Information Sciences 331 (2016) 120–136

Contents lists available at ScienceDirect

Information Sciences

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

An efficient algorithm for link prediction in temporal uncertain social networks

Nahla Mohamed Ahmed [a,b,*], Ling Chen [a,c,1]

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

ARTICLE INFO

Article history:
Received 23 April 2015
Revised 15 October 2015
Accepted 25 October 2015
Available online 30 October 2015

Keywords:
Temporal uncertain network
Possible world
Link prediction
Random walk

ABSTRACT

Due to the inaccuracy, incompleteness and noise in data from real applications, uncertainty is a natural feature of real-world networks. In such networks, each edge is associated with a probability value indicating its existence in the network. Predicting links in uncertain networks is computationally more challenging and differs semantically from predicting connections in deterministic networks. This paper presents a method for link prediction in temporal uncertain networks. In our method, the predicting problem is formalized by designing a random walk in temporal uncertain networks. The algorithm first transforms the link prediction problem in uncertain networks to a random walk in a deterministic network. Then, the similarity scores between a node and its neighbors are computed within a sub-graph around this node to reduce the computational time. The proposed method integrates temporal and global topological information in temporal uncertain networks and can obtain more accurate results. Experimental results on real social networks show that our method can predict future links efficiently in temporal uncertain social networks and achieves higher quality results than other similar methods.

1. Introduction

Networks can naturally describe various social structures. In such networks, vertices denote entities and links represent communications or relations between the entities. A social network consists of individuals and relations such as friendship or partnership. The analysis of social networks has drawn increasing attention in the field of sociology. It analyzes and explores the potential relations between social objects. In recent years, social network analysis has also attracted a great deal of interest in many business fields, such as e-business analysis and market modeling.

One of the most important research areas in network analysis is link prediction. The objective of link prediction is to detect unobserved links from existing parts of the network or forecast future links from current structures of the network. In a social security network, link prediction is used to identify hidden groups of terrorists or criminals [30], while in networks of human behavior, link prediction is used to detect and classify the behavior and motion of people [3]. Link prediction also has many applications in networks reflecting social relations such as communication networks, email networks and sensor networks. In

* Corresponding author at: College of Information Engineering, Yangzhou University, Yangzhou 225009, China. Tel.: +86 13235277310.
E-mail addresses: nahlaibrahim@uofk.edu (N.M. Ahmed), yzulchen@163.com (L. Chen).
[1] Tel.: +86 13373695018.

http://dx.doi.org/10.1016/j.ins.2015.10.036
0020-0255/

sensor networks, link prediction is used to explore dynamic temporal properties [41], ensure information transfer secrecy [26], and realize optimal routing [21].

Because relations among social members change continuously over time, links in real world social networks are constantly varying and evolving. New links may appear and existing links may vanish from the network. For example, email communications between friends, transactions between businesses, and partnerships between scientific researchers are changing over time. Therefore, link prediction algorithms should be capable of detecting dynamic relationships between members in a temporal social network.

Some social networks often have inherent uncertainty; these are referred to as uncertain social networks. Due to inaccuracy, incompleteness and noise in real applications, uncertainty is a natural feature in real world networks. In social networks, the likelihood of social interactions and relations may be affected by trust and influence issues [1,16,24]. In a telecommunications or e-mail network, communications between social organizations or individuals may occur randomly [15,36]. The probabilistic graph [2,19] model is an appropriate structure to describe such uncertainty features [6,33,39,42,45,46,47]. Each edge in this model is assigned a probability to indicate the likelihood of its existence in the network. Generally, the structural features of a network can be represented by a probabilistic function of the uncertainty on the edges and the topology features of the network. Due to the large number of possible instantiations in an uncertain network [6,39], link prediction based on such a model is extremely complicated. For instance, the problem of reachability between vertexes in deterministic networks requires only linear time, but in uncertain networks, it is a #*P*-complete problem [6]. Therefore, it is necessary to design an efficient approach to predict the potential links in temporal uncertain networks.

In this paper, we investigate the problem of link prediction in dynamic uncertain networks. In our work, the link prediction problem is formalized by designing a new method based on a random walk in temporal uncertain networks. Our method first transforms the link prediction problem in uncertain networks to a random walk in a deterministic network. Then, the similarity scores between a node and its neighbors are calculated within a sub-graph around this node to reduce the computational time. We also extend the method for solving link prediction in temporal uncertain networks. The proposed method integrates time and global topological information and can obtain more accurate results. Experimental results on real social networks show that our method can predict future links efficiently in temporal uncertain social networks and achieves higher quality results than other similar methods.

The rest of this paper is organized as follows: Section 2 reviews related work on link prediction in complex networks. Section 3 provides the problem definition and some important concepts. Section 4 presents the random walk method for link prediction in static uncertain networks, while Section 5 gives a time series random walk model in temporal uncertain networks. In Section 6, we present and analyze the experimental results of our link prediction methods on real datasets. Finally, Section 7 offers conclusions and possibilities for future work.

2. Related works

Recently, various approaches have been proposed to detect potential or future links in temporal social networks.

The similarity-based method is the most common method for link prediction. In this method, each node pair is associated with an index to indicate the similarity between 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 the existence of many common features or topological structures between the nodes [38]. Many studies in social networks show the existence of a relative similarity between individuals who are close to each other [4,13]. Structural similarity indices are often used in popular similarity-based methods. These similarity indices can be divided into three classes: local indices, quasi-local indices, and global indices. Local indices require only the neighbor information of the nodes, for example, Common Neighbors, Jaccard, Salton, Sorensen, Preferential Attachment, Hub Depressed, Hub Promoted, Adamic-Adar, Resource Allocation, and Leicht–Holme–Newman (LHN1) indices [30]. Global indices require comprehensive information for link prediction tasks. They use the global topological information of networks; such indices include Katz, Matrix Forest Index (MFI), and Leicht–Holme–Newman (LHN2) [30]. Quasi-local indices require more structural information than local indices and less information than global indices. Such indices include Local Random Walk (LRW), Superposed Random Walk (SRW) [28], and Local Path Index [29,44]. In general, global indices provide more accurate prediction compared with local indices, although the latter require less information than the former. Another class of similarity-based methods is random walk methods, which include SimRank, Random Walk with Restart, Cos+, and Average Commute Time [30].

Some of these methods are based on analysis of the topological features of the network. Purnamrita et al. [35] proposed a nonparametric method for link prediction in dynamic networks. This method partitioned the time domain into subsequences that are represented by graph snapshots. Their method predicts connections between the nodes based on their topological features and local neighbors. Murata et al. [31] explored the connection between link prediction and graph topology. They advanced a weighted proximity-based method for link prediction in social networks. Kim et al. [25] presented a method to predict future network topology using node centrality, which can identify important nodes in the future.

Machine learning strategies have also been exploited in temporal network link prediction methods. Pujari et al. [34] applied a supervised rank aggregation method for link prediction in temporal complex networks. Vu et al. [40] introduced a continuous-time regression model for temporal network link prediction. This model can be incorporated with both time-varying regression coefficients and time-dependent network statistics. Zhengzhong Zeng et al. [43] presented a method using semi-supervised learning in link prediction tasks to utilize the potential information in a large number of unlinked node pairs in networks. Yu-lin

He et al. [18] proposed a link prediction ensemble algorithm based on 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. Zhifeng Bao et al. [7] advanced a network link predictor using principal component analysis to identify features that are important to link prediction. Bringmann et al. [11] proposed an approach for link prediction in temporal networks based on techniques for association-rule mining and frequent-pattern detection. O'Madadhain et al. [32] presented a link prediction method for event-based temporal networks. Using techniques for data mining and machine learning, this method can predict future co-participation of individuals in social events. To avoid a high computational cost of optimization in the machine learning methods, some heuristic methods are employed in link prediction. Ehsan Sherkat et al. [37] introduced an unsupervised structural link prediction algorithm based on ant colony optimization. A. Catherine et al. [10] proposed an approach to predicting future links by applying the covariance matrix adaptation evolution strategy.

