Information Sciences 374 (2016) 1–14

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Sampling-based algorithm for link prediction in temporal networks

Nahla Mohamed Ahmed[a,b], Ling Chen[a,c,*], Yulong Wang[d], Bin Li[a,c], Yun Li[a], Wei Liu[a]

[a] *College of Information Engineering, Yangzhou University, Yangzhou China, 225009*
[b] *College of Mathematical Sciences, Khartoum University, Khartoum, Sudan*
[c] *State Key Lab of Novel Software Tech, Nanjing University, Nanjing China, 210093*
[d] *College of Agriculture, Yangzhou University, Yangzhou China, 225009*

ARTICLE INFO

Article history:
Received 18 February 2016
Revised 8 September 2016
Accepted 11 September 2016
Available online 12 September 2016

Keywords:
Link prediction
Random walk
Sampling
Temporal network

ABSTRACT

The problem of link prediction in temporal networks has attracted considerable recent attention from various domains, such as sociology, anthropology, information science, and computer science. In this paper, we propose a fast similarity-based method to predict the potential links in temporal networks. In this method, we first combine the snapshots of the temporal network into a weighted graph. A proper damping factor is used to assign greater importance to more recent snapshots. Then, we construct a sub-graph centered at each node in the weighted graph by a random walk from the node. The sub-graph constructed consists of a set of paths starting from the given node. Because the similarity score is computed within such small sub-graphs centered at each node, the algorithm can greatly reduce the computation time. By choosing a proper number of sampled paths, we can restrict the error of the estimated similarities within a given threshold. While other random walk-based algorithms require $O(n^3)$ time for a network with n nodes, the computation time of our algorithm is $O(n^2)$, which is the lowest time complexity of a similarity-based link prediction algorithm. Moreover, because the proposed method integrates temporal and global topological information in the network, it can yield more accurate results. The experimental results on real networks show that our algorithm demonstrates the best or comparable quality results in less time than other methods.

1. Introduction

Networks can naturally describe various social structures. In such networks, vertices denote individuals, while the edges represent relations among the individuals, such as corporations or companionship. Social network analysis has drawn increasing attention in the fields of sociology, computer science, and physics. It analyzes and explores the potential relations between social objects. Recently, complex network analysis has also drawn much attention in many commerce fields, such as e-business analysis and market modeling.

One of the most important research areas in network analysis is link prediction [46]. The objective of link prediction is to forecast prospective links from existing topological information of the network or identify unobserved links from the

[*] Corresponding author. Fax: +8651487887937.
E-mail addresses: nahlaibrahim@uofk.edu (N.M. Ahmed), yzulchen@163.com (L. Chen).

http://dx.doi.org/10.1016/j.ins.2016.09.029
0020-0255/

existing network structure. Link prediction is exploited to identify and categorize the human behavior and activity [1] in social networks. Link prediction can be applied to detect criminals and terrorists via their secret contacts [29] in social security networks. Link prediction is also employed to analyze the trend of changes in sensor networks [48], to perform web searches in the World Wide Web [17], to obtain the best possible routing [18], and to guarantee the confidentiality of information transmission [27]. In recent years, bipartite link prediction has been widely applied in areas such as recommendation [32,42,47], scientist-paper cooperation analysis [24], scientific paper impact prediction [22], medical parameter network analysis [19,20], and protein interaction prediction [13].

As relations between individuals in networks vary dynamically, links in social networks are continuously changing. Old links will possibly disappear from the network, while new ones may emerge constantly. For example, email communications between friends, transactions between businesses, and partnerships between scientific researchers change over time. Thus, the link prediction methods must be able to detect the changes of relationships among individuals in a dynamic network. In recent years, many methods for identifying latent or prospective links in dynamic networks have been proposed.

Similarity-based methods are the most common approach used for link prediction. In such methods, each pair of node is associated with an index to indicate the similarity between the corresponding nodes. This similarity quantifies the likelihood of link existence in the graph. Some essential attributes of the nodes can be used to define their similarity, such as common features or topological structures between the nodes [44]. Many studies in social networks have shown that a higher similarity may exist between individuals who are close to each other [2,11]. Structural similarity indexes are often used in popular similarity-based methods. There are three categories of similarity indexes: local indexes, quasi-local indexes, and global indexes. To calculate local indexes, only the neighbor information of each node is required. Such local indexes include Common Neighbors, Jaccard, Salton, Sorensen, Preferential Attachment, Hub Depressed, Hub Promoted, Adamic–Adar, Resource Allocation, and Leicht–Holme–Newman (LHN1) indexes [29]. Quasi-local indexes require more structural information than local indexes and less information than global indexes. Quasi-local indexes include Local Random Walk, Superposed Random Walk [26], and Local Path Index [28,50]. Global indexes require comprehensive information for link prediction tasks. They use global topological information of networks, such as the Katz, Matrix Forest Index (MFI), and Leicht–Holme–Newman (LHN2) indexes [29]. In general, the use of global indexes can yield higher-quality prediction results than quasi-local and local indexes. However, local indexes require less information than global ones. Another class of similarity-based methods is random walk methods. Those methods include SimRank [3], Random Walk with Restart, Cos+, and Average Commute Time [29]. Based on random walk, B. Chen et al. [9] presented an algorithm for predicting links to nodes of interest. The method first constructs a subgraph centered at the node the user is interested in. Then, it computes the similarity score in the subgraph.

Some methods predict potential links by exploring the structural characteristics of the network. Purnamrita et al. [36] presented a nonparametric method for link prediction in temporal networks. This method partitions the time domain into subsequences represented by graph snapshots. Their method predicts connections between nodes based on their topological features and local neighbors. Kim et al. [21] proposed an approach to identify potential links in networks. Their approach is based on node centrality, which can predict the future importance of nodes. Murata et al. [30] investigated the relationship between graph structure and link occurrence. They advanced a weighted proximity-based method for link prediction in social networks.

For temporal network link prediction, some methods employ machine-learning techniques. Vu et al. [45] proposed a continuous-time regression model that can integrate time-varying regression coefficients and time-dependent network statistics. Pujari et al. [35] used a supervised rank aggregation approach to predict potential links in temporal networks. Zeng et al. [49] presented a method using semi-supervised learning. To predict potential links in a network, their method exploits the latent information of the node pairs that are not currently linked. He et al. [15] advanced a link prediction ensemble algorithm using an ordered weighted averaging operator. The algorithm assigns weights for nine local information-based link prediction algorithms and then aggregates their results to obtain the final prediction scores. Bao et al. [4] advanced a network link predictor using principal component analysis to identify features that are important to link prediction. Madadhain et al. [33] proposed an event-based link prediction approach on temporal networks. By applying machine-learning and data-mining techniques, their approach is able to forecast potential cooperation between individuals in social events. Using data-mining techniques such as frequent-pattern and association-rule mining, Bringmann et al. [7] advanced a method for link prediction in temporal networks. To avoid the high computational cost of optimization in machine-learning methods, some heuristic methods are employed in link prediction. Catherine et al. [6] presented a method for predicting future links by applying the covariance matrix adaptation evolution strategy. Based on ant colony optimization, Sherkat et al. [43] introduced an unsupervised link prediction algorithm.

Probabilistic model-based methods are also used in link prediction on complex networks. Hu et al. [16] presented a probabilistic model to discover individual actions in social networks. They also proposed an approach employing a genetic algorithm to optimize the model. Barbieri et al. [5] proposed a stochastic link prediction model on directed graphs with node attribute features. In addition to predicting links, the model also provides explanations for the links detected. To estimate the probability of a link appearance, Gao et al. [12] presented a model that exploits various types of information in the temporal network. For link prediction in a user-object network, Ji Liu et al. [25] proposed an approach that takes both time attenuation and diversion delay into consideration. By extending the exponential random graph model, Hanneke et al. [14] advanced a set of statistical models for dynamic network link prediction. Because the probabilistic model proposed needs to know the distribution of link occurrence, it is impractical for link prediction in a real-world network.

In this work, we present a fast similarity-based algorithm to predict the future links in temporal networks. In the method, we first combine snapshots of temporal networks into a weighted graph using a damping factor to assign greater importance to more recent networks. Then, a sub-graph is generated by random walks from each node. Then, the similarity score is computed within the small sub-graph centered at each node to reduce the computation time. By choosing a proper number of sampled paths, we can restrict the error of the estimated similarities within a given threshold. While other random walk-based algorithms require $O(n^3)$ time for a network with n nodes, the computation time of our algorithm is $O(n^2)$, which is the lower bound of the time complexity for similarity-based prediction algorithms. Moreover, the proposed method integrates global topological and temporal information, and demonstrates the best or comparable quality results in less time than other methods.

