2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)

A Universal Method Based on Structure Subgraph Feature for Link Prediction over Dynamic Networks

Xiao Li*, Wenxin Liang†, Xianchao Zhang*, Xinyue Liu* and Weili Wu‡
*School of Software, Dalian University of Technology, Dalian, China
Email: lixiaodlut@mail.dlut.edu.cn, {xczhang, xyliu}@dlut.edu.cn
†School of Software Engineering
Chongqing University of Posts and Telecommunications, Chongqing, China
Email: wxliang@cqupt.edu.cn
‡Dept. of Computer Science, University of Texas at Dallas, Dalla, USA
Email: weiliwu@utdallas.edu

***Abstract*—In dynamic networks, links are annotated with timestamps showing the emerging time and the link prediction problem is to infer the future links in networks. Universal link prediction methods are highly demanded in various applications, which require universal link features that are feasible for multiple kinds of network topological structures and capable to address the difference of links with different timestamps. In this paper, we propose a novel link feature called *S*tructure *S*ubgraph *F*eature (SSF). The SSF is an outstanding link feature that is feasible to various dynamic networks due to the following superiorities: (1) the proposed structure subgraph is so far the most effective manner to represent surrounding topological features of target link and (2) the normalized influence well specifies the influence of multiple links and different timestamps in structure subgraph. We finally propose two link prediction methods by applying SSF to a linear regression model and a neural machine. Experimental results on real-world dynamic network datasets indicate that the SSF-based methods consistently provide top-class performance on various dynamic networks.**

***Index Terms*—dynamic networks, link prediction, structure subgraph**

I. Introduction

Numerous networks in real world are dynamic, which means the links in networks emerge at different time. For example, the coauthor network between scholars is a typical kind of dynamic networks, because the coauthor relationships are formed in different years. For theoretical modeling and analyzing in this paper, links in dynamic networks are annotated with timestamps indicating the time they emerged. Recently link prediction attract much interest in dynamic networks, which aims to infer the future links of a given network. Link prediction has plentiful applications in various areas, such as personalized recommendation in social or e-commerce networks [1], [2], link recovery in knowledge graphs [3], entity resolution [4], user behavior prediction [5] and missing protein interaction discovery in biochemical reaction networks [6] . Due to the increasing demand from various application in different dynamic networks, it is imperative to design universal link prediction methods that are feasible for various dynamic networks.

The basic idea to infer whether a link will be created is to measure the similarity or closeness between the two end nodes of the link [7]. State-of-the-art ranking models [8]–[11] or classification models [12]–[14] have developed various features to measure or represent the closeness. Some features for ranking models [1], [7], [9], [15]–[19] directly calculate the closeness score between two nodes. While the feature for classification models [14] is defined as a feature vector that represents the link based on surrounding structures. Table I presents the most popular link features for link prediction. The existing features either lack universal applicability to different kinds of network structures, or only focus on static networks where the links have no difference in emerging time [20], [21]. Therefore, none of them is universal for various network structures and applicable to link prediction in dynamic networks.

The problem of the features that are not feasible for multiple kinds of network structures comes from the demerit that only utilize one or two specific kinds of network topological information. For example, Common Neighbor (CN) [7] only considers the number of common neighbors of two end nodes, while Preferential Attachment (PA) [15] only calculates the degree of the two end nodes. Therefore, such kind of features may make improper evaluation about the closeness of the end nodes and hence make wrong links prediction results.

Figure 1(a) shows two surrounding networks of two target links, $A-B$ and $X-Y$, for example in Twitter network. The nodes and links represent users and comments between users (directions are ignored), respectively. This network is a typical dynamic network, where the links are annotated with timestamps showing the time of making comments, and multiple links are allowed between two nodes. We aim to predict whether a link will be created between A and B or X and Y. A, B, and C have high degrees in the network, which semantically means they are celebrities and many fans make comments to their tweets. While X and Y are more likely to be the common users and both of them are the fans of C.

This work is partially supported by National Science Foundation of China (Grant No. 61632019 and No. 61572096).
† corresponding author.

2575-8411/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDCS.2019.00122

TABLE I: Comparison of Link Features for Link Prediction

feature name	formulas	universal	dynamic
CN [7]	$\lvert\Gamma_x \cap \Gamma_y\rvert$	×	×
PA [15]	$\lvert\Gamma_x\rvert \cdot \lvert\Gamma_y\rvert$	×	×
Jac. [16]	$\frac{\lvert\Gamma_x \cap \Gamma_y\rvert}{\lvert\Gamma_x \cup \Gamma_y\rvert}$	×	×
AA [1]	$\sum_{z\in\Gamma_x\cap\Gamma_y} \frac{1}{log\lvert\Gamma_z\rvert}$	×	×
RA [17]	$\sum_{z\in\Gamma_x\cap\Gamma_y} \frac{1}{\lvert\Gamma_z\rvert}$	×	×
RW [18]	$p_x^t = M^T p_x^{t-1}$	×	×
Katz [19]	$\sum_{l=1}^{\infty} \beta^l (A^l)_{xy}$	×	×
rWRA [9]	$\sum_{z\in\Gamma_x\cap\Gamma_y} \frac{W_{xz}\cdot W_{yz}}{S_z}$	×	√
WLF [14]	link feature vector	√	×
SSF (our work)	link feature vector	√	√

Notes: "universal" means the feature is applicable to multiple kinds of network structures, "dynamic" means the feature can handle the different emerging time of links in dynamic networks. Γ_x denotes the neighbor set of node x, $\lvert\cdot\rvert$ is the size of a set. W is a weighted adjacency matrix, where W_{xz} is the link weight from x to z and $S_z = \sum_{z'\in\Gamma_z} W_{zz'}$. Random walk is defined in a recursive manner, M is the transition probability matrix defined by adjacency matrix A normalized by rows, where the entry $M_{xy} = A_{xy} / \sum_{k\in\Gamma_x} A_{xk}$. β is a damping factor.

Since both the two celebrities A and B frequently interact with another celebrity C, it is of higher probability that A and B will make comments to each other than the two C's common fans X and Y do in the future. That is, link $A-B$ has higher probability to be created than link $X-Y$.

Figure 1(b) presents several popular link features that utilize the surrounding structures to measure the closeness between $A-$B and $X-Y$. Only utilizing several specific kinds of information, common neighbors (CN) [7], Adamic-Adar (AA) [1], resource allocation (RA) [17] and reliable weighted resource allocation (rWRA) [9] can not differentiate the closeness of $A-B$ and $X-Y$, which results in predicting the same probability of $A-B$ and $X-Y$ to be created. Although preferential attachment (PA) [15] and Jaccard index (Jac.) [16] show the difference of surrounding structures between $A-B$ and $X-Y$, they still ignore the fact that C is a celebrity with high degree and plays important role to the emergence of $A-B$ and $X-Y$.

Zhang and Chen [14] propose a neural machine classification model which represents the topological information of K neighbor nodes of a target link as a feature vector (we denoted the feature as WLF in this paper).WLF utilizes all kinds of topological information encoded in the structure of surrounding fixed K nodes, which makes it applicable to various networks. However, WLF is proposed for static networks, and in many real-world networks, WLF still not effective enough to encode topological information. An example is shown in Figure 1. When $K=6$, WLF can only utilize the structure information of surrounding 6 nodes of $A-B$ and $X-Y$. WLF can not differentiate the surrounding structures of $A-B$ and $X-Y$ and ignores the roles of A, B and C in the networks. Manually increasing K to involve more nodes into the enclosing subgraph may solve the problem in this example, but it is hard to find a proper K for all networks.