Some methods for link prediction in networks are based on a probabilistic model. Hanneke et al. [17] proposed a family of statistical models for dynamic social network link prediction by extending the exponential random graph model. Ji Liu et al. [27] presented a method for link prediction in a user-object network. The method takes both time attenuation and diversion delay into consideration. Gao et al. [14] proposed a model that exploits multiple information sources in the dynamic network to obtain link occurrence probabilities. Nicola Barbieri et al. [8] presented 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. Hu et al. [20] presented a probabilistic model to detect human motion in social networks and advanced a method for labeling human motion using constraint-based genetic algorithm to optimize the model. However, such a probabilistic model requires a predefined distribution of link appearance, which is difficult to know in advance for a given network.

Link prediction in uncertain networks has become an important research issue. To predict links in uncertain networks, a similarity index is assigned to each pair of nodes that is proportional to the possibility of the link between the two nodes. Such a similarity index can be computed from uncertainties given by the graph. Asthana et al. [5] models a protein–protein interaction (PPI) network as an uncertain network for predicting the existence of a protein in an incomplete protein complex. Ghosh et al. [15] studied a wireless network and treated it as an uncertain network for designing routing protocols by extracting the most potential delivery sub-graph. Kang [22] proposed a spatiotemporal uncertain network (STUN) model to formally define probabilistic social networks and presented the concept of STUN sub-graph matching queries. In [23], Potamias et al. presented a method for computing the *K*-nearest neighbors in uncertain networks to reduce the time to compute the similarity between node pairs. However, most of these methods are based on the concept of a possible world, and it requires a substantial amount of time to enumerate all of the possible worlds for the purpose of computing the similarities between all node pairs.

3. Concepts and definitions

Definition 1. *Uncertain network*– An uncertain network can be represented by a graph $G = (V, E, P, A)$, where V is the set of nodes in G, and $|V| = n$. E is the set of edges of G. P and A are $n \times n$ matrices that denote the probability matrix and the adjacent matrix of graph G, respectively. Each edge $e \in E$ is associated with probability $P(e)$.

Definition 2. *Link prediction in a temporal uncertain network*– Given snapshots of an uncertain network $G_t = (V, E_t, P_t, A_t)$ at time $t = t_0, t_0 + 1, \ldots, t_0 + T - 1$, where T is the *window size*. The time series link prediction problem in uncertain networks is to predict the occurrence probabilities of edges at time $t_0 + T$.

In this work, we propose a similarity-based method to predict future links in a temporal uncertain network. The output of our proposed method is an $n \times n$ matrix S, in which $S(i, j)$ is a similarity index to that indicates the likelihood of edge (v_i, v_j) appearing at time $t_0 + T$. The similarity measure we use is based on the SimRank similarity score.

SimRank [30] is one of the link-based similarity measures in link prediction; it has a tremendous influence and a wide range of applications. SimRank similarity can be interpreted as the mean distance for two surfers from two nodes randomly walking to their first meet. Unlike other similarity indexes, the SimRank index is free from any domain constraints and can effectively deal with object-to-object interactions. Moreover, SimRank can be applied to both direct and indirect connections in the network.

Definition 3. *SimRank index*– The SimRank similarity between nodes a and b, which is denoted by $S(a, b) \in [0, 1]$, can be computed recursively by the following formula:

$$S(a,b) = \begin{cases} \dfrac{c}{|I(a)|.|I(b)|} \displaystyle\sum_{i=1}^{|I(a)|} \sum_{j=1}^{|I(b)|} S(I_i(a), I_j(b)), & a \neq b \\ 1 & a = b \end{cases} \tag{1}$$

where $c \in (0, 1)$ is a constant, $I(u)$ is the set of neighbor nodes of u, and $|I(u)|$ is the number of nodes in $I(u)$. Elements in $I(u)$ are referred to as $I_i(u)$, $1 \leq i \leq |I(u)|$.

Initially, the value of $S(a,b)$ is set as $S(a,b) = A(a,b)$, i.e., $S(a, a) = 1$, and $S(a,b) = 0$ for $a \neq b$. In the special case where $I(a)$ and $I(b)$ are empty sets, $S(a,b)$ is set to zero.

The matrix form of Eq. (1) for computing the SimRank index is

$$S = cW^T SW + (1 - c)I \tag{2}$$

Here, $W = [w_{ij}]$ is the transformation matrix, which can be obtained by normalizing each column of adjacent matrix A, namely, $w_{ij} = a_{ij} / \sum_{k=1}^{n} a_{kj}$. From (2), we can see that the time complexity for computing the SimRank index of a network with n nodes is $O(n^3)$.

4. Random walk in a static uncertain networks model

In this section, we introduce our prediction method in static uncertain networks based on a random walk and the SimRank index. We first give the definition of the possible world for the uncertain network. Then, we define a probabilistic random walk in probabilistic graphs based on a possible world. We present an efficient method for computing a probabilistic random walk using a sub-graph around a given node.

Definition 4. *Possible world*– Let $G = (V,E,A,P)$ be a probabilistic graph and G' be a sub-graph of G associated with the probability P; i.e., the probability of each edge $e \in E$ to be included in G' is $P(e)$. We call G' a possible world of G.

Each possible world G' is sampled with probability $P_r[G']$:

$$P_r\left[G'\right] = \prod_{e \in E'} p(e) \prod_{e \in E \setminus E'} (1 - p(e)) \tag{3}$$

Here, E' is the set of edges of G'. Obviously, there are $2^{|E|}$ possible worlds in an uncertain network G with $|E|$ edges. Let $(u, v) \in E$ be an edge in uncertain network G; we use $\Omega(u,v)$ to denote the set of all possible worlds containing edge (u,v). Obviously, there are $2^{|E|-1}$ possible worlds in $\Omega(u,v)$.

To predict links in an uncertain network using the SimRank index, we first transform the problem into a link prediction in a deterministic network. In [23], the following theorem is proposed to provide an equivalent definition for the random walk and the SimRank index in an uncertain network based on the concept of a possible world. Based on this theorem, the random walk on uncertain network G can be transformed into a standard random walk process on a deterministic network $\overline{G} = (V, \overline{E}, \overline{W})$ with a transformation matrix $\overline{W}$.

Theorem 1. *[23] For a given probabilistic graph $G = (V,E,A,P)$ and a deterministic graph $\overline{G} = (V, \overline{E}, \overline{W})$, the probabilistic random walk on G and standard random walk on $\overline{G}$ have the same properties, where $\overline{W}$ is the transformation matrix and $\overline{E} = E \cup S$ with $S = \{(u, u)\}$ as the set of all self-looping edges in G. The transformation matrix $\overline{W} = \{\overline{w}(u, v)\}$ is defined as*

$$\begin{aligned} \overline{w}(u,u) &= \prod_{(u,q) \in E} (1 - p(u,q)) \\ \overline{w}(u,v) &= \sum_{G' \in \Omega(u,v)} \frac{A(u,v)}{\sum_{(u,q) \in G'} A(u,q)} \Pr[G'] \quad (u \neq v) \end{aligned} \tag{4}$$

From Theorem 1, we can see that the probabilistic random walk on uncertain graph G and the standard random walk on deterministic graph $\overline{G} = (V, \overline{E}, \overline{W})$ with a transformation matrix $\overline{W}$have the same stationary distribution. Thus, once we have computed transformation matrix $\overline{W}$, we can apply standard random walk techniques to solve the stationary distribution problem and compute the SimRank index between node pairs. Applying $\overline{W}$ on the traditional SimRank index computation, we can obtain the SimRank index in uncertain networks by the following matrix computation:

$$S = c\overline{W}^T S\overline{W} + (1 - c)I \tag{5}$$

The complexity of computing each weight $\overline{W}(u, v)$using formula (4) is $O(2^{|E|-1})$. This is intractable for a large-scale network. Therefore, we need an efficient method to compute the elements in transformation matrix $\overline{W}$.