The remainder of this paper is organized as follows. Section 2 defines the transformation matrix for link prediction in temporal networks. Section 3 discusses local random walk methods and their indexes for link prediction. In Section 4, we present a sampling-based algorithm for link prediction. The time complexity of the algorithm is analyzed in Section 5. In Section 6, experimental results of our methods on real datasets are analyzed and compared with those of other methods. In Section 7, we conclude this work.

2. Transformation matrix for temporal networks

Because relations between social individuals are continuously varying and evolving, link probabilities in real social networks are also changing constantly. Thus, it is necessary that link prediction methods be able to detect the changes of relationships among individuals in a dynamic network. We use an undirected and un-weighted graph to represent the network. Let $V=\{v_1, v_2, ..., v_n\}$ be a set of vertices, and the temporal uncertain network can be described by snapshots $G_t=(V, E_t, A_t)$ for$t=1, 2, ..., T$, where $T$is the time window size. We use a list of symmetric matrices $A_1, A_2, \ldots, A_{T_N}$to denote the adjacency matrices of graphs $G_1, G_2, \ldots, G_{T_N}$, respectively. The binary value of $A_t(i, j)$ indicates the existence of an edge linking nodes i and j, $i, j=1, 2, ..., n$, during the time period t, $t=1, 2, ..., T$. Given the initial time t_0 and a sequence of graphs$G_{t_0}, G_{t_0+1}, \ldots, G_{t_0+T-1}$, the goal of link prediction is to predict the occurrence probabilities of edges at time t_0+T.

In this work, we specify the output of a link prediction problem as an $n \times n$ matrix $S_t(i, j)$ with each element i, jas a similarity score that is proportional to the predicted occurrence probability of edge (i, j) at time t.

Recently, some approaches have been advanced to detect potential or future links in temporal social networks. Most such methods treat temporal networks as one-time events and ignore the time that a link occurs. In this work, we exploit temporal and topological information to predict potential links. In our proposed method *TS-VLP* (time series vertex link prediction), we integrate the sequence of adjacency matrices$G_1, G_2, ..., G_T$into one combined graph $G_{1,\,T}$ with the adjacency matrix $A_{1,\,T}$. In the evolution of a temporal network, recent snapshots are more reliable for future link prediction, and they should be emphasized to obtain more accurate prediction results. To give larger emphasis to more recent information, a damping factor is used in our method. Based on the damping factor, we define the transformation matrix $A_{1,\,T}$ as follows.

Definition 1 (transformation matrix)**.** Let a temporal network be described by snapshots $G_t=(V, E_t, A_t)$ for $t=1, 2, ..., T_N$. A list of symmetric matrices $A_1, A_2, \ldots, A_{T_N}$ are the adjacency matrices of graphs $G_1, G_2, \ldots, G_{T_N}$, respectively. The transformation matrix$A_{1,\,T}$is defined as

$$A_{1,T} = \sum_{t=1}^{T} \gamma^{T-t} A_t, \quad 0 < \gamma < 1 \tag{1}$$

In (1), γ is the damping factor. Generally, setting a proper value of γ is problem-dependent. For example, small values of γ can be suitable for rapidly changing networks, while in networks with high stability, high values of γ can work more efficiently. In our proposed method, this transformation matrix $A_{1,\,T}$ is used as the adjacency matrix for link prediction on the temporal networks series $G_{t_0}, G_{t_0+1}, \ldots, G_{t_0+T-1}$.

3. Local random walk

In the similarity-based method, the purpose of link prediction is to estimate a similarity score, denoted as $S(x, y)$, for each node pair $(x, y) \in U$ in network $G=(E,V)$,here U is the universal edge set, namely, the set of edges in the complete graph with the same node set V. For a node pair $(x, y) \in U \setminus E$, the score $S(x, y)$ reflects the similarity between nodes x and y. A larger similarity $S(x, y)$ between nodes x and y indicates that they are more likely be linked by an edge.

Liu and Lü [26] investigated link prediction methods by random walk. They discovered that a local random walk can possibly obtain higher-quality prediction results than a global random walk.

3.1. Local random walk

To calculate the topological similarity between nodes x and y, a particle is first placed on node x, and it then walks randomly on the network. A sequence of $n \times n$ matrixes $\pi(\tau)$ $(t = 0,1,2,\ldots)$ is defined, where an element $\pi_{xy}(\tau)$ is the probability of a particle from x reaching y at time step τ. The initial value of a matrix element is defined as

$$\pi_{xy} = \begin{cases} 1 & \text{If } x = y \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

At time step τ, the particle randomly walks on the network according to a transformation matrix P=s[p_{xy}] and generates a new matrix $\pi(\tau+1)$. In the transformation matrix P, element p_{xy} indicates the probability that the particle on node x is going to reach node y in the next step and is defined as p_{xy}=$a_{xy}/d_{x,}$, where a_{xy} is the (x, y) element of the adjacency matrix and d_x is the degree of node x. By the random walk of the particle, matrix $\pi(\tau)$ evolves as:

$$\pi(\tau+1) = P^T\pi(\tau), \quad \tau \geq 0 \tag{3}$$

Given an initial matrix $\pi(0)_,$ $\pi(\tau)$ can be obtained iteratively using (3). The *LRW* index $S_{xy}^{LRW}(\tau)$, which measures the similarity between nodes x and y at time step τ, is thus defined as:

$$s_{xy}^{LRW}(\tau) = q_x \cdot \pi_{xy}(\tau) + q_y \cdot \pi_{yx}(\tau) \tag{4}$$

where q_x and q_y are the initial configuration functions. Liu and Lü [26] suggested a simple form to determine the value of q_x by using the degree of node x, namely

$$q_x = \frac{d_x}{|E|} \tag{5}$$

Here, $|E|$ is the number of existing links in the network.

3.2. Superposed random walk

On the basis of the LRW index, Liu and Lü [26] proposed the superposed random walk (*SRW*) index. To calculate the SWR index, particles are constantly released from one node, and LRW indexes with different lengths are calculated. The summation of those LRW indexes forms the SRW index. Let $S_{xy}^{LRW}(\tau)$ be the similarity measure between nodes x and y at time step τ, defined as follows

$$s_{xy}^{SRW}(\tau) = \sum_{l=1}^{\tau} s_{xy}^{LRW}(l) = q_x \sum_{l=1}^{\tau} \pi_{xy}(l) + q_y \sum_{l=1}^{\tau} \pi_{yx}(l) \tag{6}$$

Let n be the number of nodes in the network, and the time required to compute the matrix $\pi(\tau)$ using Eq. (3) is $O(n^3)$. The time to calculate the S^{SRW} scores using (6) on all pairs of nodes in the network is $O(\tau \bullet n^3)$. Such computation time is impractical for link prediction in large networks. Therefore, it is necessary to find an efficient approach to predict the links with less time cost. Here, we propose a fast similarity-based method to calculate the *SRW* indexes of the links related to each node. Instead of calculating the *SRW* indexes in the entire network, we estimate the *SRW* indexes within a number of sampled paths starting from each node and reduce the time complexity to $O(n^2)$, which is the lower bound of similarity-based prediction methods.

4. Sampling-based similarity computation

4.1. Approximation of SRW by path sampling

To calculate the *SRW* indexes using (6), the key issue is to calculate $\sum_{l=1}^{L} \pi_{xy}(l)$. Given nodes x and y, we denote the set of the paths of length L starting from node x as $P_L(x)$. Let p=(x, x_1, x_2, ..., x_L) be a path in $P_L(x)$ and $d_{xy}(p)$ be the minimal distance from x to y in path p. If node y is not included in path p, then $d_{xy}(p)$=∞. Because $\pi_{xy}(\tau)$ is the probability of a randomly walking particle from x reaching y at time step τ, it can be equivalently defined as

$$\pi_{xy}(\tau) = \frac{1}{|P_\tau(x)|} \sum_{p \in P_\tau(x)} I[d_{xy}(p) = \tau] \tag{7}$$

Here, the indicator function $I(.)$ is defined as

$$I[d_{xy}(p) = \tau] = \begin{cases} 1 & \text{If } d_{xy}(p) = \tau \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

Then,

$$s_{xy}^{SRW}(L) = q_x \sum_{l=1}^{L} \frac{1}{|P_L(x)|} \sum_{p \in P_L(x)} I[d_{xy}(p) = l] + q_y \sum_{l=1}^{L} \frac{1}{|P_L(y)|} \sum_{p \in P_L(y)} I[d_{xy}(p) = l]$$

Thus

$$S_{xy}^{SRW}(L) = \frac{q_x}{|P_L(x)|} \sum_{p \in P_L(x)} I[y \in p] + \frac{q_y}{|P_L(y)|} \sum_{p \in P_L(y)} I[x \in p] \tag{9}$$