In order to utilize topological structure and represent topological information in dynamic networks more efficiently, in this paper we propose the structure subgraph which is so far the most effective manner to represent the surrounding topology of target links. The nodes in a structure subgraph are called structure nodes, and a structure node is defined as an aggregation of the nodes with the same structure in a specific

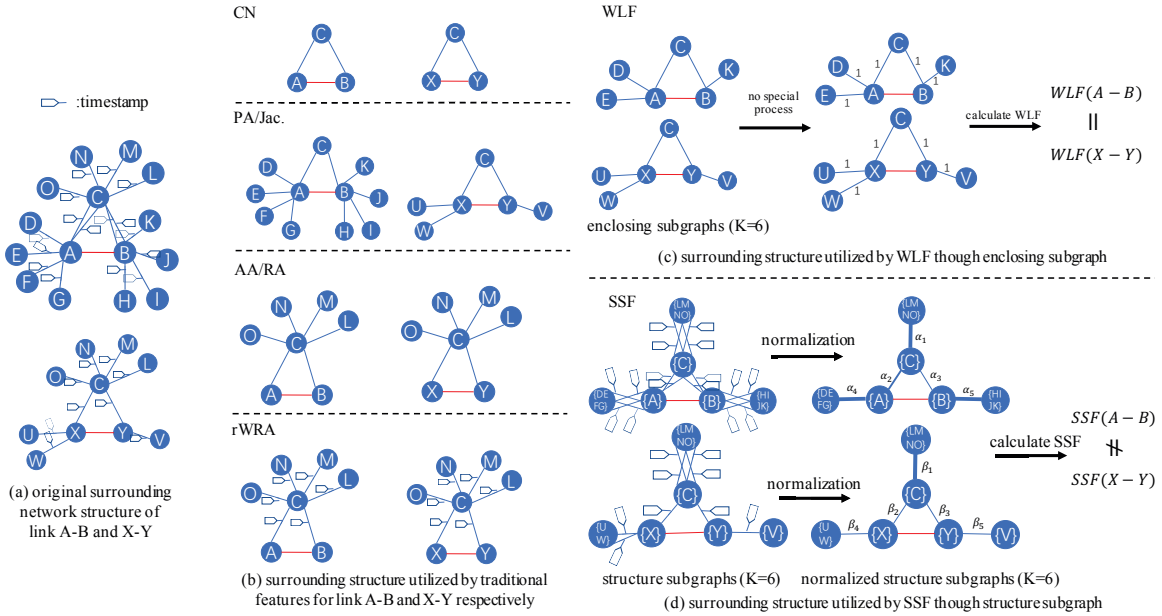


Fig. 1: Comparison of network structure utilized by various link features

surrounding network structure. Figure 1(d) shows the structure subgraphs constructed for $A-B$ and $X-Y$ when $K=6$. From Figure 1(d), we can learn that the structure subgraph is the only one that can represent all the surrounding topological information of $A-$B and $X-Y$.

On the other hand, the different timestamps of links and multiple links between nodes are important characteristics in dynamic networks, which brings different influence to the emergence of new links. It is obvious that the links emerged in farther history have less influence to the emergence of new links at present time, and the connections of two nodes with multiple links bring more influence than those with one link. To address these characteristics in dynamic networks, in this paper we first apply an exponential influence decay function to specify the influence of every single link with different timestamps. Since the nodes in the the same structure node plays the same role in networks, we further utilize a strategy to specify the influence of multiple links which sums all influence of links between two structure nodes as one normalized influence. Then, we propose the *S*tructure *S*ubgraph *F*eature (SSF) which is a feature vector calculated by unfolding the adjacency matrix of the normalized structure subgraph where the influence of links are all normalized. Figure 1(d) illustrates the process of obtaining SSF, where α and β are the values of normalized influence.

Finally, we propose two link prediction methods by applying the SSF to a linear regression model and a neural machine. Due to the abundant topological information encoded in SSF, the SSF-based methods are universally applicable for various network structures in dynamic networks. The main contributions of this paper are summarized as follows:

- We propose the structure subgraph which has outstanding ability to represent the surrounding network structures of a target link. It provides the theoretical foundation for the proposed link feature and link prediction methods.
- We integrate all the influence of links between two structure nodes into a normalized influence. The normalized influence can simultaneously specifies the effect of multiple links between two nodes and different emerging time of links in dynamic networks.
- We further propose a feature vector called *S*tructure *S*ubgraph *F*eature (SSF) by unfolding the adjacency matrix of structure subgraph where the influence of links are normalized. In stead of capturing only several kinds of information from the surrounding structures, SSF can automatically encode the abundant topological information from the structure subgraph into feature vectors, which makes SSF consistently feasible for various dynamic networks.
- We evaluate the superiority of SSF by applying it to a linear regression model and a neural machine, namely SSFLR and SSFNM. Then we compare these two link prediction methods with 11 baseline methods on 7 real-world dynamic networks. The experimental results demonstrate that the SSF-based methods outperform the baseline methods and provide consistently top-class performance for link prediction tasks in various dynamic networks.

The remainder of this paper is organized as follows. We present related work in Section II. We formalize dynamic networks and link prediction problem in Section III. Then we propose structure subgraph and structure subgraph feature in Section IV and Section V, respectively. In Section VI, we conduct extensive experiments on real-world dynamic datasets. Finally, we conclude this paper in Section VII.

II. Related Work

Link prediction problem is studied in static networks initially [7], where the links are not annotated with timestamps and networks are modeled as conventional graphs. Link prediction problem in static networks are mostly studied as missing link recovery that attempts to recover unknown links based on the existing links. Link prediction problem is extended to dynamic networks recent years that is to inferring future links according to history links, which is also defined as temporal link prediction problem [22], [23]. The different emerging time of links in dynamic networks, makes the link prediction problem in dynamic networks is more complex than it is in traditional static networks.

The basic idea to evaluate the probability that two nodes have links is to measure the similarity or closeness between them. The similarity of nodes are mostly evaluated according to the network structures around the nodes, such as CN, PA etc. These similarity evaluation features can be applied in unsupervised ranking model to select the top links with higher feature values as predicting the links will emerge [11], or can also constitute feature vectors and then be applied into classification models. Lü et al. proposed local-path index to characterize the node similarity based on the reachable paths between two nodes [8]. Zhao et al. proposed several reliable-route-based methods that introduce link weights into similarity [9], making traditional methods like CN and RA possible to deal with multiple links between nodes. Node evolution theory is studied in [10] to evaluate the emerging probability integrating both evolution perspectives to the link emergence from two end nodes, and propose that links with different surrounding structures should be applied with different similarity evaluation feature.

There are also link prediction methods based on non-negative matrix factorization which map the nodes into latent feature space by factorizing the adjacency matrix of networks into two latent feature matrices, and the predicted new network is derived from the production of the latent feature matrices [24]–[26]. Researchers make an assumption of consistency in dynamic networks that dynamic networks transform smoothly over time [27], [28]. Since the consistency can be maintained through constrain factors in factorization, matrix factorization based link prediction methods are widely applied in dynamic network [28]. Yu et al. leverage the time-dependent matrix factorization method which naturally expresses the evolving network by learning a low rank representation of the underlying adjacency matrix [28]. Gao et al. study the link direction prediction problem and propose a latent matrix factorization

method [23]. Yu et al. consider the future networks are the function of time and introduce evolutionary network analysis into link prediction that considers the network structure as a function of time and is the first to study the link weight prediction [29]. Graphlet transition based features are proposed by [30] to form low-dimensional features of node pairs.

Deep learning models are introduced into link prediction recently. Li et al. propose a framework based on boltzmann machine which predicts links based on individual transition variance and influence introduced by local neighbors [31]. Wang et al. design a hierarchical Bayesian model to jointly model high-dimensional node attributes and link structures [32]. Cukierski et al. utilize random forest to separate real links from fake links [33]. Neural machine is utilized to automatically learn latent information from input feature vector [14]. A multi-neural-network framework is propose for link prediction over aligned networks that alleviates cold start problem in social networks [34]. Ozcan et al. propose Nonlinear Autoregressive Neural Network (NARX) to study link prediction in evolving heterogeneous networks addressing the challenge of different link types, different effects of various similarity measures and nonlinear temporal evolution information [35].