4.1. Computing the transformation matrix $\overline{W}$

Our link prediction method for an uncertain network consists of two steps. First, calculate the transformation matrix $\overline{W}$ as defined by (4), and then calculate the SimRank by (5).

Because there are $2^{|E|-1}$ possible worlds in $\Omega(u,v)$, a substantial amount of time is required to calculate the transformation matrix element $\overline{W}(u, v)$ in $\overline{W}$ according to (4) by enumerating all of the possible worlds in $\Omega(u,v)$. We give an efficient method to calculate the transformation matrix $\overline{W}$.

Definition 5. *Sub-graph around a node* – Let u be a node in $\overline{G} = (V, \overline{E}, \overline{W})$ and $\Gamma(u) = \{v|(u, v) \in \overline{E}\}$ be the set of neighbor nodes of u, and the sub-graph around u is defined as $G(u) = (V(u), E(u), \overline{W}_u)$, here $V(u) = \{u\} \cup \Gamma(u)$,$E(u) = \{(w, v)|(w, v) \in \overline{E}, v, w \in V(u)\}$, $\overline{W}_u$is the transformation matrix of $G(u)$.

We give the following theorem to show that the transformation weight $\overline{W}(u, v)$ can be calculated within $G(u)$, the sub-graph around u, instead of enumerating all of the $2^{|E|-1}$ possible worlds of $\Omega(u,v)$ in the whole network G.

Theorem 2. *Let u be a node in G; the transformation weight from u to another node v in G can be calculated by*

$$\overline{W}(u,v)=\begin{cases}\overline{W}_u(u,v) & \text{if } v\in G(u)\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

Proof. Let $\overline{A}$ be the adjacent matrix of $\overline{G}$.

Case 1 If $v\notin G(u)$, because$(u,v)\notin E\subset\overline{E}$, the adjacent matrix element $\overline{A}(u,v)=A(u,v)=0$. Therefore,

$$\overline{w}(u,v)=\sum_{G'\in\Omega(u,v)}\frac{A(u,v)}{\sum_{(u,q)\in G'}A(u,q)}\Pr[G']=0$$

.

Case 2: If $v\in G(u)$ and $v\neq u$, then $\overline{A}(u,v)=A(u,v)$. We define the set of edges outside $G(u)$ as $\overline{E}(u)=\{e|e\notin G(u),e\in\overline{E}\}$.

Suppose $G(u)$ has m possible worlds with edge (u,v): $G_1(u)$, $G_2(u),\ldots,G_m(u)$. For a possible world $G_i(u)$ $(i=1,2,\ldots,m)$, we define the set of edges in $G_i(u)$ as $E_i(u)=\{e|e\in G_i(u),e\in\overline{E}\}$. Denote the set of edges in $G(u)$ but not in $G_i(u)$ as $E'_i(u)=\{e|e\in E(u),e\notin G_i(u),e\in\overline{E}\}$.

For the possible world $G_i(u)$, we can define a possible world of $\overline{G}$ as $\overline{G}_{ik}(u)=E_i(u)\cup\overline{E}_{ik}(u)$, where $\overline{E}_{ik}(u)\in\overline{E}(u)$is a set of edges not in $E(u)$. We call possible world $\overline{G}_{ik}(u)$ an extension of $G_i(u)$. Suppose there are a total of m_i extensions of $G_i(u)$, named $\overline{G}_{i1}(u)$ $\overline{G}_{i2}(u),\ldots,\overline{G}_{im_i}(u)$.

For the possible world $\overline{G}_{ik}(u)=E_i(u)\cup\overline{E}_{ik}(u)$, we define the set of edges $\overline{E}'_{ik}(u)=\{e|e\in\overline{E}(u),e\notin\overline{E}_{ik}(u)\}$.

Then, by (4) we have

$$\overline{w}(u,v)=\sum_{G'\in\Omega(u,v)}\frac{A(u,v)}{\sum_{(u,q)\in G'}A(u,q)}P_r[G']=\sum_{i=1}^{m}\sum_{k=1}^{m_i}\frac{A(u,v)}{\sum_{(u,q)\in G_{ik}}A(u,q)}P_r[G_{ik}]$$

Because

$$P_r[G_{ik}]=\coprod_{e\in G_{ik}}p_r(e)\prod_{e\notin G_{ik}}\left[1-p_r(e)\right]=\prod_{e\in E_i(u)}p_r(e)\prod_{e\in\overline{E}_{ik}(u)}p_r(e)\prod_{e\in E'_i(u)}\left[1-p_r(e)\right]\prod_{e\in\overline{E}'_{ik}(u)}\left[1-p_r(e)\right],$$

Then,

$$\overline{w}(u,v)=\sum_{i=1}^{m}\frac{A(u,v)}{\sum_{(u,q)\in G_i(u)}A(u,q)}\prod_{e\in E_i(u)}p_r(e)\prod_{e\in E'_i(u)}[1-p_r(e)]\sum_{k=1}^{m_i}\prod_{e\in\overline{E}_{ik}(u)}p_r(e)\prod_{e\in\overline{E}'_{ik}(u)}[1-p_r(e)]$$

Because $\sum_{k=1}^{m_i}\prod_{e\in\overline{E}_{ik}(u)}p_r(e)\prod_{e\in\overline{E}'_{ik}(u)}[1-p_r(e)]$ is the summation of the probabilities of all possible worlds in $\overline{E}(u)=\{e|e\notin G(u),e\in\overline{E}\}$, we then have

$$\sum_{k=1}^{m_i}\prod_{e\in\overline{E}_{ik}(u)}p_r(e)\prod_{e\in\overline{E}'_{ik}(u)}[1-p_r(e)]=1$$

Therefore,

$$\overline{w}(u,v)=\sum_{i=1}^{m}\frac{A(u,v)}{\sum_{(u,q)\in G_i(u)}A(u,q)}\prod_{e\in E_i(u)}p_r(e)\prod_{e\in E'_i(u)}[1-p_r(e)]=\sum_{i=1}^{m}\frac{A(u,v)}{\sum_{(u,q)\in G_i(u)}A(u,q)}p_r(G_i(u))=\overline{W}_u(u,v)\qquad\square$$

Theorem 2 shows that $\overline{W}(u,v)$can be calculated within the sub-graph $G(u)$, which is much smaller than the whole network G.

4.2. Calculate $\overline{W}_u(u,v)$ in sub-graph G(u)

To calculate $\overline{W}_u(u,v)$ in sub-graph $G(u)$, it still requires a large amount of time to enumerate all of the possible worlds in sub-graph $G(u)$. Therefore, we advance an efficient way to enumerate all of the possible worlds in $G(u)$.