Here, the indication function $I[x \in p]$ is defined as

$$I[x \in p] = \begin{cases} 1 & x \text{ is included in path } p \\ 0 & \text{otherwise} \end{cases}$$

Let

$$T(L, x, y) = \frac{1}{|P_L(x)|} \sum_{p \in P_L(x)} I[y \in p] \tag{10}$$

and letting $T(L, y, x) = \frac{1}{|P_L(y)|} \sum_{p \in P_L(y)} I[x \in p]$ be the two terms in (9),

$$S_{xy}^{SRW}(L) = q_x T(L, x, y) + q_y T(L, y, x) \tag{11}$$

By (10), we can see that $T(L, x, y)$ can be computed by counting the number of paths including node y in $P_L(x)$. Therefore, we need to enumerate all of the paths starting from x in G. However, the time complexity is exponentially proportional to the path length L and thus is inefficient when L increases. To reduce the computation time, we propose a sampling-based method to estimate the number of such paths. The key idea is that we randomly sample R paths from the network and estimate $T(L, x, y)$ by (10) based on the sampled paths. Let $Q_L(x)$ be the set of paths sampled from $P_L(x)$, and $|Q_L(x)|=R$. Then, the approximation of $T(L, x, y)$ is:

$$T'(L, x, y) = \frac{1}{R} \sum_{p \in Q_L(x)} I[y \in p] \tag{12}$$

Using $T'(L, x, y)$ to approximate $T(L, x, y)$ in (10), the final similarity score $S_{xy}^{SRW}(L)$ for link prediction defined in (6) is approximated by:

$$\hat{S}_{xy}^{SRW}(L) = q_x T'(L, x, y) + q_y T'(L, y, x) \tag{13}$$

In Eq. (13), two terms of $\hat{S}_{xy}^{SRW}(L)$, namely, $T'(L, x, y)$ and $T'(L, y, x)$, can be calculated separately in the sampled path sets $Q_L(x)$ and $Q_L(y)$. In set $Q_L(x)$, we need only to count the number of paths including y. Similarly, in set $Q_L(y)$, we need only to count the number of paths including x.

4.2. Path selection by random walk

To generate a path with length L from the given node x, we conduct random walks of L steps from x using t_{ij} as the transition probability from vertex v_i to v_j. We define

$$t_{ij} = \frac{a_{ij}}{\sum_{v_k \in N(v_i)} a_{ik}} \tag{14}$$

Here, a_{ij} is the (I,j) element of transformation matrix $A_{1,\ T}$, and $N(v_i)=\{v|v \in V,\ (v,v_i) \in E\}$ is the set of neighbors of v_i.

The random walk-based algorithm for generating the set of sampled paths $Q_L(x)$ is shown in Algorithm 1.

It can easily be seen that the time complexity of the algorithm *Generating_Paths*(x,L,R) is $O(LR)$. However, we can perform random walks efficiently in a distributed network. Sarma et al. [39] presented several distributed algorithms whose time complexities are $O((RL)^{2/3}D^{1/3})$, $O(\sqrt{R.L.D} + R)$ [37,40] for performing R random walks, where L is the length of the random walk and D is the diameter of the network. For a dynamic network where the network topology changes over time, they presented a distributed random walk algorithm [38] with time complexity $O(\sqrt{\tau.\Phi})$, where τ is the dynamic mixing time and Φ is the dynamic diameter of the network. They also give an unconditional lower bound on distributed random walk computation [41,31].

4.3. The size of the sampling path set

Because $Q_L(x)$ is a set of sampling paths from the complete set $P_L(x)$, using $T'(L, x, y)$ on $Q_L(x)$ to approximate $T(L, x, y)$ on $P_L(x)$ may cause error. Obviously, if $Q_L(x)$ has a larger size R, the result will be more accurate, but the computation will require more time. Therefore, we should find a proper size R for the sampling set $Q_L(x)$ so that we can restrict the error under a given bound $\varepsilon > 0$. Let $x \in V$ be a node in G; we need to construct a sample path set $Q_L(x)$ such that, for every node $y \in V$ $(y \neq x)$, the difference between $T(L, x, y)$ and the approximated one $T'(L, x, y)$ is not greater than ε.

We estimate such a proper size R of the sampling set $Q_L(x)$ by theoretical analysis of the random path sampling algorithm. In general, the $T(L, x, y)$ value can be viewed as a probability measure defined over all paths in $P_L(x)$. Thus, we can adopt the results from the *Vapnik-Chernovenkis* learning theory to analyze the proposed sampling-based algorithm. To begin

```
Algorithm 1 Generating_Paths(x,L,R);.
Input:
      $x$: the given node;
      $L$: length of the sampled paths;
      $R$: number of sampled paths;
      $A$: the adjacent matrix of $G$;
Output:
      $Q_L(x)$: the set of sampled paths;
Begin
  1.Compute the transition probability matrix $T=[t_{ij}]$ according to Eq. (14);
  2. Initialization: $Q_L(x)=\phi$;$x_{k0}=x$;
  3. For $k$=1 to $R$ do
     /* Let the $k$-th path in $Q_L(x)$ be $p_k=(x, x_{k1}, x_{k2}, ..., x_{kL})$;*/
        For $j$=1 to $L$ do
           Select a node in $N(x_{k,\, j-1})$ according to the transition probability;
           Name the selected node as $x_{k,\, j}$ and add it to $p_k$;
        End for $j$;
        $Q_L(x)=Q_L(x)\cup\{p_k\}$;
    End for $k$;
End
```

with, we first introduce some basic definitions and fundamental results from the *Vapnik-Chernovenkis* learning theory and then demonstrate how to utilize these concepts and results to obtain a proper size R of the sampling set $Q_L(x)$.

Definition 2 (shatter)**.** Let $(S;H)$ be a range space, where S denotes a domain and H is a range set on S. For any set $B\subseteq S$, pH(B) $=\{S\cap A\colon A \in H\}$ is the projection of H on B. The set B is shattered by H if pH(B)$=2^{|B|}$, where $2^{|B|}$ is the power set of B.

Definition 3 (Vapnik–Chervonenkis dimension)**.** The Vapnik–Chervonenkis dimension of H is the maximum cardinality of a subset of S that can be shattered by H.

We use $VC(H)$ to denote the Vapnik-Chervonenkis dimension, which is abbreviated to “VC dimension” in the rest of this paper.

In our problem, the domain S is presented by $P_L(x)$, which is the set of paths starting from node x with length L .We use $p_{x,v}$ to denote a path starting from node x and ending at node v and $|p_{xv}|$ to denote the length of path p_{xv}. We define the range set H for node x as $H_x=\{p_{x,\,v}|v \in V,\ |p_{x,\,v}| \le L\}$. Let $p_{x,v}\in H_x$ and $p \in P_L(x)$; if $p_{x,v}$is a sub-path of p, we denote them as $p_{x,v}\in p$. Let P be a subset of $P_L(x)$; if, for every path $p\in P$, it satisfies $p_{x,v}\in p$, then we denote them as $p_{x,v}\in P$. Let Q be a subset of $P_L(x)$. If, for every subset P of Q, we have a $p_{x,v}\in H_x$ such that $p_{x,v}\in P$, then we say that Q can be shattered by H. Let $Q_L(x)$ be the subset of $P_L(x)$ that has the maximum cardinality among the subsets that can be shattered by H. Then, $|Q_L(x)|$ is the VC-dimension of H, which is denoted as $VC(H)$.

Let $X=\{x_1,..., x_R\}$ be a set of independent random variables sampled by a distribution Φ over domain S. For a set $A\subseteq S$, we use $\Phi(A)$ to denote the probability an element $x \in A$ being sampled from S. Let the empirical estimation of $\Phi(A)$ on X be

$$\Phi_X(A) = \frac{1}{R}\sum_{i=1}^{R} I_A(x_i) \tag{15}$$

where $I_A(x_i)$ is the indicator function defined as:

$$I_A(x_i) = \begin{cases} 1 & x_i \in A \\ 0 & \text{otherwise} \end{cases}$$

The question of interest is how well we can estimate $\Phi(A)$ using its unbiased estimator, the empirical estimation $\Phi_X(A)$. We first give the definition of the ε-approximation as follows.

Definition 4 (ε-approximation)**.** Let Φ be a probability distribution defined on S and H be a range set on S. For $\varepsilon \in (0, 1)$, an ε-approximation to (H, Φ) is a set A of elements in S such that

$$\sup_{A\in H}|\Phi(A) - \Phi_X(A)| \le \varepsilon \tag{16}$$

One important result of the *Vapnik–Chervonenkis* theory is that, if we know the VC-dimension of H, we can construct an ε-approximation by randomly sampling elements from the domain under the distribution Φ. This is summarized in the following theorem [23].