III. Problem Formalization

A dynamic network in a time period $[t_0, t_n]$ contains continuously emerging links with timestamps. In other words, the links with timestamps emerge as a stream. We create the dynamic network from a blank graph and keep adding links and the end nodes into the graph from the start of the stream. Finally the dynamic network in time period $[t_0, t_n]$ can be represented by the graph as shown in Figure 2. In the graph, each link is annotated by a timestamp to record the emerging time, and multiple links are also allowed between nodes.

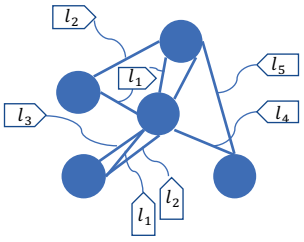


Fig. 2: An example of dynamic network

Definition 1: (Dynamic Network): A dynamic network is a graph $G = (V, E, L)$. $V = \{n_1, n_2, ..., n_m\}$ is the set of nodes in the network; $L = \{l_1, l_2, ..., l_s\}$ is the set of different timestamps in the network; $E = \{e_1, e_2, ..., e_k\}$ is the set of links, where $e_k = (n_i, n_j, l_k)$ is a triple denoting the link between node n_i and node n_j at timestamp $l_k \in L$.

Note that $e_i \in E$ and $e_j \in E$ may contain the same elements in the situation that two nodes created multiple links at one timestamp. We further define a period of dynamic network $G_{(t_p,t_q)}$ in period $[t_p, t_q)$, in which all the timestamps in $G_{(t_p,t_q)}$ are within $[t_p, t_q)$.

Definition 2: (Link Prediction Problem): Given a period of dynamic network $G_{(t_p,t_q)} = (V, E', L')$ and a link set E^p, where $L' = \{l_k | l_k \in L; t_p \leq l_k < t_q\}$, $E' = \{e_k | e_k = (n_i, n_j, l_k); e_k \in E; n_i, n_j \in V; l_k \in L'\}$ and $E^p = \{e_t | e_t = (n_a, n_b, l_t); n_a, n_b \in V; l_t = t_q\}$, the link prediction problem is to predict whether a target link $e_t \in E^p$ will emerge.

In this paper, in order to evaluate the link prediction results, we set $l_1 \leq t_p < t_q = l_t < l_s$ and $G_{(t_p,t_q)}$ actually is a portion of G, so that E contains the truth of whether e_t emerges or not.

IV. Structure Subgraph

In this section, we propose h-hop structure subgraph that represents the surrounding network structures of a target link e_t, then K-structure subgraph is proposed to encode the topology within K different structure nodes selected from h-hop structure subgraph and will be represented as structure subgraph feature in next section.

A. H-hop Structure Subgraph

Preliminarily the distance from a node n_i to a target link e_t is defined as:

$$d(n_i, e_t) = min(|\mathcal{P}(n_i, n_a)|, |\mathcal{P}(n_i, n_b)|) \quad (1)$$

where n_a and n_b are two end nodes of link e_t, $\mathcal{P}(\cdot, \cdot)$ denotes the shortest path between two nodes, and $|\mathcal{P}(\cdot, \cdot)|$ is the length of the path.

We formally define the surrounding subgraph of a target link e_t within h hop as the h-hop subgraph.

Definition 3: (h-hop Subgraph): Given a graph $G_{(t_p,t_q)} = (V, E', L')$, the h-hop subgraph of a given target link e_t is defined as $G_{h \to e_t} = (V_h, E_h, L')$, where $V_h = \{n_i | n_i \in V; d(n_i, e_t) \leq h\}$, $E_h = \{e_k | e_k = (n_i, n_j, l_k); e_k \in E'; n_i, n_j \in V_h; l_k \in L'\}$.

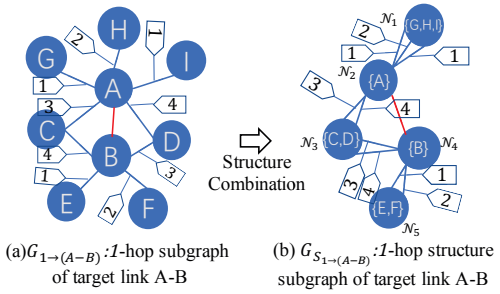
In a h-hop subgraph, the nodes with the same structures play same topological roles in the network and hence make the same impact on the emergence of the target link. We combine the nodes with the same structure into one structure node and continue combining the structure nodes until there are no structure nodes with the same structure in the structure subgraph. To clearly define the structure subgraph, we first define the structure nodes as follows.

Definition 4: (Structure Node): Given a h-hop subgraph $G_{h \to e_t} = (V_h, E_h, L)$, and the sets of neighbor nodes of two nodes $n_i \in V_h$ and $n_j \in V_h$ are denoted as Γ_{n_i} and Γ_{n_j}, respectively, then n_i and n_j have the same structure iff $\Gamma_{n_i} = \Gamma_{n_j}$. A structure node $\mathcal{N}_x$ is defined as a set collecting all the nodes that have the same structure in $G_{h \to e_t}$. Specially, the two end nodes n_a and n_b of the target link e_t are special structure nodes that only contain themselves.

For example, Figure 3(a) presents the 1-hop subgraph of link A-B, we can learn that the nodes G, H and I have the same structure because $\Gamma_G = \Gamma_H = \Gamma_I = \{A\}$. Thus, G, H and I are combined into one structure node $\mathcal{N}_1 = \{G, H, I\}$ in Figure 3(b).

Algorithm 1 Structure Combination Algorithm

Input: $G_{h\to e_t}$
Output: $G_{S_{h\to e_t}}$
1: initialization step $t \leftarrow 0$
2: $G^{t+1}_{S_{h\to e_t}} = (V^{t+1}_S, E^{t+1}_S, L) \leftarrow G_{h\to e_t}$
3: create empty structure subgraph: $G_{S^t_{h\to e_t}} = (V^t_S, E^t_S, L)$
4: **while** $G^t_{S_{h\to e_t}} \neq G^{t+1}_{S_{h\to e_t}}$ **do**
5: $\quad G^t_{S_{h\to e_t}} \leftarrow G^{t+1}_{S_{h\to e_t}}$
6: $\quad$ reset $G^{t+1}_{S_{h\to e_t}}$ to empty
7: $\quad$ **for** n_i in V^t_S **do**
8: $\quad\quad$ **if** $\Gamma_{n_i} = \Gamma_{n_j}$ $(n_j \in \mathcal{N}_x)$ **then**
9: $\quad\quad\quad$ add n_i into $\mathcal{N}_x$
10: $\quad\quad$ **else**
11: $\quad\quad\quad$ create $\mathcal{N}_x = \{n_i\}$
12: $\quad\quad\quad$ add $\mathcal{N}_x$ into V^{t+1}_S
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad$ update E^{t+1}_S
16: **end while**
17: $G_{S_{h\to e_t}} \leftarrow G^{t+1}_{S_{h\to e_t}}$
18: **return** $G_{S_{h\to e_t}}$



(a)$G_{1\to(A-B)}$:1-hop subgraph of target link A-B

(b) $G_{S_{1\to(A-B)}}$:1-hop structure subgraph of target link A-B

Fig. 3: Simple example of subgraph and structure subgraph

Definition 5: (Structure Link): A structure link between two structure nodes $\mathcal{N}_x$ and $\mathcal{N}_y$ is defined as a triple $\mathcal{E} = (\mathcal{N}_x, \mathcal{N}_y, E_k)$, where $E_k = \{e_k | e_k = (n_i, n_j, l_k); e_k \in E_h, n_i \in \mathcal{N}_x; n_j \in \mathcal{N}_y; l_k \in L'\}$. $\mathcal{E}$ represents all the links connecting the nodes in $\mathcal{N}_x$ and $\mathcal{N}_y$.

Definition 6: (h-hop Structure Subgraph): Given a h-hop subgraph $G_{h\to e_t}$ of a target link e_t, the h-hop structure subgraph of e_t is defined as $G_{S_{h\to e_t}} = (V_S, E_S, L')$, where $V_S = \{\mathcal{N}_1, \mathcal{N}_1, .., \mathcal{N}_p\}$; $E_S = \{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_q\}$; L is the set of all timestamps.