Let $S\subset E(u)$ be a subset of edges in $G(u)$ and $p(k,S)=\sum_{\substack{S_{(k)}\subset S\\|S_{(k)}|=k}}p(S_{(k)})$ be the summation of the possibilities of all subsets with exactly k edges in S, here $k\leq|S|$. Then, we have the following theorem:

Theorem 3. *Let $S_1 \subset E(u)$ and $S_2 \subset E(u)$ be two subsets of edges in $G(u)$, $S_1 \cap S_2 = \Phi$; then, we have*

$$p(k, S_1 \cup S_2) = \sum_{g=0}^{\min(k,|S_1|)} \big[p(g, S_1) * p(k-g, S_2)\big]. \tag{6}$$

Proof. We divide $S_1 \cup S_2$ into m subsets, where $m = \min\{k, |S_1|\}$. The g-th subset is defined as:

$$S^{(g)} = \{\{e_1, \ldots, e_g, \ldots, e_k\} | e_1, \ldots, e_k \in S_1 \cup S_2 \wedge \{e_1, \ldots, e_g\} \subseteq S_1 \wedge \{e_{g+1}, \ldots, e_k\} \subseteq S_2\}$$

Then, $p(S^{(g)}) = p(g, S_1) \times p(k-g, S_2)$, $g = 1,2,\ldots,m$. Therefore, we have

$$p(k, S_1 \cup S_2) = \sum_{g=0}^{m} p(S^{(g)}) = \sum_{g=0}^{\min(k,|S_1|)} [p(g, S_1) * p(k-g, S_2)] \qquad \square$$

Let the edges in $G(u)$ be $e_1, e_2, \ldots, e_D$, where $e_1, e_2, \ldots, e_d$ are the edges connecting u with nodes $v_1, v_2, \ldots, v_d$. Denote the set of edges $\{e_i, e_{i+1}, \ldots, e_{i+l-1}\}$ as $S_i(l)$. Let $p(k,S_i(l))$ be the summation of the possibilities of all possible worlds with exactly k edges in $S_i(l)$. Based on Theorem 3, we use dynamic programming to compute all of the $p(k,S_i(l))$ values in $G(u)$ for $k = 0,1,\ldots, D$, $i = 1,2,\ldots,D\text{-}1$, $l = 1,2,\ldots, D-i$. Once we have calculated all of the $p(k,S_i(l))$ values, $\overline{W}_u(u,v)$can be computed by the following theorem:

Theorem 4. *Let $e = (u,v)$ be an edge in $G(u)$; we then have*

$$\overline{W}_u(u,v) = \sum_{e \in S_i(l)} \sum_{k=1}^{D} \sum_{i=1}^{D-1} \sum_{l=1}^{D-i} \frac{P(k, S_i(l))}{k} \tag{7}$$

Proof. Suppose there are m possible worlds containing edge (u,v) in $G(u)$, which are denoted by $G_1(u)$, $G_2(u)$,..., $G_m(u)$. By (4), we know that

$$\overline{W}_u(u,v) = \sum_{j=1}^{m} \frac{A(u,v)}{\sum_{(u,q) \in G_j(u)} A(u,q)} p_r(G_j(u))$$

We divide the set of $\{G_1(u) \cup G_2(u) \cup \ldots \cup G_m(u)\}$ into D groups $H_1(u)$, $H_2(u)$,..., $H_D(u)$, where every possible world in $H_k(u)$ has exactly k edges ($k = 1,2, .., D$). Then, $\bigcup_{j=1}^{D} H_j(u)$ includes all of the m possible worlds with edge (u,v) in $G(u)$. It is obvious that $P(H_k(u)) = \sum_{e \in S_i(l)} \sum_{i=1}^{D-1} \sum_{l=1}^{D-i} p(k, S_i(l))$. Then, we have

$$\begin{aligned}\overline{W}_u(u,v_i) &= \sum_{j=1}^{m} \frac{A(u,v_i)}{\sum_{(u,q) \in G_j(u)} A(u,q)} p_r(G_j(u)) = \sum_{k=1}^{D} \frac{A(u,v)}{\sum_{(u,q) \in H_k(u)} A(u,q)} p(H_k(u)) \\ &= \sum_{k=1}^{D} \frac{A(u,v)}{\sum_{(u,q) \in H_k(u)} A(u,q)} \sum_{v \in S_i(l)} \sum_{i=1}^{D-1} \sum_{l=1}^{D-i} p(k, S_i(l)) = \sum_{e \in S_i(l)} \sum_{k=1}^{D} \sum_{i=1}^{D-1} \sum_{l=1}^{D-i} \frac{P(k, S_i(l))}{k}\end{aligned} \qquad \square$$

By Theorems 2 and 4, we also have

$$\overline{W}(u,v) = \overline{W}_u(u,v) = \sum_{e \in S_i(l)} \sum_{k=1}^{D} \sum_{i=1}^{D-1} \sum_{l=1}^{D-i} \frac{P(k, S_i(l))}{k}.$$

Based on Theorems 3 and 4, we present the following algorithm *ComSim* (computing the similarities) for calculating the $\overline{W}$ values for all of the neighbors of u.

It is easy to see that the time complexity of algorithm *ComSim*(u) is

$$\sum_{r=1}^{l-1} \sum_{i=1}^{D/2^r} \sum_{k=1}^{2^r} 2^{r-1} = \sum_{r=1}^{l-1} \sum_{i=1}^{D/2^r} 2^r \cdot 2^{r-1} = \sum_{r=1}^{l-1} D \cdot 2^{r-1} = D \sum_{r=1}^{l-1} 2^{r-1} = D(D-1) = O(D^2)$$

Algorithm *ComSim*(u);
Input:
u: a node in G;
d: degree of node u;
v_1,v_2,…,v_d: neighbors of u;
D: number of edges in $G(u)$;
$e_1, e_2, \ldots, e_d$: the edges connecting u with v_1,v_2,…,v_d;
$e_{d+1}, e_{d+2}, \ldots, e_D$: the other edges in $G(u)$;
$p_r(u,v_1)$,…, $p_r(u,v_D)$): probability of the edges $e_1, e_2, \ldots, e_D$;
Output: $\overline{W}(u, v_i)$, $i = 1, \ldots, d$: the weights for SimRank;
Begin
For $i = 1$ **to** D **do**
$P(0, S_i(1)) = 1\text{-}p_r(u,v_i)$; $P(1, S_i(1)) = p_r(u,v_i)$;
If $i <= d$ **then** $Q(1, S_i(1)) = \{i\}$; $\overline{W}(u, v_i) = p_r(u, v_i)$ **endif**;
Endfor i;
$l = \log_2 D$;
For $r = 1$ **to** l **do**
$t = 2^r$;
For $i = 1$ **to** D-t **step** t **do**
For $k = 0$ **to** t **do**
$$p(k, S_i(t)) = \sum_{g=0}^{\min(k, |S_i(t/2))} [p(g, S_i(t/2) * p(k-g, S_{i+t/2}(t/2)];$$
$Q(k, S_i(t)) = Q(g, S_i(t/2)) \cup Q(k-g, S_{i+t/2}(t/2))$;
For $j = 1$ **to** d **do**
If $j \in Q(k, S_i(t))$**then**
$\overline{W}(u, v_j) = \overline{W}(u, v_j) + p(k, S_i(t))/k$;
End if
End for j
Endfor k
Endfor i
Endfor r;
End

Here, D is the number of edges in $G(u)$. Let the degree of u be d; it is easy to see that $D = O(d^2)$ because there are d nodes in $G(u)$. Therefore, the time complexity of algorithm *ComSim* is $O(d^4)$.