Theorem 1. *Let Φ be a distribution on a domain S and H be a range set on S with VC-dimension VC(H)=d. For an error bound ε and a probability $\delta \in (0, 1)$, let Q be a set of $|Q|$ points sampled from S according to Φ, with*

$$|Q| = \frac{1}{\varepsilon^2 \cdot c}\left[d \cdot \ln\frac{1}{c} + \ln\frac{1}{\delta}\right] \tag{17}$$

Here, $c > 0$ is a constant. Then, Q is an ε-approximation to (H, Φ) with a probability of at least $1-\delta$.

In our setting, we set the domain to $P_L(x)$, which is the set of all paths of length L starting from node x. We first give the following theorem to show an upper bound of the VC-dimension of H in Theorem 1.

Theorem 2. *For a node $x \in V$, let $P_L(x)$ be the set of all of the paths starting from node x with length L and the range set H on domain $P_L(x)$ for node x be $H_x=\{p_{x,\,v}\colon v \in V, |p_{x,\,v}| \le L\}$. Then, the VC-dimension of H_x satisfies*

$$VC(H_x) \le \log_2 L + 1 \tag{18}$$

Proof. Assume that $VC(H)=l$.By Definition 2, we know that there must be a subset Q of $P_L(x)$ such that $|Q|=l$, and Q can be shattered by H. That is to say, for every subset Q_i of Q, there must be $p_{x,v_i} \in H$ such that Q_i consists of all paths with p_{x,v_i} as its sub-path. For different sub-sets Q_i and Q_j of Q, they must correspond to different ranges p_{x,v_i} and p_{x,v_j} in H. Obviously, there are $2^{|Q|}=2^l$ sub-sets of Q.

Let $p=(x, u_1, u_2, ..., u_L)$ be a path in Q. Then, the power set of Q, denoted as $2^{|Q|}$, can be divided into two parts: one consists of the subsets of Q that include p, and the other consists of the subsets of Q that do not include p. We denote the former as $2_p^{|Q|}$ and the latter as $2_p^{-|Q|}$. Obviously, $2_p^{|Q|}$ and $2_p^{-|Q|}$ have an equal number of subsets. Therefore, there are 2^{l-1} subsets in $2_p^{|Q|}$, and they correspond to 2^{l-1} different ranges in H. For every subset Q_i in $2_p^{|Q|}$, it has a sub-path $p_{x,u_i} = (x, u_1, u_2, ..., u_i)$ of p such that every path in Q_i has a sub-path p_{x,u_i}. Therefore, the corresponding ranges must take the form $p_{x,u_i} = (x, u_1, u_2, ..., u_i)$, $(i=1, 2, ..., L)$, and there are at most L such ranges. Therefore, we have $2^{l-1} \le L$, namely $l \le \log{}_2 L+1$. ■

According to Eq. (17) and Theorem 2, for given values of the error bound ε and probability δ, if the size of sample set $Q_L(x)$ is set as

$$R = |Q_L(x)| = \frac{1}{\varepsilon^2 . c}\left((\log_2 L + 1). \ln\frac{1}{c} + \ln\frac{1}{\delta}\right) \tag{19}$$

then $T'(L, x, y)$ is an ε-approximation of $T(L, x, y)$ with probability of at least 1-δ, namely, they satisfy

$$\sup_{p_{xy}\in P_L(x)} \left|T(L,x,y) - T'(L,x,y)\right| \le \varepsilon \tag{20}$$

4.4. Algorithm for similarity estimation involving a given node

To calculate $\hat{S}_{xy}^{SRW}(L)$ by Eq. (11), the two terms $T'(L, x, y)$ and $T'(L, y, x)$ need to be calculated separately in the sampled path sets $Q_L(x)$ and $Q_L(y)$. Namely, the number of paths including *y*in $Q_L(x)$ should be counted. Additionally, the number of paths including x in $Q_L(y)$ should be counted. Here, we present an algorithm to estimate the similarity terms involving a given node x in the sampled path set $Q_L(x)$. Given a network $G=(V,E)$, a node x in V, an error bound ε, and a probability δ, our algorithm first computes a set of paths starting from x. After such set $Q_L(x)$ is constructed, the value of $T'(L, x, y)$ is calculated for x with every node y in the sampled paths.

The framework of our algorithm *VLP* (vertex link prediction) to estimate the similarity terms $T'(L, x, y)$ for a given node x is as follows.

```
Algorithm 2 VLP($x$, $\delta$,$\varepsilon$).
Input:
        $A$:Adjacency matrix of network $G=(V,E)$;
        $\delta$: The probability;
        $\varepsilon$: The error bound;
        $x$:The vertex been queried;
        L: The largest length of the path considered;
Output:
        $T'(L, x, y)$: similarity term for vertex $x$ with node $y$ in $G$;
Begin
1. Set $T'(L, x, y)=0$ for all node pairs $x$ and $y$;
2. Calculate sample size $R$ by Eq. (19) according to the error bound $\varepsilon$ and probability $\delta$;
3. Generating_Paths ($x$,$L$,$R$);
     /*Generate $R$ random paths, store them in set $Q_L(x)$*/
4.For each path $p$ in $Q_L(x)$ do
        /* suppose $p_k=(x, u_1, u_2, ..., u_L)$*/
        For $i$=1 to $L$ do
        $T'(L,x,u_i) = T'(L,x,u_i) + \frac{1}{R}$;
        End for $i$;
    End for $p$;
End
```

Based on the algorithm *VLP*, which estimates the similarity terms $T'(L, x, y)$ for a given node x, we present an algorithm *TS-VLP* (time serious vertex link prediction) to predict potential links in time series networks by integrating time and topology information. In the method, the combined graph $G_{t_0,T}$ is first computed using formula (1). Then, it estimates the similarity terms $T'(L, x, y)$ for each node x in graph $G_{t_0,T}$ using the *VLP* algorithm. Based on the estimated similarity terms $T'(L, x, y)$ for all of the nodes, the similarity between the node pairs can be obtained. The framework of the *TS-VLP* algorithm is as follows.

```
Algorithm 3 TS-VLP($A_1$, $A_2$, ...$A_T$, $L$, $\delta$,$\varepsilon$).
Input:
        $A_1$, $A_2$, ...$A_T$:Adjacency matrices of time series networks $G_1$, $G_2$, ..., $G_T$, where $G_i = (V, E_i)$;
        $\delta$: The probability;
        $\varepsilon$: The error bound;
        $L$: The largest length of the path considered;
Output:
        [$\widehat{S}_{xy}^{SRW}$]: Similarity index matrix;
Begin
1. Generate a combined time series graph $G_{1,\,T}$ according to Eq. (1);
2. For every vertex $x$ in $G_{1,\,T}$ do
   /* compute similarity term $T'(L, x, y)$ for vertex $x$ with other node $y$ in $G_{1,\,T}$*/
      $VLP(x, \delta, \varepsilon)$;
   End for $x$;
3. For each node pair $(x,y)$ in $G$ do
        $\widehat{S}_{xy}^{SRW} = q_x T'(L, x, y) + q_y T'(L, y, x)$;
      Endfor
End
```

5. Time complexity analysis

Let n be the number of nodes in the network and L and R be the length and the number of sampled paths, respectively. Obviously, the time complexity for the *Generating_Paths* algorithm to generate a set of sampled paths is $O(R.L)$. For a given vertex x, the *VLP* algorithm takes $O(R.L)$ time to compute the similarity terms $T'(L, x, y)$ between x and other vertices in the network. For a temporal network presented by a sequence of time series graphs, line 1 of the *TS-VLP* algorithm constructs the combined time series graph $G_{1,\,T}$ in $O(n^2)$ time. Line 2 of the algorithm computes the similarity term $T'(L, x, y)$ for each vertex x with other nodes in $O(R.L)$ time. Therefore, it requires $O(R.L.n)$ time to compute the similarity terms for all n nodes in the network. Finally, line3 computes the similarity between all pairs of nodes in $O(n^2)$ time. Because R and L can be treated as constants, the time complexity of our proposed algorithm *TS-VLP* is $O(n^2)$. Because there are n^2 node pairs in the network, n^2 similarity indexes must be computed for the network. Therefore, $O(n^2)$ is the lower bound of the time complexity of a similarity-based link prediction algorithm.