Figure 3(b) presents the 1-hop structure subgraph of link A-B, which is derived by combining the structures in 1-hop subgraph. Algorithm 1 presents the details of structure combination algorithm, which takes a h-hop subgraph as an input and figure out a h-hop structure subgraph as an output.

In Algorithm 1, iterating all the nodes (Line 7 to 14) takes maximum time of $O(|V|^2)$, and updating link set (Line 15) takes maximum time $O(|E|)$. Thus the time complexity of each inner loop (Line 7 to 15) is $O(|E| + |V|^2)$. Proof by contradiction can easily demonstrate the loop (Line 4 to 16) actually need one step to converge. Therefore, the time complexity of Algorithm 1 is $O(|E| + |V|^2) = O(|V|^2)$.

The size of input of Algorithm 1 determines the computation cost. Since the input $G_{h\to e_t}$ is a h-hop subgraph which usually contains nodes and links much less than $|E|$ and $|V|$, it barely reaches the worst case time complexity $O(|E| + |V|^2)$.

Algorithm 1 can ensure all topological structures in a subgraph conserved in corresponding structure subgraph. Therefore the h-hop structure subgraph is an equivalent representation of h-hop surrounding subgraph. The ability of structure subgraph to conserve all topologies with much less nodes make it possible to design a feature with much more efficient manner to encode various topological information. Structure subgraph also provides a novel perspective to the network structure. From structure subgraphs, we can easily observe what kinds of roles the nodes play around the target link, which is not only useful in link prediction, but also meaningful in other areas like social analysis and entity resolution.

B. K-Structure Subgraph

Although h-hop structure subgraph can be represented by adjacency matrix as the link feature of e_t, the sizes of h-hop structure subgraphs of links are usually different, causing link features are represented in different length. To uniformly extract features of the same size, we derive the K-structure subgraph from the h-hop structure subgraphs. The h-hop structure subgraphs of all target links are required to contain at least K structure nodes, so that for each target link e_t, we can select K structure nodes from its h-hop structure subgraph and construct a structure subgraph with these K structure nodes, whose adjacency matrix is of uniform size of $K \times K$ and utilized to calculate the *S*tructure *S*ubgraph *F*eature (SSF) of e_t.

Initially, $h = 1$, we derive 1-hop structure subgraph $G_{S_{1\to e_t}}$ from 1-hop subgraph $G_{1\to e_t}$ through Algorithm 1. If the number of structure nodes of $G_{S_{1\to e_t}}$ is less than K, namely $|V_S| < K$, then we continue increasing h to include more nodes with different structure into h-hop subgraph $G_{h\to e_t}$ and repeat extracting h-hop structure subgraph $G_{S_{h\to e_t}}$ from $G_{h\to e_t}$, until the number of structure nodes of $G_{S_{h\to e_t}}$ satisfies $|V_S| \geq K$.

Next, we assign orders to all the structure nodes in $G_{S_{h\to e_t}}$. The order of structure nodes determines which structure nodes are selected. Note that the structure nodes that contain the end nodes of a target link e_t must be selected. In this paper, the Palette-WL algorithm [14] is adopted to ensure the order numbers of the two end nodes of e_t always be 1 and 2, and structure nodes that farther to e_t will have higher order numbers. The detail of the algorithm is shown in Algorithm 2. The order of a structure node $\mathcal{N}$ is denoted as $C(\mathcal{N})$, so $C(\mathcal{N}_x) = 1$ and $C(\mathcal{N}_y) = 2$ for $\mathcal{N}_x = \{n_a\}, N_y = \{n_b\}$, where n_a and n_b are the two end nodes of e_t.

The top K structure nodes and the structure links between them constitute the K-structure subgraph of a target link e_t which is defined as Definition 7.

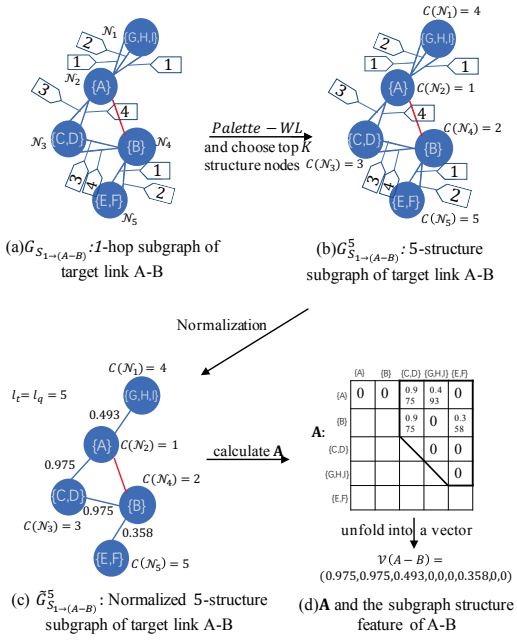


Fig. 4: Illustration of extracting structure subgraph feature of link A-B when $K = 5$

Definition 7: (K-structure subgraph): Given a h-hop structure subgraph $G_{S_{h\to e_t}} = (V_S, E_S, L')$ that satisfies $|V_S| \geq K$, the K-structure subgraph is a part of $G_{S_{h\to e_t}}$ that contains K structure nodes and formally defined as $G^K_{S_{h\to e_t}} = (V^K_S, E^K_S, L')$. $V^K_S = \{\mathcal{N}_x | \mathcal{N}_x \in V_S; C(\mathcal{N}_x) \leq K\}$ and $E^K_S = \{\mathcal{E}_k | \mathcal{E}_k = (\mathcal{N}_x, \mathcal{N}_y, E_k); \mathcal{E}_k \in E_S; \mathcal{N}_x, \mathcal{N}_y \in V^K_S\}$ where $E_k = \{e_k | e_k = (n_i, n_j, l_k); n_i \in \mathcal{N}_x; n_j \in \mathcal{N}_y; l_k \in L'\}$.

In Figure 4(a), when $K = 5$, since $G_{S_{1\to(A-B)}}$ exactly contains 5 structure nodes, all the structure nodes are ordered and selected to construct the 5-structure subgraph $G^5_{S_{1\to(A-B)}}$ as shown in Figure 4(b), which is the same as $G_{S_{1\to(A-B)}}$ in this simple example.

Selecting only K nodes from the h-hop structure subgraph may cause information lost, however in real application, K is usually set not less than 10, which is sufficient for differentiating links at most case. The experiments in this paper also demonstrate that the best performance of link prediction on real-world dynamic networks falls around $K = 10$, which indicates that it is not necessary to encode all topological information of h-hop structure subgraph into link feature.

V. Structure Subgraph Feature

The K-structure subgraph cannot be directly represented as normal adjacency matrix, because the there are multiple links with different timestamps between structure nodes. The different number of history links between two structure nodes and the different emerging time of these links make the relation between the two structure nodes have different influence on the emergence of new links. We elaborately design an adjacency matrix that specifies the influence of multiple links and different emerging time in K-structure subgraph $G^K_{S_{h\to e_t}}$, and propose *S*tructure *S*ubgraph *F*eature (SSF) that is effective in dynamic networks.