Based on algorithm *ComSim*(u), which computes $\overline{W}$ for u with all its neighbors, we present an algorithm named *LPUN* (link prediction in uncertain networks) to predict future links in the whole uncertain network G. Algorithm *LPUN* performs *ComSim*(u) for every node u in the network. For each node pair (u,v), $\overline{W}(u, v)$can be calculated in *ComSim*(u) or *ComSim*(v). To avoid duplicate calculation of $\overline{W}(u, v)$, the algorithm performs *ComSim*(u) in the order of the clustering coefficient of the nodes. The clustering coefficient C_u of a node u is defined as:

$$C_u = \frac{2k_u}{d_u(d_u - 1)} \tag{8}$$

Here, k_u is the number of triangles connecting with u, and d_u is the degree of u. If a node u has a larger clustering coefficient, its sub-graph $G(u)$ will have richer topological information. The algorithm sorts the nodes in descending order of their clustering coefficients. The larger the clustering coefficients a node u has, the earlier that *ComSim*(u) is performed. Therefore, if the clustering coefficient of node u is larger than that of node v, $\overline{W}(u, v)$ is calculated by *ComSim*(u).

Algorithm *LPUN* (G)
Input: $G = (V, E, P, A)$: uncertain network;
P: the matrix of probabilities on edges;
A: the adjacent matrix;
Output: S: matrix of SimRank index
Begin
1. **For** every node u in G **do**
Compute the clustering coefficient C_u of u according to (8);
End for;
2. Sort the nodes in G in descending order of their clustering coefficients; let the order of the nodes be v_1,v_2,…,v_n after sorting;
3. **For** $j = 1$ **to** n **do**
$\overline{w}(v_j, v_j) = \prod_{(v_j, q) \in E} (1 - p(v_j, q))$;
If there is a node v in $G(v_j)$ such that $\overline{W}(v_j, v)$has not been computed **then**
Execute *ComSim*(v_j);
Endfor;
4. Calculate SimRank index $S = c\overline{W}^T S\overline{W} + (1 - c)I$
End

Let d be the maximum degree of all of the nodes in G and n be the number of nodes in G. To compute the clustering coefficients of n^2 nodes in G, step 1 in the algorithm requires $O(d.n^2)$ time. Step 2 sorts n nodes in O(n.log$n$) time. Because it requires $O(d^4)$

time to perform algorithm *ComSim*, the time complexity of step 3 is$O(d^4.n)$. SimRank computation in Step 4 requires $O(n^3)$ time. Because d can be treated as a constant, the time complexity of the algorithm *LPUN* is $O(n^3)$. Compared with the SimRank computation for a deterministic network, which also requires $O(n^3)$ time, our algorithm is efficient for an uncertain network.

5. Time series-random walk method

Because relations among social members are continuously changing over time, link probabilities in real-world social networks are also constantly varying and evolving. Therefore, link prediction algorithms should be capable of detecting dynamic relationships between members in the networks. The temporal uncertain network can be described by snapshots $G_t = (V, E_t, P_t, A_t)$ for $t = t_0, t_0 + 1, \ldots, t_0 + T - 1$, where T is the *window size*, P_t denotes the probability matrix of G_t, and A_t is the adjacent matrix. The time series link prediction problem in temporal uncertain networks is to predict the occurrence probabilities of edges at time $t_0 + T$.

Recently, some approaches have been advanced to detect potential or future links in temporal uncertain social networks. Most such methods treat the temporal networks as one-time events and ignore the time that the link occurs. In this work, we exploit temporal and topological information to predict potential links. In our proposed method *TS-RW* (time series random walk), we first compute probabilistic random walk transformation matrices $\overline{W}_{t_0}(u,v), \overline{W}_{t_0+1}(u,v), \ldots, \overline{W}_{t_0+T-1}(u,v)$ for the given temporal uncertain networks $G_{t_0}, G_{t_0+1}, \ldots, G_{t_0+T-1}$ with window size T. Then, we combine the sequence of $\overline{W}_{t_0}(u,v), \overline{W}_{t_0+1}(u,v), \ldots, \overline{W}_{t_0+T-1}(u,v)$ to one transformation matrix$\widetilde{W}$. In the evolution of the temporal network, recent snapshots are more reliable for future link prediction; they should be emphasized to obtain more accurate prediction results. In our method, a damping factor is used to assign greater importance to more recent information. Based on the damping factor, transformation matrix $\widetilde{W}$ is defined as:

$$\widetilde{W} = \sum_{t=t_0}^{t_0+T-1} \gamma^{t_0+T-1-t} \overline{W}_t \quad 0 < \gamma < 1 \tag{9}$$

In (9), γ is the damping factor. Our algorithm *TS-RW* first computes the transformation matrixes $\overline{W}_{t_0}(u,v), \overline{W}_{t_0+1}(u,v), \ldots, \overline{W}_{t_0+T-1}(u,v)$ using the algorithm *LPUN*. Next, it integrates these transformation matrixes into one transformation matrix $\widetilde{W}$. Then, it applies the SimRank index based on transformation matrix $\widetilde{W}$ to obtain the similarity score. The framework of algorithm *TS-RW* is as follows.

Algorithm *TS-RW* (link prediction in temporal uncertain networks)
Input:
$G_t = (V, E_t, P_t, A_t)$ $(t = t_0, t_0 + 1, \ldots, t_0 + T - 1)$: sequence of uncertain networks;
P_t: the edge probability matrix of G_t;
A_t: the adjacent matrix of G_t;
γ: damping factor $(0 < \gamma < 1)$;
c the decay factor for SimRank $(0 < c < 1)$;
Output:
S: time series random walk similarity matrix, where $S(i, j)$ is the occurrence probability score of edge (v_i, v_j) at time $t_0 + T$;
Begin
1. Initialize $\widetilde{W}$ as a zero $n \times n$ matrix;
2. **For** $t = t_0$ **to** t_0+T-1 **do**
Execute *LPUN*(G_t) to get $\overline{W}_t$;
$\widetilde{W} = \gamma\widetilde{W} + \overline{W}_t$;
Endfor
3. Calculate SimRank index $S = c\widetilde{W}^T S\widetilde{W} + (1 - c)I$
End

Let d be the maximum degree of all of the nodes in the sequence of graphs $G_{t_0}, G_{t_0+1}, \ldots, G_{t_0+T-1}$ and n be the maximum number of nodes in the network. Then, the time complexity for step 1 to initialize then $n \times n$ matrix $\widetilde{W}$is $O(n^2)$. Because $O(d^4.n)$ time is required for performing algorithm *LPUN*, the time complexity of step 2 is $O(T.d^4.n)$. SimRank computation in Step 3 requires $O(n^3)$ time. Because d can be treated as a constant, the time complexity of the algorithm *TS-RW* is $O(n^3)$. Compared with the SimRank computation for a static and deterministic network, which also requires $O(n^3)$ time, our algorithm is efficient for a temporal uncertain network.

6. Experimental results

To evaluate our proposed method for link prediction in a temporal uncertain social network, we test it by a series of experiments on several real temporal uncertain social networks. All of the experiments were performed on an Intel Core i3 computer running Windows 7 with 4GB memory. The algorithm was coded using the Python programming language. First, we introduce the six datasets used in the experiments and explain the experimental setup.

6.1. Datasets tested

6.1.1. High school dynamic contact networks http://www.sociopatterns.org

High school datasets are selected from a high school in Marseilles, France. They provide the dynamic uncertain networks of contacts between the students [12]. The first dataset is collected during four days in December 2011. Denoted by HS-2011, the dataset shows the contacts between students of three classes. The probability of each edge represents the time percentage of

Table 1
Main features of the datasets.

