{5}$$

here $\Phi(v_i)$ is the vector representation of node v_i. Node sequences sampled by random walk are fed into the skip-gram model to learn vector representations of nodes (network embedding).

Algorithm 2: TxTWalk Network Embedding

Input: A continuous-time dynamic network $G = \{V, E_T, T\}$, number of context windows X, minimum walk length ω, maximum walk length l, maximum number of walk μ, set of temporal random walk H_t, embedding dimension D.
Output: Node embedding matrix **Z**.

while $\mu - X > 0$ **do**
  Sample an edge e = (u,v,t) by uniform distribution;
  $S_t = RandomWalkSampling(G, e = (u, v), X = \omega + \mu - C - 1, l)$;
  if $|S_t| > \mu$; **then**
    Append S_t to H_t; $X \leftarrow X + |S_t| - \omega + 1$;
  end
end
Z = Skip-gram(μ, D, H_t);
return **Z**;

The learning algorithm is shown in Algorithm 1 and Algorithm 2, where ω is the minimum length of temporal random walk and is equivalent to the context window size for skip-gram[12].

5 Experiments

Experiments on five network datasets are conducted to demonstrate the effectiveness of the proposed **TxTWalk** model. Based on the experimental results, the following questions are to be answered.

- **Q1** Is **TxTWalk** model better than the baseline methods on the tested networks (especially better than **CTDNE** [13], an effective dynamic network embedding method)?

– **Q2** How different values of the parameters σ and γ can affect the performance of **TxTWalk**?

5.1 Experimental Setup

Datasets. The proposed **TxTWalk** model for dynamic network embedding is tested by applying to link prediction on five real-world network datasets of different scales. The datasets are listed with their topological information in Table 1 and detailed in the following.

Table 1. Topological information of the datasets

Data	$\|V\|$	$\|E\|$	Average degree
Enron	150	268	3.72
Forum	899	33720	74
FB-messages	1899	61734	63
CollegeMsg	1899	59835	112
ca-cit-HEPTH	22907	2673133	233

- **Enron**[18] is a email communication network. Nodes in this network represent employees while each edge represents a email between two employees.
- **Forum**[14] is a network that records users activities on Facebook. Nodes in this network indicate unique users and edges from u to v point out user u follows the user v.
- **FB-messages**[5] is a social network of students at University of California. The network includes the users that sent or received at least one message. Nodes in this network represent users while edges represents communications between users.
- **CollegeMsg**[16] is a network of messages sending at the University of California, Irvine. Nodes are users that could search for others to initiate conversations. An edge (u, v, t) means user u sent a private message to user v at time t.
- **ca-cit-HEPTH**[19] is from arXiv and covers all paper citations. Edge from paper u to paper v indicate that paper u cites another paper v.

Baseline Methods. For comparison, **TxTWalk** are compared to the following network embedding methods that either use temporal information or not.

- **Deepwalk**[15] is a static network embedding method which firstly treats nodes as words and generates short random walks as sentences to learn vector representations of nodes.
- **LINE**[20] defines the first-order and the second-order similarity for nodes, and optimizes the skip-gram model to learn nodes representations. It is also a static network embedding method.

- **node2vec**[8] employs a second order biased random walk controlled by two hyper-parameters to obtain node sequences for learning static network embedding.
- **CTDNE**[13] is a dynamic network embedding method which can make use of temporal information by temporal random walk.

Evaluation Metric. In the experiments, Area Under Curve (**AUC**) is used as the evaluation metric, which represents the area under the Receiver Operating Characteristic (**ROC**) curve [17]. The model performs better if **AUC** is closer to 1.

Parameter Settings. In all experiments, the parameters $\sigma = 0.85$ and $\gamma = 0.6$. The number of walks $\mu = 80$ and the length of walk $l = 10$. For all baseline methods, other parameters are set to the suggested values in the original papers, and the embedding dimension is fixed to 128.

5.2 Experimental Results

Link Prediction To test the effectiveness of **TxTWalk** model, it was firstly trained to learn the vector embedding of each node. Then, the feature vector of an edge is computed by averaging the vectors of the two ending nodes of that edge. Logistic regression (LR) with hold-out validation of 25% edges is used on all datasets. Experiments are repeated for 10 random initialization and the averaged results compared with other baseline methods are shown in table 2. From table 2, **TxTWalk** outperforms almost all the baseline methods on all networks no matter whether the temporal information is used or not, which answers the question **Q1**. It is interesting to find that node2vec is slightly better than **TxTWalk** when tested on CollegeMsg network. By carefully examining the network, it is found that large portion of the edge associated times in CollegeMsg network falls in a narrow time interval, indicating that the temporal information in this network plays less important role.