The computation time complexity for local index-based link prediction algorithms is $O(n^2)$. For instance, for local indexes such as *CN*, let k be the average degree of nodes in the network, and computing the common neighbors for each node pair (i, j) takes $O(k)$. Therefore, the time complexity for computing the *CN* index for all node pairs is $O(n^2.k)$. Similarly, other local indexes such as *JC*, *AA*, *RA*, and *HPI* also have the same time complexity of $O(n^2.k)$. However, because those local indexes only use the topological information of first-order neighbors, the quality of their prediction results is much lower than that of *TS-VLP*. For the quasi-local indexes *LRW* and *SRW*, $O(n^3)$ time is needed to calculate the matrix $\pi(\tau)$ in *LRW* and $O(\tau \bullet n^3)$ time to calculate the *SRW* scores. For computing a global topological path-based index such as *LHN2*, the time complexity is also $\mathrm{O}(n^3)$, which is much slower than the *TS-VLP* method. Although the *TS-VLP* method requires much less time than quasi-local and global index-based methods, it still can obtain high-quality prediction results. *TS-VLP* consumes less computational time because it considers the structural information based on a local random walk and can yield a better prediction result in fewer time steps than global random walk- and other local random walk-based methods. Compared with all of those similarity-based methods, the *TS-VLP* algorithm can achieve much higher performance for dynamic link prediction.

6. Experimental results and analysis

In this section, we conduct a set of experiments to test our method for link prediction in temporal social networks. We used several real temporal social networks in our experiments. All experiments were conducted on an Intel Core i3 computer with the Windows 7 operating system and 4 GB memory.

6.1. Datasets tested[1]

I. Reality mining(*R.Mining*)

This data set was created as a section of the Reality Commons Project by the Reality Mining experiment performed in 2004. The network in the dataset consists of 96 students of the Massachusetts Institute of Technology (MIT) and their

[1] All network datasets are available at KONECT, May 2015, http://konect.uni-koblenz.de/networks/.

Table 1
Main features of the datasets.

Datasets	#Nodes	# Links	%Unique links	Avg. deg	Avg. deg ($T=5$)	P. length	T_N
R. Mining	96	3200	0.4763	66.67	7.94	4 h	42
Haggle	274	16,482	0.2194	120.31	12.53	1 h	48
LKMLR	2783	9417	0.6188	6.77	1.13	1Day	30
I. Msg	1899	37,844	0.7313	39.87	7.11	7 Days	28
CoRR	1872	2616	0.4036	2.795	1.165	Year	12
IEEE	11,855	18,553	0.607	3.130	0.824	Year	19

communication via mobile phones over 9 months. The students are represented by the nodes in the network, and their communications are represented by edges linking the nodes [10].

II. Haggle

This network represents contacts between people carrying wireless devices. The data describes a network where each person is represented by a node, and each edge linking two persons shows that there exists a contact between them [8].

III. Linux kernel mailing list replies(*LKMLR*)

This is the communication network of the Linux kernel mailing list. Nodes are persons identified by their email addresses, and each directed edge represents a reply from one user to another.

IV. UC Irvine messages(*I. Msg*)

This network contains sent and received mail among the members of an online community of students from the University of California, Irvine. Each member of the community is represented by a node in the network, and the mails sent or received by the members are presented by the edges [34].

V. Digital Bibliography Library Project (DBLP)

DBLP is an English literature integrated database system for results of a study with a core of authors in computer science. We select two datasets: IEEE Transactions (*IEEE*) and Computing Research Repository (*CoRR*). Those datasets comprise literature from academic seminars or academic journals in the DBLP database. We treat each dataset as a network, where each node represents an author and each link represents the co-authorship between the authors of the corresponding nodes. They are partitioned into sequences of graphs. The length of the time period of this partition is a year for all datasets.

The main topological features of the tested networks represented by the datasets are shown in Table 1, which includes the number of nodes (*#Nodes*), number of links (*#Links*), percentage of non-duplicated links (*%Unique Links*), nodes' average degree within all time steps (*Avg. Deg*), nodes' average degree within $T=5$ time steps (*Avg. DegT*=5), length of the time series sequence (T_N), and length of the time period(*P.Length*).

6.2. Experimental setup

For each temporal network in the dataset tested, we input T_N snapshots of the graphs $G_1, G_2, \ldots, G_{T_N}$. At each time step t_0, $t_0 = 1, 2, \ldots, T_N - T$, we used the next T graphs, $G_{t_0}, G_{t_{0+1}}, \ldots, G_{t_0+T-1}$, to test the link prediction algorithm for detection of the potential links in G_{t_0+T}. Because we already know the topological structure of G_{t_0+T}, the prediction result can be assessed by comparing the links predicted with the actual presence of the links in G_{t_0+T}.

In the first part of our experiments, we tested our *TS-VLP* algorithm and compared the quality of its results with that of the other algorithms based on a reduced static graph. The reduced static graph-based method has been frequently used in methods for link prediction in temporal networks. In this method, networks represented by the snapshots of the graphs are first reduced and represented by a static graph. Then, an algorithm for link prediction in static graphs is exploited to predict potential links in the reduced static graph. That is to say, the series $G_{t_0}, G_{t_0+1}, \ldots, G_{t_0+T-1}$ is transformed into one reduced binary graph $G_{t_0,T}$ represented by a new adjacency matrix $A_{t_0,T}$ with the elements defined as

$$A_{t_0,T}(i,j) = \begin{cases} 1 & \text{if } \exists k \in [t_0, t_0+T-1] : A_k(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Then, a static network link prediction method, such as *Common Neighbor*, *Jaccard*, *Resource Allocation*, *Hub Promoted Index*, *Local Random Walk*, *Superposed Random Walk*, and *Leicht Holme Newman2*, is used on the reduced static graph $G_{t_0,T}$, and the result is used as the final link prediction solution on $G_{t_0,T}$. Those reduced static graph-based methods are denoted as *CN*, *JC*, *RA*, *HPI*, *LRW*, *SRW*, and *LHN2*, respectively. In the other parts of our experiments, we test *TS-VLP* under different parameter values, such as error bound ε, probability δ, and length of sampled path L, and show the influence of different parameter values on the precision of the results and computational time of the algorithm.

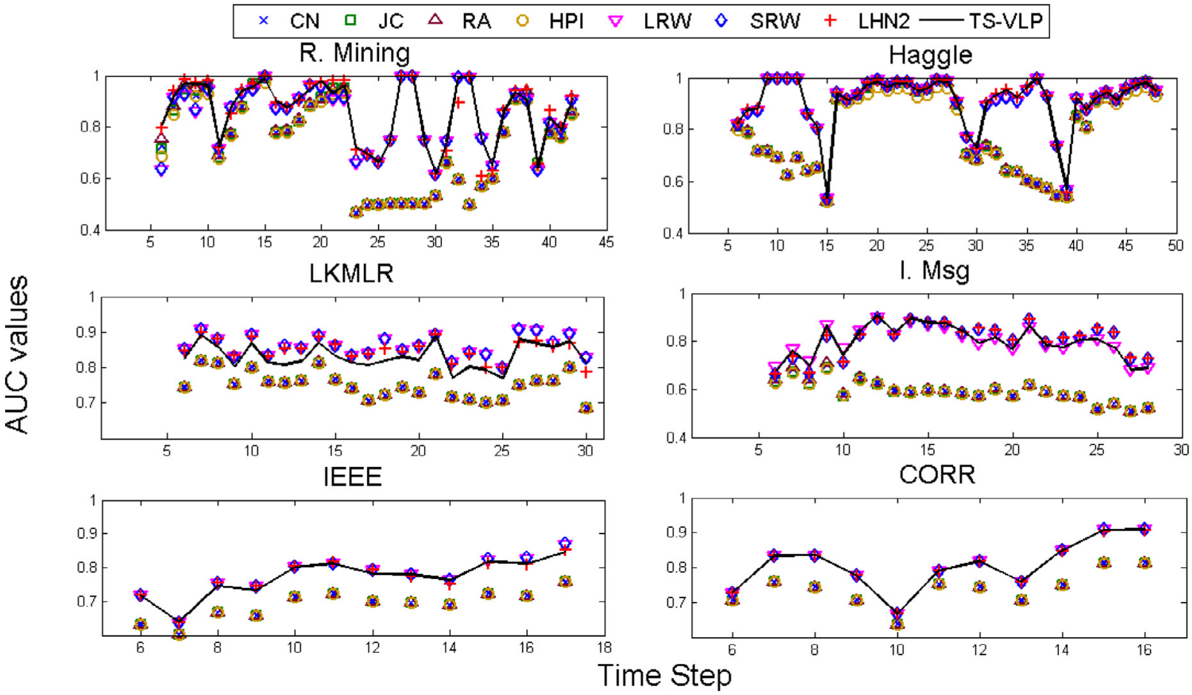

Fig. 1. AUC values at each time step obtained by different methods.