Datasets	#Nodes	# Links	P. length	T_N
HS-2011	126	8564	2 Hours	20
HS-2012	180	45047	3 Hours	28
Gulf -VISIT	174	5168	1 Year	20
Gulf-CRIT	174	2488	1 Year	20
Balkan-REC	325	3692	1 Year	15
Balkan-MAG	325	2687	1 Year	15

Table 2
Optimal γ values for the datasets.

Datasets	HS-2011	HS-2012	Gulf-VISIT	Gulf-CRIT	Balkan-REC	Balkan-MAG
γ	0.9	0.6	0.9	0.9	0.4	0.3

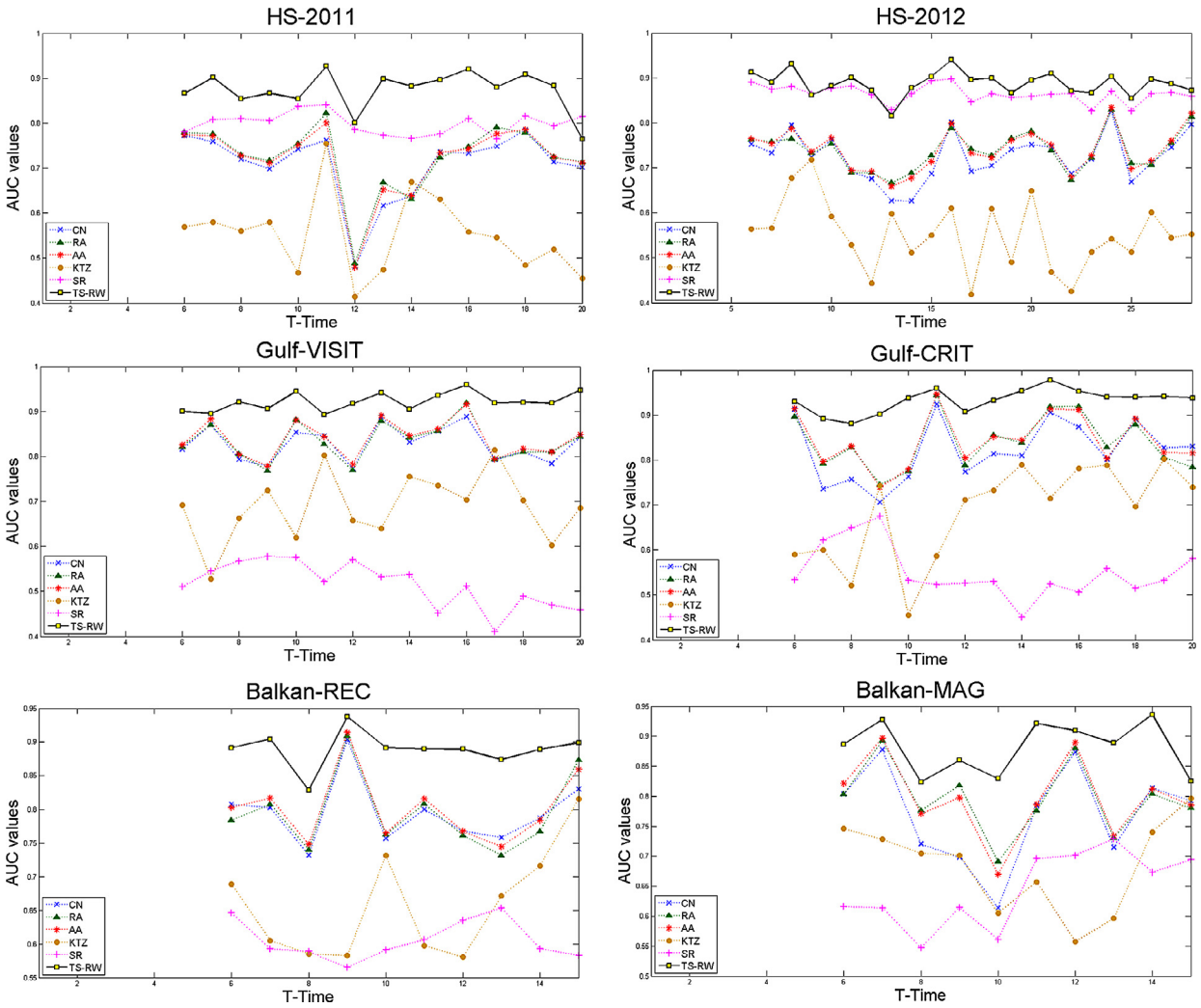

Fig. 1. Performance of *TS-RW* and the other methods.

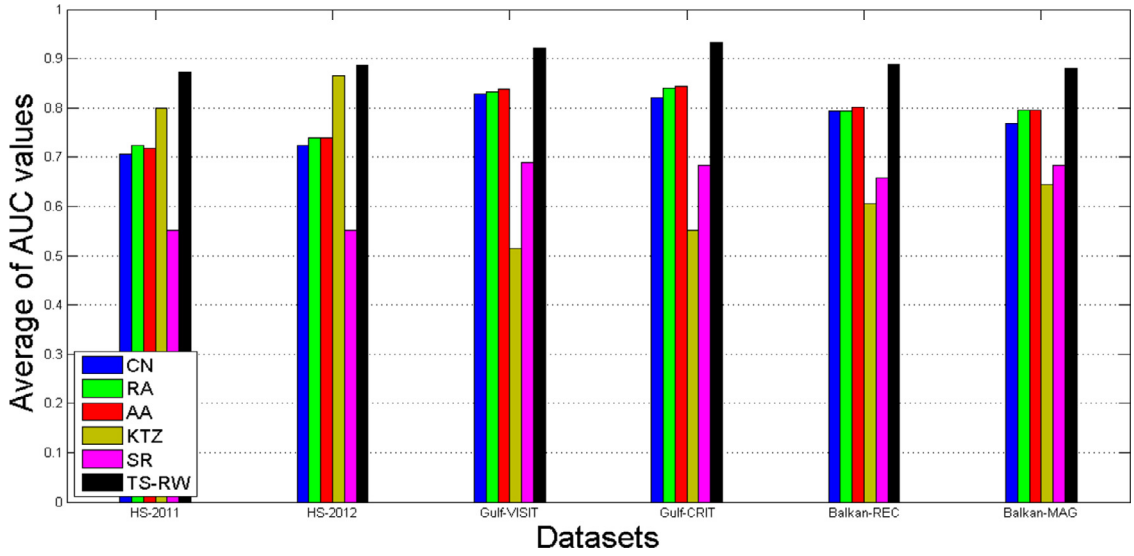

Fig. 2. Comparison of the average AUC values of *TS-RW* and other methods.