Algorithm 2 The Palette-WL Algorithm

Input: $G_{S_{h\to e_t}}, e_t$
Output: C : the order list for all $\mathcal{N}_x \in V_S$

initialize the order of all $\mathcal{N}_x \in V_S$ increasingly with the distance to e_t.
while $C(\mathcal{N}_i)$ is not converged **do**
  for $\mathcal{N}_x$ in V_S **do**
    $h(\mathcal{N}_x) \leftarrow C(\mathcal{N}_x) + \frac{\sum_{\mathcal{N}_p \in \Gamma_{\mathcal{N}_x}} log(P(C(\mathcal{N}_p)))}{\lceil \sum_{\mathcal{N}_q \in V_S} log(P(C(\mathcal{N}_q))) \rceil}$
    (where $P(n)$ is the n^{th} prime number)
  end for
  rank the node by $h(\mathcal{N}_x)$ in ascending order
  $C(\mathcal{N}_x) \leftarrow$ the order of $\mathcal{N}_x$
end while
return C

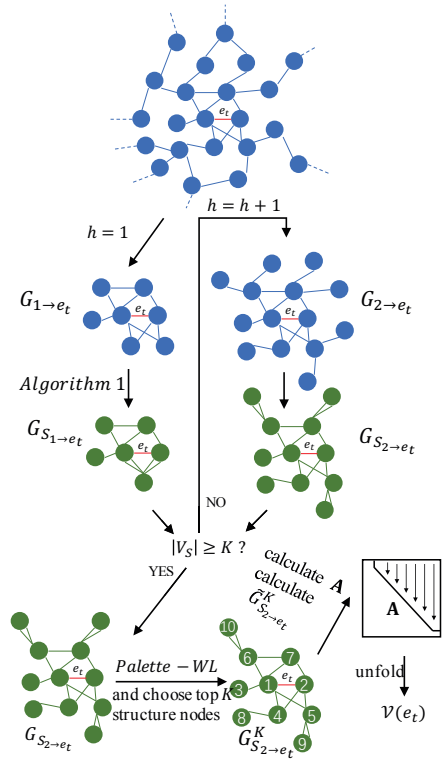


Fig. 5: Illustration of the whole process of SSF feature extraction ($K = 10$)

A. Normalized K-structure Subgraph

Since e_t is a future link to be predicted, we uniformly set the timestamp of e_t as $l_t = t_q$. The influences of history links decay with the time since links in farther historical time will have less impact on the emergences of links at present time. According to [28], the remaining influence $f(l_t, l_s)$ of a

history link $e = (n_i, n_j, l_s)$ at present time l_t can be calculated according to their timestamps as:

$$f(l_t, l_s) = exp^{-\theta(l_t - l_s)}, \tag{2}$$

where $\theta \in (0, 1)$ is a damping factor to control the speed of decay. In this paper, we uniformly set $\theta = 0.5$ to obtain an average performance [28].

Since the nodes contained in a same structure node play the same role in the network topology and influence the emergence of target link e_t in the same way, the influence of all the links between two structure nodes can be integrated as a normalized influence of one link. For a structure link $(\mathcal{N}_x, \mathcal{N}_y, E_k) \in E_S^K$ in a K-structure subgraph $G_{S_{h\to e_t}}^K = (V_S^K, E_S^K, L')$, $E_k = \{e_k | e_k = (n_i, n_j, l_k); n_i \in \mathcal{N}_x; n_j \in \mathcal{N}_y; l_k \in L'\}$ containing all links between the nodes in $\mathcal{N}_x$ and $\mathcal{N}_y$ can be normalized to $\widetilde{E}_k = \{(\widetilde{n}_i, \widetilde{n}_j, \widetilde{l}_k)\}$ which only contains one link. $\widetilde{n}_i$ and $\widetilde{n}_j$ are two arbitrary nodes in $\mathcal{N}_x$ and $\mathcal{N}_y$, respectively. $\widetilde{l}_k$ is the normalized influence of all $e_k \in E_k$, which is defined as Definition 8.

Definition 8: (Normalized Influence:) Given a link set $E_k = \{e_k | e_k = (n_i, n_j, l_k); n_i \in \mathcal{N}_x; n_j \in \mathcal{N}_y; l_k \in L'\}$ where $\mathcal{N}_x, \mathcal{N}_y \in V_S^K$, $\widetilde{l}_k$ is the normalized influence of all the links $e_k \in E_k$ which is calculated as:

$$\widetilde{l}_k = \sum_{(n_i, n_j, l_k) \in E_k} f(l_t, l_k) \quad = \sum_{(n_i, n_j, l_k) \in E_k} exp^{-\theta(l_t - l_k)}. \tag{3}$$

When all the links between arbitrary two structure nodes in K-structure subgraph $G_{S_{h\to e_t}}^K$ are normalized to one link, the original $G_{S_{h\to e_t}}^K$ is transformed into a normalized K-structure Subgraph which is defined as Definition 9.

Definition 9: (Normalized K-structure Subgraph): Given a K-structure subgraph $G_{S_{h\to e_t}}^K = (V_S^K, E_S^K, L')$, the normalized K-structure subgraph is defined as $\widetilde{G}_{S_{h\to e_t}}^K = (\widetilde{V}_S^K, \widetilde{E}_S^K, \widetilde{L}')$. $\widetilde{V}_S^K = V_S^K$, $\widetilde{E}_S^K = \{\widetilde{\mathcal{E}}_k | \widetilde{\mathcal{E}}_k = (\mathcal{N}_x, \mathcal{N}_y, \widetilde{E}_k); \mathcal{N}_x, \mathcal{N}_y \in \widetilde{V}_S^K\}$ where $(\mathcal{N}_x, \mathcal{N}_y, \widetilde{E}_k)$ is derived from $(\mathcal{N}_x, \mathcal{N}_y, E_k)$ by normalizing E_k to $\widetilde{E}_k = \{(\widetilde{n}_i, \widetilde{n}_j, \widetilde{l}_k)\}$. $\widetilde{L}'$ is a set collecting all $\widetilde{l}_k$.

Figure 4(c) illustrates the normalized 5-structure subgraph of link $A - B$.

B. Structure Subgraph Feature Representation

Let $\mathbf{A}$ be the the adjacency matrix of a normalized K-structure subgraph $\widetilde{G}_{S_{h\to e_t}}^K$. For $C(\mathcal{N}_x) = m$ and $C(\mathcal{N}_y) = n$, the entry of $\mathbf{A}(m, n)$ $(m, n \in \mathbb{N}; m, n \leq K)$ is the normalized influence of structure link $(\mathcal{N}_x, \mathcal{N}_y, \widetilde{E}_k)$, which is calculated by (4).

$$\mathbf{A}(m, n) = \begin{cases} \widetilde{l}_k, & \text{if } \widetilde{E}_k = \{(\widetilde{n}_i, \widetilde{n}_j, \widetilde{l}_k)\} \neq \emptyset; \\ 0, & \text{if } \widetilde{E}_k = \emptyset. \end{cases} \tag{4}$$

Since the target link e_t is to be predicted, $\mathbf{A}(1, 2)$ and $\mathbf{A}(2, 1)$ are unknown and uniformly set as 0. $\mathbf{A}$ is symmetrical because the dynamic networks in this paper are undirected graphs. Thus, we can define the *S*tructure *S*ubgraph *F*eature (SSF) of e_t by unfolding the upper right half of $\mathbf{A}$ by column. [1]

Algorithm 3 SSF Extraction Algorithm

Input: G, e_t, K
Output: $\mathcal{V}(e_t)$
1: initialize $h \leftarrow 0$, $G_{h\to e_t}, G_{S_{h\to e_t}}$
2: **while** $|V_S| \leq K$ **do**
3: $h \leftarrow h + 1$
4: construct $G_{h\to e_t}$
5: $G_{S_{h\to e_t}} \leftarrow$ *Algorithm 1* $(G_{h\to e_t})$
6: **end while**
7: C = *Algorithm 2* $(G_{S_{h\to e_t}}, e_t)$
8: extract K-structure subgraph $G_{S_{h\to e_t}}^K$
9: construct $\widetilde{G}_{S_{h\to e_t}}^K$
10: calculate adjacency matrix $\mathbf{A}$ by (4)
11: $\mathcal{V}(e_t) \leftarrow$ unfold the upper right half of $\mathbf{A}$ by (5)
12: **return** $\mathcal{V}(e_t)$

Definition 10: (Structure Subgraph Feature (SSF)): Given an adjacency matrix $\mathbf{A}$ of normalized K-structure subgraph $\widetilde{G}_{S_{h\to e_t}}^K$, the *S*tructure *S*ubgraph *F*eature (SSF) of a target link e_t is a vector $\mathcal{V}(e_t)$ derived from the upper right half of $\mathbf{A}$, which is calculated as :

$$\mathcal{V}(e_t) = conn(\mathbf{A}(m, n)); 3 \leq n < K, 1 \leq m < n, \tag{5}$$

where $conn(\cdot)$ means connecting the elements as a vector.