Once a prediction algorithm has computed the similarities of all of the node pairs, we use area-under-curve (AUC) to assess the precision of the results by different methods tested. The AUC value is defined as the expected ratio of an existing link with a higher similarity than the missing link. To calculate the AUC value of the prediction result, we randomly choose n pairs of links to compare their similarity scores. In each pair of links, one link is from the set of existing links E_{t_0+T}, and the other is from the set of non-existing links $E'_{t_0+T} = U - E_{t_0+T}$. For n independent comparisons, if n' times existing links yield larger scores than non-existing one, and n' ' times gain equal scores, the AUC value is defined as

$$AUC = (n' + 0.5n'')/n \tag{22}$$

Generally, a greater AUC value indicates higher quality of the prediction result. Obviously, the AUC value for a link prediction result satisfies $AUC \in [0, 1]$. Therefore, the AUC value of the optimal prediction result is 1, and the AUC value of the result by random prediction is 0.5. Therefore, a prediction result with an AUC value less than 0.5 is invalid.

6.3. Experimental results and analysis

We set the window size $T=5$, damping factor $\gamma=0.8$, constant $c=0.95$, error bound $\varepsilon=0.3$, length of sampled paths $L=15$, and probability $\delta=0.3$ for all datasets.

In the first part of our experiments, we tested and compared the performance of the proposed method *TS-VLP* with that of the reduced static graph-based methods *CN*, *JC*, *RA*, *HPI*, *LRW*, *SRW*, and *LHN2* on the four datasets. Fig. 1 shows the AUC values of the results at different time steps. In the figure, a continuous line is used to indicate the results obtained by *TS-VLP*, while the results obtained by other methods are signed by markers. Fig. 2 compares the average AUC scores of the results by *TS-VLP* with those by other methods. Fig. 2 also compares their average implementation times on all datasets. From Figs. 1 and 2, we can clearly see that our *TS-VLP* algorithm achieves a much higher accuracy than the reduced static graph-based methods *CN*, *JC*, *RA*, and *HPI* on all datasets. There are two reasons for our method *TS-VLP* yielding higher-quality results. One reason is that it considers more global information, and it can also integrate time with global topology information efficiently. The other reason is that the error bound ε used is significantly small, and *TS-VLP* also exploits the time factor, which is ignored in all other methods. However, the most powerful part of our algorithm is its fast implementation speed. Fig. 2 shows that *TS-VLP* requires much less running time than other methods, especially on large networks, and runs even faster than local index-based methods. *TS-VLP* can obtain fast speeds because it performs the link prediction for each node only in a small sub-graph instead of the entire graph. Compared with the other quasi-local index-based methods, such as *LWR* and *SWR*, although the *TS-VLP* method consumes much less time, it still can obtain high-quality prediction results. In *TS-VLP*, because the sampled paths in the sub-graph can reflect the global topological information, it can obtain higher quality prediction results in much less computational time.

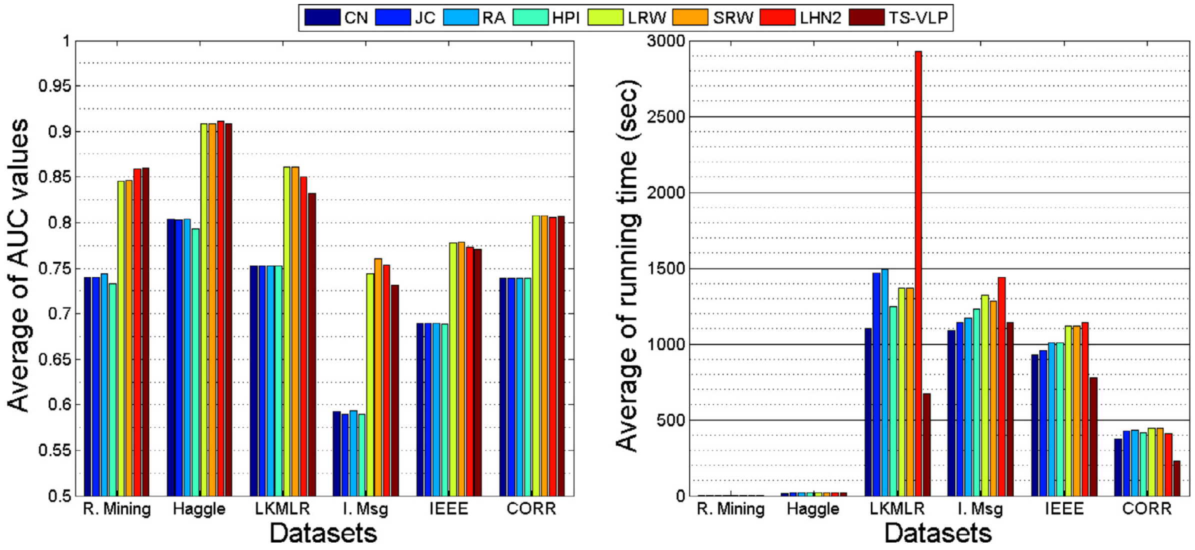

Fig. 2. Average AUC values and average running time for different methods.

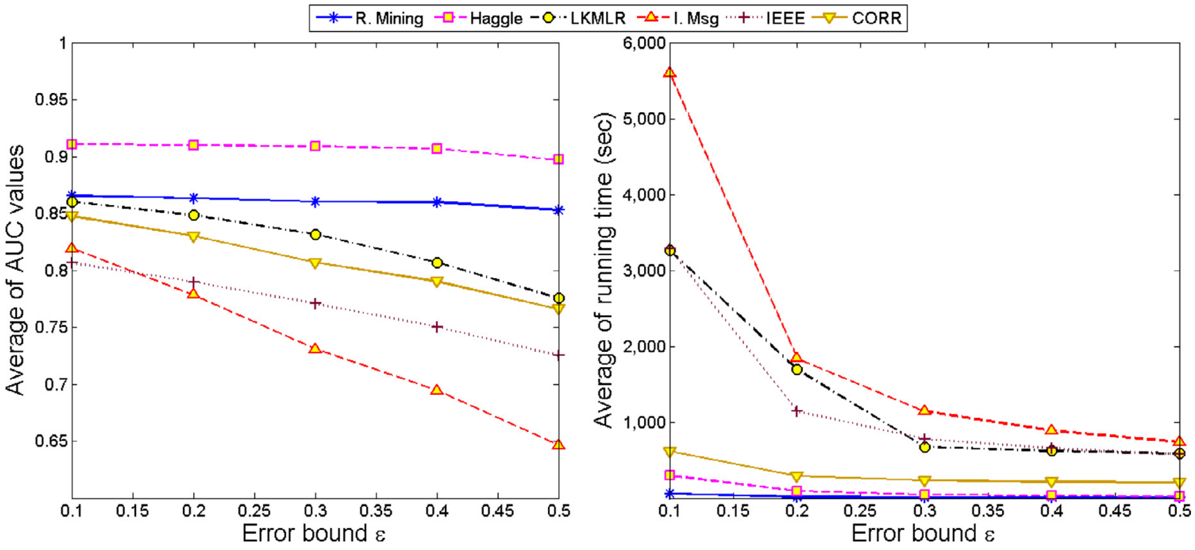

Fig. 3. Average AUC values and average running time under different ε values.

In the second part of the experiments, we investigated the relationship between the AUC scores of the results and the values of the error bound ε. We tested the *TS-VLP* algorithm with different error bound ε values and set the probability $\delta = 0.05$, and the number of sampled path length $L = 10$ on the datasets *R. Mining* and *Haggle* and $L = 15$ on the datasets *LMKLR* and *I. Msg*. Fig. 3 shows that the performance of each method is in roughly reverse proportion to the value of ε. Because *LKMLR* and *I. Msg* are relatively dense graphs, the error bound highly affects the performance of *TS-VLP* on those datasets. However, the decrements of *TS-VLP* performance on other datasets are relatively small. Using a larger value of ε, implementation of the method is faster because it generates fewer sampling paths in the sub-graph for each node. Therefore, we need to make a good balance between the computation time and the quality of the result by setting a proper value of ε. However, such a proper value of ε is dataset-dependent. It is still an open problem to set a proper value of ε for a given dataset to obtain the satisfied prediction result in a short period of time. In our experiments, we set the value of ε for a given dataset by considering the density of the network. For a network of dense connections, we set a smaller ε value to obtain a larger number of sampled paths *R* and to cover more global topological information.