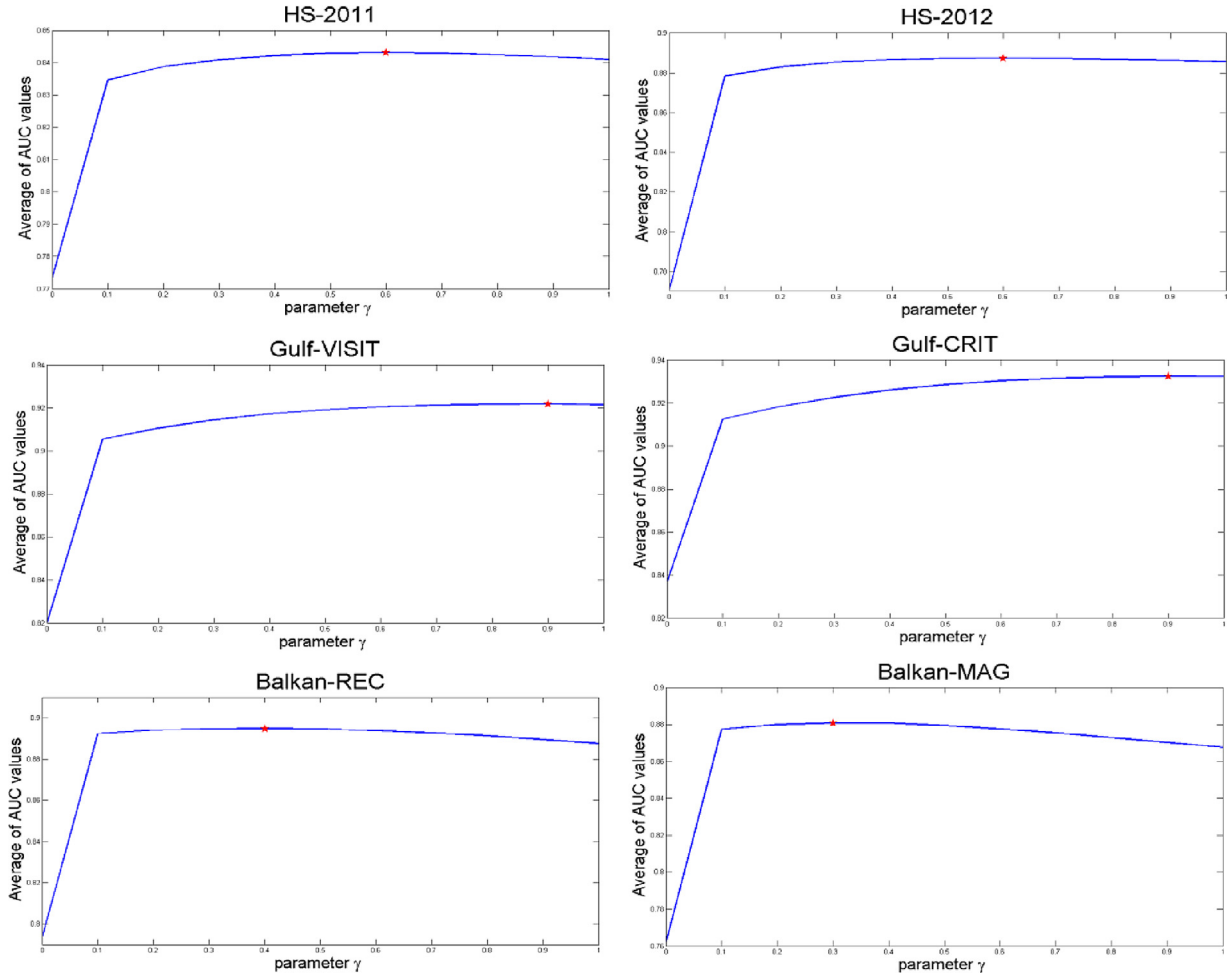

Fig. 3. Changes in the average AUC values for ITM with varying values of γ.

close-range face-to-face proximity during 2 h intervals between pairs of students. The second dataset, which is denoted by HS-2012, presents the contacts between 5 classes of students during a week in November 2012; the probability of an edge represents the time percentage of close-range face-to-face proximity during 3-hour intervals between a pair of students.

6.1.2. Gulf data set

Gulf data [9] is collected from Gulf region and Arabian Peninsula states for the period from April 1979 to March 1999. This time period is partitioned into 20 parts, we performed link prediction analysis on the yearly graphs.

In Gulf networks, nodes represent states inside the Gulf region and Arabian Peninsula. A Gulf network is a directed network; we studied two types of arcs in this network and use each type to form a dataset. The network that consists of the first type of arc is denoted by Gulf – VISIT. In this network, each arc is denoted by $e^1(u,v)$ and represents official visits between u and v. A probability in the arc $e^1(u,v)$ is the percentage of visits from node u to node v with respect to all visits in the graph. The network that consists of the second type of arc is denoted by Gulf–CRIT. In this network, each arc is denoted by $e^2(u,v)$ and represents the occurrence of criticism; a probability in the arc $e^2(u,v)$ is the percentage of criticism from node u to node v with respect to all criticism in the graph.

6.1.3. Balkan data set

This data set covers the states of the Balkan region in Southeast Europe for the period April 1989 to July 2003 [9]. This time period is partitioned into 15 parts; we performed link prediction analysis on the yearly graphs.

In the Balkan networks, nodes represent states inside the Balkan region. The Balkan network is a directed network; we studied two types of arcs in this network and use each type of arc to form a dataset. The network that consists of the first type of arc

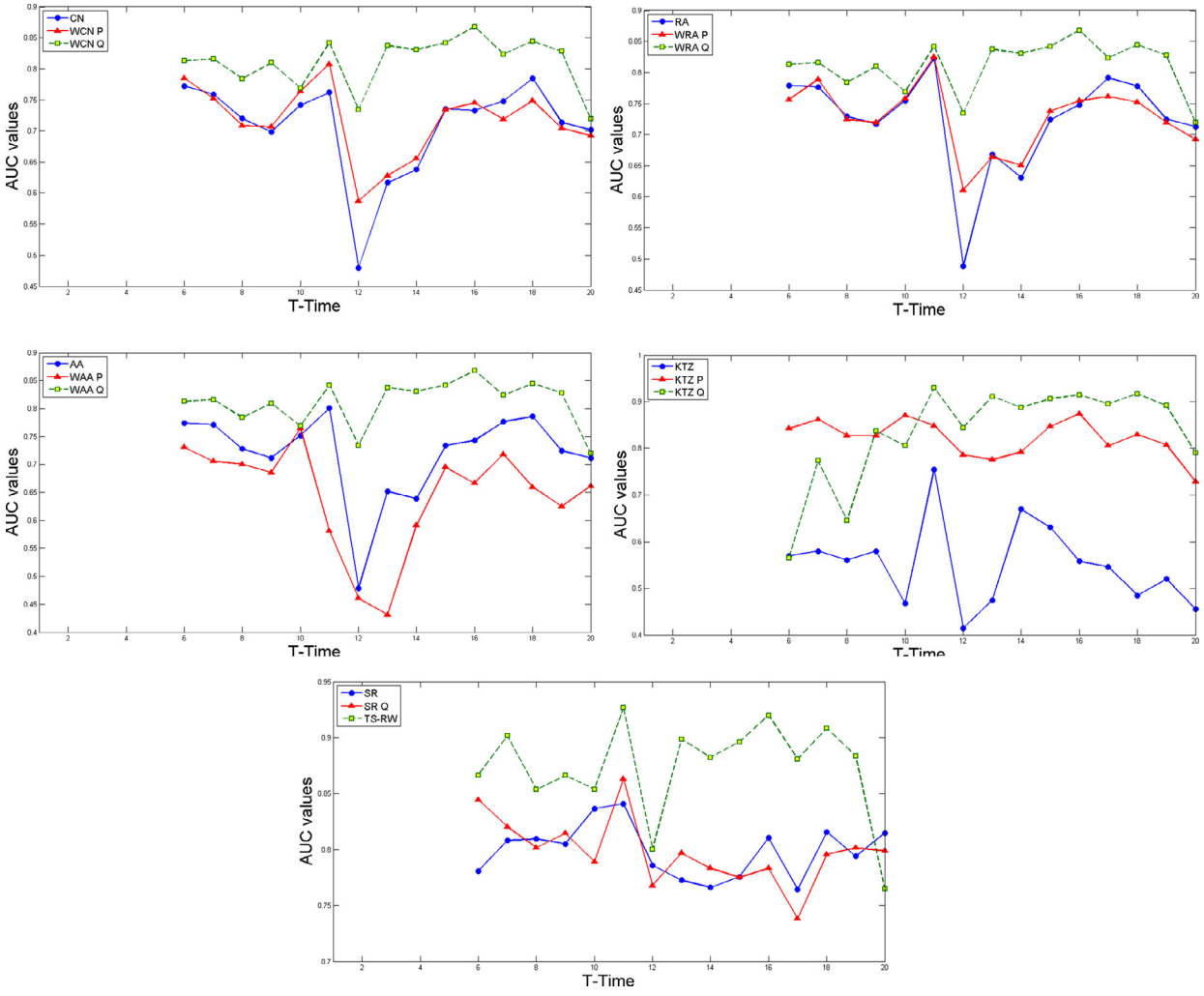

Fig. 4. Performance of different methods using HS-2011.