Table 2. **AUC** score compared to baselines with 75% training links.

Data	**Deepwalk**	**LINE**	**node2vec**	**CTDNE**	**TxTWalk**
Enron	82.12 ± 0.58	83.04 ± 0.48	83.70 ± 0.57	84.12 ± 0.65	$\mathbf{87.02} \pm 0.60$
Forum	71.30 ± 0.42	72.48 ± 0.43	71.10 ± 0.41	74.53 ± 0.61	$\mathbf{77.33} \pm 0.50$
FB-messages	68.00 ± 0.38	67.74 ± 0.57	67.54 ± 0.62	83.88 ± 0.60	$\mathbf{87.75} \pm 0.44$
CollegeMsg	90.22 ± 0.67	90.74 ± 0.68	$\mathbf{92.37} \pm 0.71$	89.88 ± 0.54	91.05 ± 0.52
ca-cit-HEPTH	66.97 ± 0.65	67.51 ± 0.27	68.42 ± 0.48	83.51 ± 0.50	$\mathbf{84.41} \pm 0.41$

When the temporal information is taken into account, it also shows that **TxTWalk** is more effective than CTDNE[13]. To fully reveal the superiority of **TxTWalk** to CTDNE on link prediction, different proportions of edges are used to form training set of FB-messages network. The results are shown in table 3. From table 3, **TxTWalk** works better in FB-messages than **CTDNE** under different proportions of training links, which shows that **TxTWalk** is more effective than **CTDNE**.

Table 3. The comparison between **TxTWalk** and **CTDNE** on link prediction with different proportion of training links of FB-messages network.

Data	**30%**	**45%**	**60%**	**75%**	**90%**
TxTWalk	**80.02** $\pm$ 0.48	**82.85** $\pm$ 0.50	**85.27** $\pm$ 0.42	**87.75** $\pm$ 0.44	**88.12** $\pm$ 0.62
CTDNE	79.78 $\pm$ 0.49	80.81 $\pm$ 0.44	82.01 $\pm$ 0.46	83.88 $\pm$ 0.60	84.22 $\pm$ 0.52

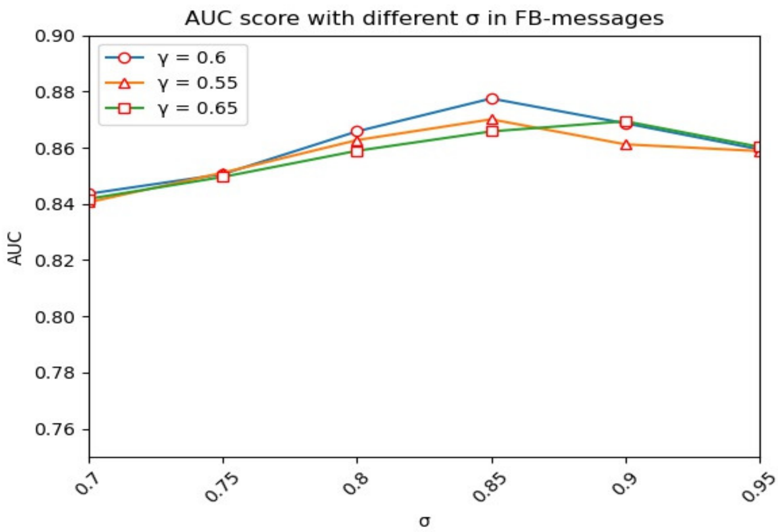

Fig. 3. Parameters sensitivity analysis

Parameters sensitivity analysis For the question **Q2**, the impact of different values of the parameters is evaluated on FB-messages network. Different values of σ={0.7, 0.75, 0.8, 0.85, 0.9, 0.95} and γ={0.55, 0.6, 0.65} are searched respectively, and results of link prediction on FB-messages network with 75% training link are given. As shown in figure 3, with the increase of σ value, AUC first increases and then decreases, and the score is highest when γ=0.6 and σ=0.85. When σ is too large, as can be shown in equation 3, σ takes no effect. If σ is too small then $min(\sigma, \frac{t_{max}-t+1}{t-t_{min}+1}) = \sigma$, leading random walk to obtain useless information. Similarly, the value of γ can not be too large or too small, the reason of which is similar to that of σ. Therefore, γ=0.6 and σ=0.85 are the optimal setting of parameters, which were used in all experiments.