In the third part of our experiments, the impact of *L*, which is the length of sampled paths, on the prediction results is studied. We set the probability $\delta = 0.05$ and the error bound $\varepsilon = 0.1$ on the datasets *R. Mining* and *Haggle* and $\varepsilon = 0.05$ on *LMKLR* and *I. Msg*. Fig. 4 shows the changes of AUC values by the *TS-VLP* method with different values of *L* for all datasets. If we set $L = 0$, the method achieves the lowest performance of 0.5, which is equal to the totally random prediction. The AUC

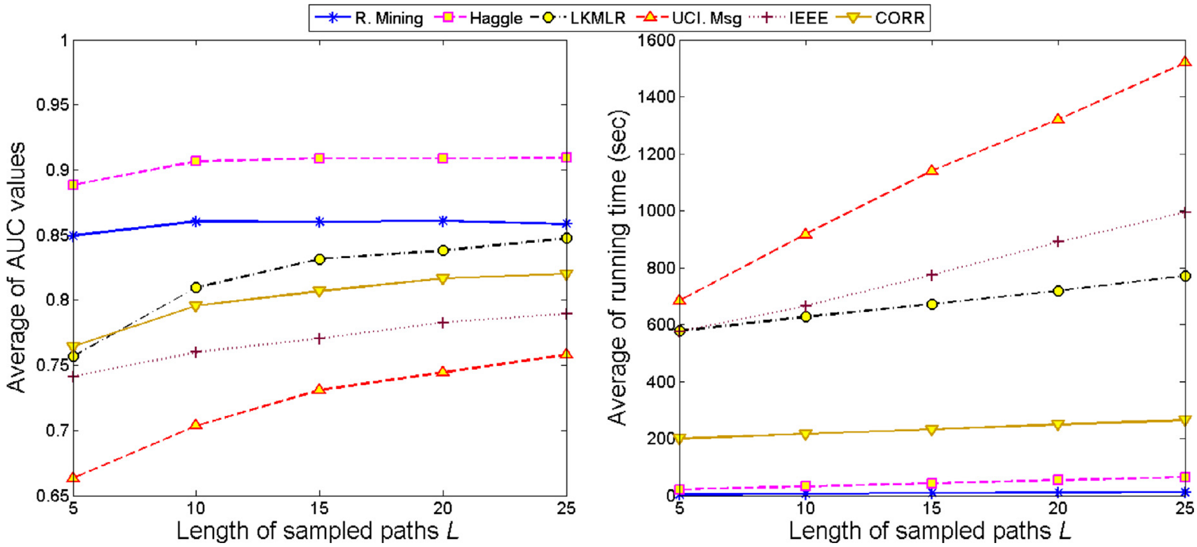

Fig. 4. Average AUC values and average running time under different L values.

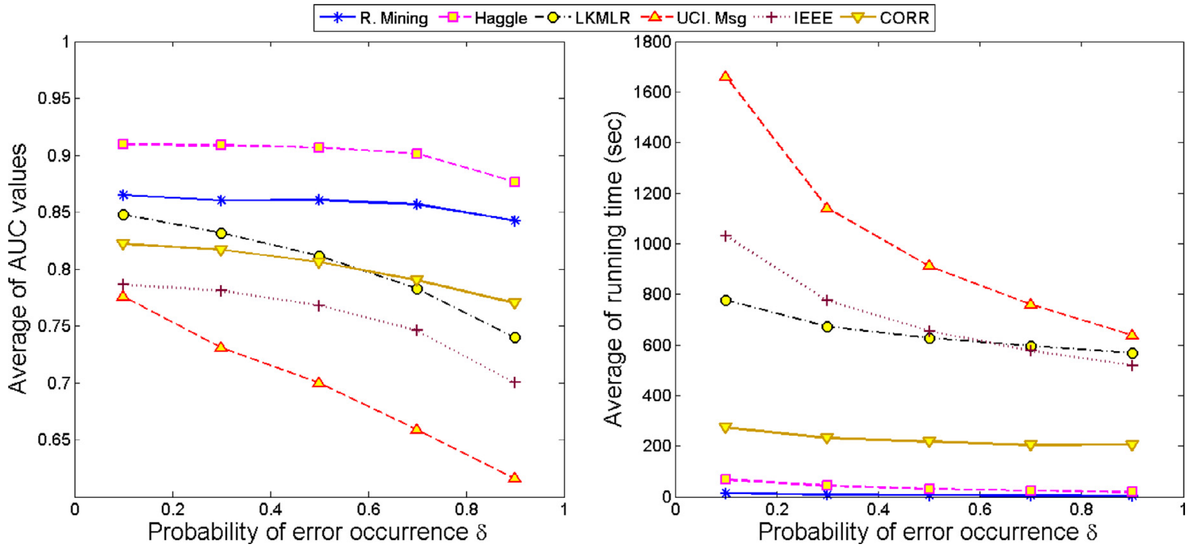

Fig. 5. Average AUC values and average running time under different δ values.

values are raised sharply on all datasets when the value of L is increased. This is because longer paths of a given node have a higher and more direct influence for predicting its future links. Because the sizes of the *R. Mining* and *Haggle* datasets are relatively small, large values of L yield slight improvements in the accuracy, while on other datasets it provides higher increment of AUC values. Alternately, smaller values of L reduce the computation time of the *TS-VLP* algorithm. Similarly, we should make a good balance between the computation time and the quality of the result by setting a proper value of L. It is also still an open problem to set a proper value of L for a given dataset. In general, such a proper value of L is also dataset-dependent. In our experiments, we set the value of L for a given data set by considering the size of the network. For a network of larger size, we set a larger L value to obtain a larger sub-graph radius and cover more global information. For instance, we set the value of L to 10 in the smaller networks of *R. Mining* and *Haggle* and set L to 15 for the larger datasets *LMKLR* and *I. Msg*.

Finally, in the fourth part of our experiments, the influence of the error occurrence probability δ is tested and analyzed. In this part, we fix the length of sampled paths $L=10$ and the error bound $\varepsilon=0.1$ on the *R. Mining* and *Haggle* datasets, while on the large datasets *LMKLR* and *I. Msg*, we set $L=15$ and $\varepsilon=0.05$. Fig. 5 shows the AUC values, and running times of the *TS-VLP* method under different values of δ for all datasets. It is clear that the accuracy of the method decreases at small values of δ. A smaller value of δ can ensure that the size of the sub-graph constructed is large enough to consider all valuable information. Using large values of δ, the size of the sub-graph is relatively small, and longer paths connected to the centered node could be discarded. Because the lack of such paths leads to the loss of useful information for link prediction, the accuracy of *TS-VLP* suffers relatively higher decrease at large values of δ. Alternately, smaller values of δ increase the computation time of the *TS-VLP* algorithm. This is because the size of a sampled sub-graph is exponentially in reverse proportion to δ. In our experiments, because the *I. Msg* network is denser than *LKMLR*, the implementation of *TS-VLP* on *I. Msg* is slower than the implementation on *LKMLR*, although the size of *LKMLR* is the largest. Similar to other parameters we have studied, we should create a good balance between the computation time and quality of the result by setting a proper value of δ. In general, such a proper value of δ is dataset-dependent. In our experiments, we set the size of δ according to the density of the network.

7. Conclusions and future works

We investigated the problem of link prediction in temporal networks. In this work, we present a fast link prediction algorithm *TS-VLP* that can achieve high-quality prediction results via random walks in temporal networks. The method first constructs a sub-graph centered at each node. To confine the error of the estimated *SRW* similarities within a given threshold ε, we select a proper size of such sub-graph using the Vapnik–Chervonenkis dimension. The computation time can be greatly reduced, since the algorithm *TS-VLP* estimates the similarity score only within a small sub-graph. While other quasi-global index-based methods require $O(n^3)$ time, our algorithm can obtain high-quality results in only $\mathrm{O}(n^2)$ time. For a similarity-based prediction, $\mathrm{O}(n^2)$ is the lower bound of computation time. Therefore, algorithm *TS-VLP* reached the optimal time complexity. Experimental results on several real networks have shown that our algorithm *TS-VLP* can achieve superior computational efficiency while keeping high accuracy of link prediction results.

One challenge in our algorithm is to set proper parameter values to create a good balance between the quality of the results and the computational time. However, such proper value highly depends on the dataset and the application. Generally, for larger and denser networks, we set smaller values of ε and δ and a larger value of L to cover more global information. For applications requiring high processing speeds, ε and δ should be assigned larger values, and L should be relatively small. For applications requiring high accuracy of results, smaller ε and δ values and a larger L value should be set. It is our future work to find an efficient way to obtain the optimal values of ε, δ and L for a given dataset.

Acknowledgments

This research was supported in part by the Chinese National Natural Science Foundation under grant Nos. 61379066, 61070047, 61379064, and 61472344, the Natural Science Foundation of Jiangsu Province under contracts BK20130452, BK2012672, and BK2012128, and the Natural Science Foundation of Education Department of Jiangsu Province under contracts 12KJB520019, 13KJB520026, and 09KJB20013.