Figure 4(d) shows the adjacency matrix of normalized 5-structure subgraph and the SSF of link $A - B$.

In addition, we can relax the entries of $\mathbf{A}$ in real applications and let them encode other information to further increase the flexibility SSF. In the experiments of this paper, for the structure link $(\mathcal{N}_x, \mathcal{N}_y, \widetilde{E}_k)$ that $C(\mathcal{N}_x) = m$ and $C(\mathcal{N}_y) = n$, we set

$$\mathbf{A}(m, n) = 1/(min(d(\mathcal{N}_x, e_t), d(\mathcal{N}_y, e_t))),$$

where $d(\mathcal{N}_x, e_t)$ is the length of shortest path[1] from $\mathcal{N}_x$ to e_t in $\widetilde{G}_{S_{h\to e_t}}^K$.

Figure 5 presents an example of the whole process of extracting SSF of e_t at $K = 10$ from the dynamic network $G_{(t_p, t_q)}$. Here we ignore the multiple links between nodes and timestamps for legibility. The extraction starts from $h = 1$ and continues extracting $G_{S_{h\to e_t}}$ until $|V_S| \geq K$. Then, the Palette-WL Algorithm is applied to assign unique orders to each structure node in $G_{S_{h\to e_t}}$. Next, $G_{S_{h\to e_t}}^K$ and $\widetilde{G}_{S_{h\to e_t}}^K$ are constructed. Finally, the adjacency matrix $\mathbf{A}$ and the SSF of e_t, $\mathcal{V}(e_t)$ is calculated according to (4) and (5).

The detailed process of SSF extraction is illustrated in Algorithm 3. Iteratively combining the subgraph (Line 2 to 6) costs average time of $O(K(|V|^2)$. The time complexity of the Palette-WL algorithm (Line 7) is $O(K^3)$ [14]. And the time complexity of computing the normalized K-structure subgraph (Line 9) and the adjacency matrix (Line 10) is $O(|E| + K^2)$. Therefore, the total time complexity of SSF extraction is $O(K(|V|^2) + K^3 + |E| + K^2) = O(K^3 + K(|V|^2))$.

[1] When calculating the shortest path, all $\widetilde{l}_k$ are set reciprocal.

TABLE II: Statistics of Datasets

Datasets	$\|V\|$	$\|E\|$	Avg. Degree	Time Span
Eu-Email	309	61046	395.12	803 h
Contact	274	28245	103.08	96 h
Facebook	4313	42346	19.63	366 d
Co-author	744	7034	18.90	20 y
Prosper	1264	8874	14.04	60 m
Slashdot	2680	9904	9.52	240 d
Digg	3215	9618	5.98	240 h

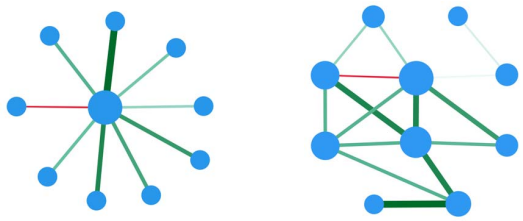
VI. Experiments

In this section, we conduct experiments to evaluate the effectiveness of SSF for link prediction tasks. Here we consider the link prediction problem as a classification problem, that is to classify the links that will be created in future or not.

A. Datasets Description

Eu-Email [36]: This network is generated using email data from a large European research institution. A node represents a institution member, and edges represent e-mail communications between institution members.
Contact [37]: This network represents contacts between people measured by carried wireless devices. A node represents a person, and edges between two persons shows that there were contacts between them.
Facebook [38]: This network is generated from a small subset of posts to other user's wall on Facebook. The nodes of the network are Facebook users, and each directed edge represents one post, linking the users writing a post to the users whose wall the post is written on.
Co-author: This network is generated from a subset of DBLP [39], representing co-author relationships between researchers. A node is an author, and edges between two nodes represent the authors have published papers together.
Prosper [40]: The data are loans between users of the Prosper.com website and This network is directed and denotes who loaned money to whom.
Slashdot [41]: This is the reply network of technology website Slashdot. Nodes are users and edges are replies.
Digg [42]: This is the reply network of the social news website Digg. Each node in the network is a user of the website, and each directed edge denotes that a user replied to another user.

All these networks are dynamic where edges are annotated with timestamps showing the emerging time. In this paper, we ignore the direction of edges, since we only predict if there will be a link between two nodes. Table II presents the detail of these datasets. "Time Span" is the length of the period of dynamic networks, specifically, 'h', 'd', 'm' and 'y' stand for 'hour', 'day', 'month' and 'year' respectively. The number of different timestamps of these networks are normalized according to the time span. For example, the Eu-email network spans 803 hours, thus we annotate 803 different timestamps to the links ranging $[1, 803]$.



(a) Facebook network (b) Co-author network

Fig. 6: The most frequent K-structure subgraph patterns in two dynamic networks when $K = 10$

B. K-Structure Subgraph Visualization

To intuitively understand how effective the structure subgraph represents the surrounding network structures of links, here we present two visualized K-structure subgraphs obtained from Facebook and Co-author networks. We say two K-structure subgraphs follow the same pattern when they have the same connection relations among structure nodes (multiple links between them are ignored). We randomly choose 2,000 links from the two dynamic networks respectively and extract the $G^K_{S_{h\to e}}$ of the selected links with $K = 10$. Then we select the most frequent pattern of $G^K_{S_{h\to e}}$ as shown in Figure 6. The blue nodes are structure nodes, the red link is the target link and the green links are structure links. The thickness of each structure link is associated with the average number of links that the structure link combines in all the K structure subgraphs that follow the same pattern. The size of a structure nodes is related to the degree.

Figure 6(a) shows the the most frequent pattern of K-structure subgraphs in Facebook network. Note that although the neighbor nodes seem have the same structure which is against to the definition of K-structure subgraph, they are actually not the same in $G_{S_{1\to e_t}}$ because only K structure nodes are selected from $G_{S_{1\to e_t}}$. The structure subgraph pattern in Figure 6(a) shows that links are formed with nodes with high degree. Since the links in the network represents the replies to facebook posts, the pattern actually indicates the fact that users in the Facebook network often write posts to the walls of famous people who usually receive tremendous number of replies.

Figure 6(b) shows the most frequent pattern of K-structure subgraphs in Co-author network. This pattern of K-structure subgraph is dense and the structure nodes are well connected to each other, indicating that coauthor relationship are usually formed in small research groups. More specifically, the pattern can reflect the fact that the scholars create coauthor relationship with the scholars who have common coauthors and with famous scholars who has a large number of co-author relationships.

Although network structure of Facebook network and Co-author network are obviously different, our proposed K-structure subgraph can automatically capture meaningful features from surrounding network topologies. Therefore struc-

ture subgraph feature can encode useful topological information and be consistently effective in various networks.

C. Experimental Evaluation and Discussions

In this section, we create two link prediction methods by applying SSF to a linear regression model and an neural machine classification model, namely SSFLR and SSFNM. We compare the two methods on link prediction tasks with 11 baseline methods (including two variant versions SSFLR-W and SSFNM-W) on 7 real-world dynamic networks.

1) Link Prediction Methods: Above mentioned CN [7], PA [15] , Jaccard (Jac.) [16], AA [1] , RA [17], rWRA [9], Random Walk (RW) [18], Katz [19] and WLF-based methods [14] are all adopted as baselines. The detail of link prediction methods are described as follows.

CN, PA, Jaccard (Jac.), AA, RA, rWRA, Random Walk (RW) and Katz: the unsupervised ranking models using corresponding features;

NMF [24] : Non-negative matrix factorization method for link prediction;

WLLR [14]: the linear regression model adopting WLF;

WLNM [14]: the neural machine classification model adopting WLF;

SSFLR: the linear regression model adopting SSF;

SSFLR-W: the variant version of SSFLR that treats dynamic networks as static networks. SSFLR-W adopt SSF-W, which is SSF without considering timestamps by replacing $\mathbf{A}(m, n)$ with common $0/k$ entries, where 0 or k represents there are 0 or k links between structure nodes;

SSFNM: the neural machine classification model adopting SSF;

SSFNM-W: the neural machine classification model adopting SSF-W.