6 Conclusions and discussions

In this article, a novel dynamic network embedding model **TxtWalk** is proposed. **TxtWalk** employs time-relaxed temporal random walk to sample node sequences that respect the times of edge emergence. In contrast to previous random walk based dynamic network embedding methods, such as CTDNE, requiring strict time respecting, **TxTWalk** relaxes such requirement to allow edges appear earlier within a small interval to be sampled to produce node sequences. **TxTWalk** first uses a time-relaxed temporal random walk to sample a series of node sequences starting from given nodes, and then employs skip-gram model to learn network embedding. Experiments are conducted on five networks and the results demonstrate that **TxTWalk** is effective and outperforms state-of-the-art network embedding methods.

Although Skip-gram is adopted in **TxTWalk** for leaning node representations, other neural network model can be used. In addition, **TxTWalk** is currently applied only to homogeneous dynamic networks in this article, extensions of **TxTWalk** to deal with complex dynamic heterogeneous networks are also feasible. We leave these for future work.

References

1. Aggarwal, C., Subbian, K.: Evolutionary network analysis: A survey. ACM Comput. Surv. **47**(1) (May 2014). https://doi.org/10.1145/2601412, https://doi.org/10.1145/2601412
2. Ahmed, A., Shervashidze, N., Narayanamurthy, S., Josifovski, V., Smola, A.J.: Distributed large-scale natural graph factorization. In: Proceedings of the 22nd International Conference on World Wide Web. p. 37–48. WWW '13, Association for Computing Machinery, New York, NY, USA (2013). https://doi.org/10.1145/2488388.2488393
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation **15**(6), 1373–1396 (2003). https://doi.org/10.1162/089976603321780317
4. Cao, S., Lu, W., Xu, Q.: Grarep: Learning graph representations with global structural information. In: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. p. 891–900. CIKM '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2806416.2806512
5. Opsahl, T., Panzarasa, P.: Clustering in weighted networks. Social networks **31**(2), 155–163 (2009)
6. Fu, T.y., Lee, W.C., Lei, Z.: Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. p. 1797–1806. CIKM '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3132847.3132953, https://doi.org/10.1145/3132847.3132953

7. Godard, C., Mac Aodha, O., Firman, M., Brostow, G.J.: Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
8. 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. pp. 855–864 (2016)
9. Hamilton, W.L., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. In: Proceedings of the 31st International Conference on Neural Information Processing Systems. pp. 1025–1035 (2017)
10. Hisano, R.: Semi-supervised graph embedding approach to dynamic link prediction. In: Cornelius, S., Coronges, K., Gonçalves, B., Sinatra, R., Vespignani, A. (eds.) Complex Networks IX. pp. 109–121. Springer International Publishing, Cham (2018)
11. Li, J., Dani, H., Hu, X., Tang, J., Chang, Y., Liu, H.: Attributed network embedding for learning in a dynamic environment. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. p. 387–396. CIKM '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3132847.3132919, https://doi.org/10.1145/3132847.3132919
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
13. Nguyen, G.H., Lee, J.B., Rossi, R.A., Ahmed, N.K., Koh, E., Kim, S.: Continuous-time dynamic network embeddings. In: Companion Proceedings of the The Web Conference 2018. p. 969–976. WWW '18, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE (2018)
14. Opsahl, T.: Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. Social Networks (2011)
15. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 701–710 (2014)
16. Pietro Panzarasa, Tore Opsahl, K.M.C.: Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. In: Journal of the American Society for Information Science & Technology (2009)
17. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining. pp. 43–48 (1997)
18. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015), http://networkrepository.com
19. Rossi, R.A., Ahmed, N.K.: The network data repository with interactive graph analytics and visualization. In: AAAI (2015), http://networkrepository.com
20. Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., Mei, Q.: Line: Large-scale information network embedding. In: Proceedings of the 24th International Conference on World Wide Web. pp. 1067–1077 (2015)
21. Xie, Y., Li, C., Yu, B., Zhang, C., Tang, Z.: A survey on dynamic network embedding. CoRR **abs/2006.08093** (2020), https://arxiv.org/abs/2006.08093