References

[1] H.S. Ahmed, B.M. Faouzi, J. Caelen, Detection and classification of the behavior of people in an intelligent building by camera, Int. J. Smart Sensing Intel. Syst. 6 (4) (2013) 1317–1342.
[2] L.M. Aiello, A. Barrat, R. Schifanella, Friendship prediction and homophily in social media, *ACM Trans. Web* (TWEB) 6 (2) (2012) 9.
[3] I. Antonellis, H.G. Molina, C.C. Chang, Smrank++: query rewriting through link analysis of the click graph, PVLDB 1 (1) (2008) 408–421.
[4] Z.F. Bao, Y. Zeng, Y.C. Tay, LP son, social network link prediction by principal component regression, in: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2013 August.
[5] N. Barbieri, F. Bonchi, G. Manco, Who to follow and why: link prediction with explanations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 1266–1275.
[6] C.A. Bliss, Morgan R. Frank, C.M. Danforth, P.S. Dodds, An evolutionary algorithm approach to link prediction in dynamic social networks, J. Comput. Sci. 5 (5) (2014) 750–764 September.
[7] B. Bringmann, M. Berlingerio, F. Bonchi, A. Gionis, Learning and predicting the evolution of social networks, IEEE Intell. Syst. (2010) 26–34.
[8] A. Chaintreau, P. Hui, J. Crowcroft, C. Diot, R. Gass, J. Scott, Impact of human mobility on opportunistic forwarding algorithms, IEEE Trans. Mob. Comput. 6 (6) (2007) 606–620.
[9] B. Chen, L. Chen, B. Li, A fast algorithm for predicting links to nodes of interest, Inf. Sci. 329 (2016) 552–567.
[10] N. Eagle, A. Pentland, Reality mining: sensing complex social systems, Person. Ubiquitous Comput. 10 (4) (2006) 255–268.
[11] S. Gao, L. Denoyer, P. Gallinari, Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks, J. China Univ. Posts Telecommun. 19 (2012) 172–181.
[12] S. Gao, L. Denoyer, P. Gallinari, Temporal link prediction by integrating content and structure information, in: CIKM'11, , Glasgow, Scotland, UK, 2011, pp. 1169–1174.
[13] M.Q. Ge, A. Li, M.H. Wang, A bipartite network-based method for prediction of long non-coding rna–protein interactions, Genomics, Proteomics Bioinf. 14 (1) (2016) 62–71.
[14] S. Hanneke, W.J. Fu, E.P. Xing, Discrete temporal models of social networks, Electron. J. Stat. 4 (2010) 585–605.
[15] Y.L. He, J.N.K. Liu, Y.X. Hu, X.Z. Wang, OWA operator based link prediction ensemble for social network, Expert Syst. Appl. 42 (1) (2015) 21–50.
[16] F.Y. Hu, H.S. Wong, Labelling of human motion based on CBGA and probabilistic model, Int. J. Smart Sensing Intel. Syst. 6 (2) (2013) 583–609.
[17] G. Jeh, J. Widom, Scaling personalized web search. WWW, (2003), 271–279.
[18] X. Jia, F. Xin, W.R. Chuan, Adaptive spray routing for opportunistic networks, Int. J. Smart Sensing Intell. Syst. 6 (1) (2013) 95–119.
[19] B. Kaya, M. Poyraz, Supervised link prediction in symptom networks with evolving case, Measurement 56 (2014) 231–238.
[20] B. Kaya, M. Poyraz, Unsupervised link prediction in evolving abnormal medical parameter networks, Int. J. Mach. Learn. Cybern. 7 (1) (2016) 145–155.
[21] H. Kim, J. Tang, R. Anderson, C. Mascolo, Centrality prediction in dynamic human contact networks, Comput. Netw. 56 (2012) 983–996.
[22] P. Klimek, A.S. Jovanovic, R. Egloff, R. Schneider, Successful fish go with the flow: citation impact prediction based on centrality measures for term—document networks, Scientometrics 107 (3) (2016) 1265–1282.
[23] Y. Li, P.M. Long, A. Srinivasan, Improved bounds on the sample complexity of learning, J. Comput. Syst. Sci. 62 (3) (2001) 516–527.

[24] Y.H. Li, A.M. Wen, Q. Lin, R.X. Li, D. Lu, Name disambiguation in scientific cooperation network by exploiting user feedback, Artif. Intell. Rev. 41 (4) (2014) 563–578.
[25] J. Liu, G. Denga, Link prediction in a user_object network based on time-weighted resource allocation, Physica A 388 (2009) 3643–3650.
[26] W. Liu, L. Lü, Link prediction based on local random walk, Europhys. Lett. 89 (5) (2010) 58007.
[27] Z.H. Liu, J.F. Ma, Y. Zeng, Secrecy transfer for sensor networks: from random graphs to secure random geometric graphs, Int. J. Smart Sensing Intell. Syst. 6 (1) (2013) 77–94.
[28] L. Lü, C.H. Jin, T. Zhou, Similarity index based on local paths for link prediction of complex networks, Phys. Rev. E 80 (2009) 046122.
[29] L.Y. Lü, T. Zhou, Link prediction in complex networks: a survey, Physica A 390 (2011) 1150–1170.
[30] T. Murata, S. Moriyasu, Link prediction based on structural properties of online social networks, New Generat. Comput. 26 (3) (2008) 245–257.
[31] D. Nanongkai, A.D. Sarma, G. Pandurangan, A tight unconditional lower bound on distributed random walk computation, PODC (2011) 257–266.
[32] A. Nigam, N.V. Chawla, Link prediction in a semi-bipartite network for recommendation, Lect. Notes Comput. Sci. 9622 (2016) 127–135.
[33] J. O'Madadhain, J. Hutchins, P. Smyth, Prediction and ranking algorithms for event-based network data, in: Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2005, pp. 23–30.
[34] T. Opsahl, P. Panzarasa, Clustering in weighted networks, Soc. Netw. 31 (2) (2009) 155–163.
[35] M. Pujari, R. Kanawati, Supervised rank aggregation approach for link prediction in complex networks, in: WWW 2012 Companion, Lyon, France, 2012, pp. 1189–1196.
[36] S. Purnamrita, D. Chakrabarti, and M. Jordan. Nonparametric link prediction in dynamic networks, *arXiv preprint arXiv*, (2012), 1206–6394.
[37] A.D. Sarma, A.R. Molla, G. Pandurangan, Near-optimal random walk sampling in distributed networks, INFOCOM (2012) 2906–2910.
[38] A.D. Sarma, A.R. Molla, G. Pandurangan, Distributed computation in dynamic networks via random walks, Theor. Comput. Sci. 58 (1) (2015) 45–66.
[39] A.D. Sarma, D. Nanongkai, G. Pandurangan, Fast distributed random walks, PODC (2009) 161–170.
[40] A.D. Sarma, D. Nanongkai, G. Pandurangan, P. Tetali, Efficient distributed random walks with applications, PODC (2010) 201–210.
[41] A.D. Sarma, D. Nanongkai, G. Pandurangan, P. Tetali, Distributed random walks, J. ACM 60 (1) (2013) 2.
[42] M. Savić, M. Ivanović, M. Radovanović, Z. Ognjanović, A. Pejović, T.J. Krüger, The structure and evolution of scientific collaboration in Serbian mathematical journals, Scientometrics 101 (3) (2014) 1805–1830.
[43] E. Sherkat, M. Rahgozar, M. Asadpour, Structural link prediction based on ant colony approach in social networks, Physica A 419 (1) (2015) 80–94.
[44] D. Sun, T. Zhou, J.G. Liu, R.R. Liu, C.X. Jia, B.H. Wang, Information filtering based on transferring similarity, Phys. Rev. E 80017101 (2009).
[45] D.Q. Vu, A.U. Asuncion, D.R. Hunter, P. Smyth, Continuous-time regression models for longitudinal networks, in: Advances in Neural Information Processing Systems 24: *Proceedings of the 25th Annual Conference on Neural Information Processing Systems*, 2011, pp. 1–9.
[46] P. Wang, B.W. Xu, Y.R. Wu, X.Y. Zhou, Link prediction in social networks: the state-of-the-art, Sci. China Inf. Sci. 58 (1) (2015) 1–38.
[47] X.M. Wang, Y. Liu, F. Xiong, Improved personalized recommendation based on a similarity network, Physica A 456 (15) (2016) 271–280.
[48] L.T. Yang, S. Wang, H. Jiang, Cyclic temporal network density and its impact on information diffusion for delay tolerant networks, Int. J. Smart Sensing Intell. Syst. 4 (1) (2011) 35–52.
[49] Z.Z. Zeng, K.J. Chen, S.B. Zhang, H.J. Zhang, A link prediction approach using semi-supervised learning in dynamic networks, in: 2013 *Sixth International Conference on Advanced Computational Intelligence* (ICACI), 2013, pp. 276–280.
[50] T. Zhou, L. Lü, Y.C. Zhang, Predicting missing links via local information, Eur. Phys J B 71 (4) (2009) 623–630.