2) Experimental Settings: For supervised learning methods like WLLR, WLNM, SSFLR, SSFNM,SSFLR-W and SSFNM-W, training phrase is required for link prediction. We choose the last timestamp of the dynamic networks as the present time l_t, then select 70 percent of the real links at l_t as positive samples for training, and the remaining links are selected as positive samples for test. We randomly select fake links as negative samples and set them have the same number as positive samples in both training set and test set. For the methods that proposed in static networks, we ignore all the timestamps and multiple history links between nodes to construct the static version of the 7 datasets. For unsupervised ranking models, we treat the training set as prior knowledge to decide the threshold for classifying links based on their feature value.

For NMF, we use the static version of the dynamic networks at $[1, l_t - 1]$ as the history network and use NMF to directly predict the adjacency matrix of networks at l_t. The β in Katz is 0.001 and the weights of links for rWRA are set as the number of history links between two nodes. The neural machine for SSFNM and WLNM has three fully-connected hidden layers with 32, 32, 16 neurons activated by ReLu and a softmax layer as the output layer. The mini batch size is 10, epoch is 2000, and learning rate is 0.001.

We utilize two popular evaluation metrics in classification tasks, AUC and F1 score, to evaluate the performance of link prediction. For both of the two metrics, higher values indicate better link prediction performance.

3) Link Prediction Results: Table III presents the results of link prediction of SSF-based methods and the baseline methods. Here we set $K = 10$ for both WLF-based methods and SSF-based methods. We will further study the performance of SSFNM with different K in the next subsection. Most of the best values fall on SSFLR and SSFNM, which demonstrates the superiority of SSF.

Among all link prediction methods that ignore the timestamps in dynamic networks, only the methods based on WLF and SSF (WLNM, SSFLR-W and SSFNM-W) have relatively consistent performance on all the 7 dynamic networks, while others are only effective on several datasets. The reason is that, instead of utilizing one or two types of topological information, WLF and SSF can utilize all the topological information encoded in the surrounding K nodes and K structure nodes, which makes them adaptive to various networks.

Although SSFLR-W, WLLR, SSFNM-W and WLNM all do not specify the influence of timestamps in dynamic networks, SSFLR-W and SSFNM-W are based on SSF-W, which utilize structure subgraph that combines nodes into structure nodes, while WLLR and WLNM are based on WLF which traditionally adopt enclosing subgraph containing normal nodes. From table III, we can observe that SSFLR-W surpass WLLR on most datasets. Due to the deficiency of linear regression model of learning high-dimensional latent patterns, SSFLR-W can not make full use of SSF-W and thus can not consistently outperform WLLR. On the other hand, benefiting from the ability of neural machine of learning latent patterns, SSFNM-W outperforms WLNM on all datasets. This demonstrates the great effect of structure subgraph. Combining nodes with the same topological structure makes structure subgraph much more effective to represent network topologies and ensures SSF-W and SSF can encode plentiful multiple types of topological information.

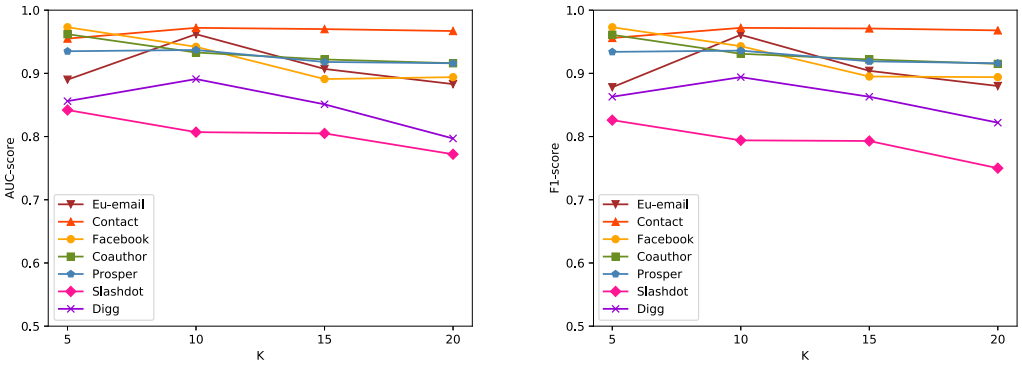
By specifying the influence of multiple links and different timestamps with normalized influences in dynamic works, SSFLR and SSFNM achieve even better performance than SSFLR-W and SSFNM-W on almost all dynamic networks (only except SSFNM on Slashdot dataset and SSFLR on Digg dataset). The results indicate that it is necessary to consider the influence of multiple links and different emerge time under the context of dynamic networks. The proposed normalized influence makes sense for dynamic networks and promotes the performance of link prediction in most situations.

Based on all above results, we can conclude that the SSF-based methods outperform the baseline methods and provide consistently top-class performance on various dynamic networks.

4) Influence of K: We study the influence of K on the link prediction performance of SSF-based methods. Figure 7

TABLE III: Results of Link Prediction on 7 Datasets

Methods \ Datasets	Eu-email		Contact		Facebook		Coauthor		Prosper		Slashdot		Digg	
	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1
CN	0.903	0.901	0.802	0.802	0.694	0.604	0.681	0.550	0.469	0.100	0.616	0.413	0.562	0.230
Jac.	0.904	0.904	0.804	0.804	0.695	0.606	0.682	0.546	0.496	0.091	0.617	0.411	0.562	0.228
PA	0.695	0.693	0.803	0.803	0.303	0.302	0.416	0.410	0.294	0.290	0.677	0.674	0.713	0.711
AA	0.923	0.927	0.804	0.804	0.695	0.606	0.680	0.550	0.469	0.090	0.616	0.413	0.562	0.229
RA	0.920	0.920	0.807	0.808	0.694	0.605	0.689	0.550	0.469	0.100	0.615	0.412	0.562	0.229
rWRA	0.923	0.923	0.803	0.804	0.694	0.606	0.681	0.551	0.469	0.095	0.614	0.409	0.562	0.230
Katz	0.952	0.952	0.802	0.802	0.837	0.837	0.630	0.631	0.286	0.287	0.686	0.687	0.697	0.698
RW	0.940	0.941	0.870	0.852	0.827	0.769	0.820	0.783	0.716	0.662	0.802	0.708	0.641	0.454
NMF	0.774	0.856	0.557	0.660	0.641	0.639	0.584	0.761	0.632	0.757	0.798	**0.858**	0.697	0.799
WLLR	0.914	0.842	0.711	0.595	0.767	0.629	0.816	0.679	0.737	0.680	0.705	0.752	0.737	0.757
SSFLR-W	0.900	0.798	0.876	0.784	0.934	0.889	0.857	0.778	0.726	0.662	0.821	0.679	0.825	0.784
WLNM	0.896	0.896	0.727	0.660	0.790	0.740	0.883	0.881	0.786	0.790	0.807	0.832	0.881	0.874
SSFNM-W	0.925	0.921	0.853	0.867	0.933	0.929	0.861	0.854	0.802	0.804	**0.853**	0.843	0.886	0.874
SSFLR	0.937	0.812	**0.984**	0.820	0.855	0.830	0.911	0.827	**0.973**	0.792	0.832	0.531	0.683	0.531
SSFNM	**0.962**	**0.961**	0.972	**0.972**	**0.942**	**0.943**	**0.933**	**0.931**	0.937	**0.936**	0.831	0.821	**0.891**	**0.894**



(a) The AUC scores of SSFNM with different K

(b) The F1 scores of SSFNM with different K

Fig. 7: AUC and F1 scores of SSFNM with different K on different datasets

shows the AUC scores and F1 scores of SSFNM on different datasets with K = 5, 10, 15 and 20, respectively. Although the peaks are different on different datasets, most peaks are achieved when $K \leq 15$, which indicates that we do not need a very large K to reach the best performance in real applications and select only K structure nodes for K-structure subgraph will not cause significant decrease of the performance on link prediction tasks. The reason is that there are noise data in real dynamic networks, e.g. missing links and false links, increasing K will introduce more noise data into link features, leading to the deficiency of methods for link prediction.

VII. Conclusions

In this paper, we studied the link prediction problem in dynamic networks and designed a universally applicable link prediction method for dynamic networks. We first proposed the structure subgraph which can efficiently represent the surrounding network structures of a target link. Next, we proposed the normalized influence to address the influence of multiple links between two nodes and different emerging time of links in dynamic networks. Then, we proposed the *S*tructure *S*ubgraph *F*eature (SSF), which is derived from the adjacency matrix of the normalized K-structure subgraph. The proposed K-structure subgraph can automatically encode useful surrounding topology of a target link and the manner of representing K-structure subgraph through adjacency matrix makes SSF can directly encode all topological information into a feature vector. Furthermore, the normalized influence makes SSF be effective to deal with influence of timestamps in dynamic networks. Finally, we proposed two link prediction methods by applying SSF to a linear regression model and a neural machine. Comparing with 11 baseline link prediction methods on 7 real-world dynamic networks, the experimental results demonstrate that the two SSF-based methods outperform the baseline methods and provide consistently top-class performance on link prediction tasks over various dynamic networks.

References

[1] L. A. Adamic, E. Adar, Friends and neighbors on the web, Social Networks 25 (2003) 211–230.

[2] Y. Koren, R. M. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42.

[3] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proceedings of the IEEE 104 (2016) 11–33.
[4] I. Bhattacharya, Collective entity resolution in relational data, TKDD 1 (2006) 5.
[5] V. E. Krebs, Mapping networks of terrorist cells, 2002.
[6] Y. Qi, Z. Bar-Joseph, J. Klein-Seetharaman, Evaluation of different biological data and computational classification methods for use in protein interaction prediction., Proteins 63 3 (2006) 490–500.
[7] D. Liben-Nowell, J. M. Kleinberg, The link prediction problem for social networks, JASIST 58 (2003) 1019–1031.
[8] L. Lü, C.-H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks., Physical review. E, Statistical, nonlinear, and soft matter physics 80 4 Pt 2 (2009) 046122.
[9] J. Zhao, L. Miao, H. Fang, Q. M. Zhang, M. Nie, T. Zhou, Predicting missing links and their weights via reliable-route-based method, Computer Science 5.
[10] H. Wang, W. Hu, Z. Qiu, B. Du, Nodes' evolution diversity and link prediction in social networks, IEEE Transactions on Knowledge and Data Engineering 29 (2017) 2263–2274.
[11] W. Liang, X. Li, X. He, X. Liu, X. Zhang, Supervised ranking framework for relationship prediction in heterogeneous information networks, Appl. Intell. 48 (5) (2018) 1111–1127. doi:10.1007/s10489-017-1044-7.
[12] Y. Sun, R. Barber, M. Gupta, C. C. Aggarwal, J. Han, Co-author relationship prediction in heterogeneous bibliographic networks, 2011 International Conference on Advances in Social Networks Analysis and Mining (2011) 121–128.
[13] C. Dai, L. Chen, B. Li, Y. Li, Link prediction in multi-relational networks based on relational similarity, Inf. Sci. 394 (2017) 198–216.
[14] M. Zhang, Y. Chen, Weisfeiler-lehman neural machine for link prediction, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, 2017, pp. 575–583. doi:10.1145/3097983.3097996.
[15] Barabási, R. Albert, Emergence of scaling in random networks, Science 286 5439 (1999) 509–12.
[16] P. Jaccard, The distribution of the flora in the alpine zone, New Phytologist 11 (2) (1912) 37–50.
[17] T. Zhou, L. L, Y. C. Zhang, Predicting missing links via local information, European Physical Journal B 71 (4) (2009) 623–630.
[18] W. Liu, L. Lu, Link prediction based on local random walk 89 (5) (2010) 58007–58012(6).
[19] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.
[20] L. L, T. Zhou, Link prediction in complex networks: A survey, Physica A Statistical Mechanics and Its Applications 390 (6) (2010) 1150–1170.
[21] V. Martínez, F. B. Galiano, J. C. Cubero, A survey of link prediction in complex networks, ACM Comput. Surv. 49 (2016) 69:1–69:33.
[22] D. M. Dunlavy, T. G. Kolda, E. Acar, Temporal link prediction using matrix and tensor factorizations, TKDD 5 (2011) 10:1–10:27.
[23] S. Gao, L. Denoyer, P. Gallinari, Temporal link prediction by integrating content and structure information, in: CIKM, 2011.
[24] C. J. Lin, Projected gradient methods for nonnegative matrix factorization, Neural Computation 19 (10) (2007) 2756–2779.
[25] A. K. Menon, C. Elkan, Link prediction via matrix factorization, in: ECML/PKDD, 2011.
[26] X. Ma, P. Sun, G. Qin, Nonnegative matrix factorization algorithms for link prediction in temporal networks using graph communicability, Pattern Recognition 71 (2017) 361–374. doi:10.1016/j.patcog.2017.06.025.
[27] L. Zhu, D. Guo, J. Yin, G. V. Steeg, A. Galstyan, Scalable temporal latent space inference for link prediction in dynamic social networks, IEEE Transactions on Knowledge and Data Engineering 28 (2016) 2765–2777.
[28] W. Yu, W. Cheng, C. C. Aggarwal, H. Chen, W. Wang, Link prediction with spatial and temporal consistency in dynamic networks, in: IJCAI, 2017.
[29] W. Yu, C. C. Aggarwal, W. Wang, Temporally factorized network modeling for evolutionary network analysis, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM 2017, Cambridge, United Kingdom, February 6-10, 2017, 2017, pp. 455–464.
[30] M. Rahman, M. A. Hasan, Link prediction in dynamic networks using graphlet, in: ECML/PKDD, 2016.
[31] X. Li, N. Du, H. Li, K. Li, J. Gao, A. Zhang, A deep learning approach to link prediction in dynamic networks, in: SDM, 2014.
[32] H. Wang, X. Shi, D.-Y. Yeung, Relational deep learning: A deep latent variable model for link prediction, in: AAAI, 2017.
[33] W. Cukierski, B. Hamner, B. Yang, Graph-based features for supervised link prediction, The 2011 International Joint Conference on Neural Networks (2011) 1237–1244.
[34] X. Cao, H. Chen, X. Wang, W. Zhang, Y. Yu, Neural link prediction over aligned networks, in: AAAI, 2018.
[35] A. Ozcan, S. G. Ögüdücü, Link prediction in evolving heterogeneous networks using the narx neural networks, Knowledge and Information Systems (2017) 1–28.
[36] H. Yin, A. R. Benson, J. Leskovec, D. F. Gleich, Local higher-order graph clustering, in: KDD, 2017.
[37] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on opportunistic forwarding algorithms, IEEE Trans. on Mobile Computing 6 (6) (2007) 606–620.
[38] B. Viswanath, A. Mislove, M. Cha, K. P. Gummadi, On the evolution of user interaction in Facebook, in: Proc. Workshop on Online Social Networks, 2009, pp. 37–42.
[39] J. Tang, J. Zhang, L. Yao, J.-Z. Li, R. Cui, Z. Su, Arnetminer: extraction and mining of academic social networks, in: KDD, 2008.
[40] J. Kunegis, KONECT – The Koblenz Network Collection, in: Proc. Int. Conf. on World Wide Web Companion, 2013, pp. 1343–1350.
[41] V. Gmez, A. Kaltenbrunner, V. Lpez, Statistical analysis of the social network and discussion threads in Slashdot, in: Proc. Int. World Wide Web Conf., 2008, pp. 645–654.
[42] M. D. Choudhury, H. Sundaram, A. John, D. D. Seligmann, Social synchrony: Predicting mimicry of user actions in online social media, in: Proc. Int. Conf. on Computational Science and Engineering, 2009, pp. 151–158.