is denoted by Balkan-REC. In this network, each arc is denoted by $e^1(u,v)$ and represents sending and receiving messages. A probability of the arc $e^1(u,v)$ is the percentage of sent messages from node u to node v with respect to all messages in the graph. The network that consists of the second type of arc is denoted by Balkan-MAG. In this network, each arc is denoted by $e^2(u,v)$ and represents agreements made between u and v. A probability in the arc $e^2(u,v)$ is the percentage of agreements from node u to node v with respect to all agreements in the graph.

Table 1 shows the main features of the datasets, including the number of nodes (#Nodes), number of links (#Links), length of the time series sequence (T_N), and the length of the time period (P. Length).

6.2. Experiment setup

For every dataset, we took T_N snapshot 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, $G_{t_0}, G_{t_o+1} \ldots G_{t_0}, G_{t_o+T-1}$, to test the static and time series link prediction methods for predicting G_{t_o+T}. Because the topological structure of G_{t_o+T} was already known, our prediction result could be evaluated by comparing the links predicted with the actual presence of the links inG_{t_0+T}.

In the first part of our experiments, we tested the proposed algorithm *TS-RW* and compared the quality of the results with the methods based on a reduced static graph. The reduced static graph method has been commonly used in algorithms for link prediction in temporal networks. In this method, networks in the time series are first reduced to a static graph representation, and then, a static graph link prediction algorithm is applied to detect potential links in the reduced static graph. In other words, graph series $G_{t_0}, G_{t_0+1}, \ldots, G_{t_0+T-1}$ is reduced to a single binary graph $G_{t_0,T}$ with a corresponding adjacency matrix $A_{t_0,T}$, with

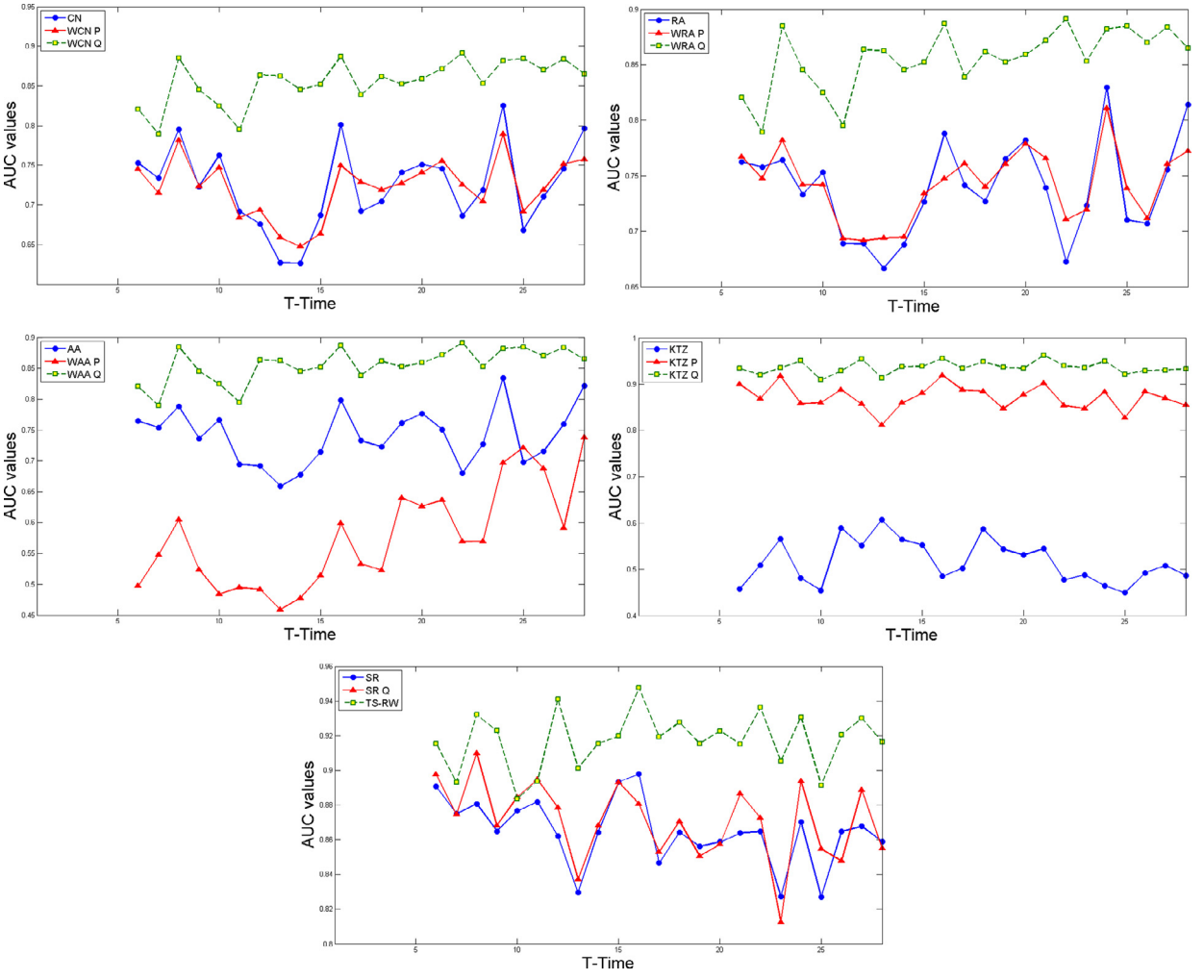

Fig. 5. Performance of different methods using the HS-2012 dataset.

the entries in $A_{t_0,T}$ given by

$$A_{t_0,T}(i,j) = \begin{cases} 1 & \text{if } \exists k \in [t_0, t_0+T-1] : A_k(i,j) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Then, static network link prediction methods are applied to the reduced static graph G_{t_o+T}, with the results for $G_{t_0,T}$ output as the solution of the temporal link prediction for G_{t_o+T}.

In our experiments for testing the reduced static graph method, we used the similarity measurements of *Common Neighbor*, *Resource Allocation*, *Adamic/Adar*, *Katz*, and *SimRank,* denoted by *CN*, *RA*, *AA*, *KTZ*, and *SR*, respectively.In the second part of our experiments, we use three types of representations for the probabilistic temporal networks in reduced static graph-based methods and compare the performances with our algorithm *TS-RW*. The first type of representation is the binary static graph given in Eq. (10). The second type is based on a static weighted graph where the probability on each edge is used as a weight. The third type is based on the dynamic weighted graph where the weights are reduced each time by a damping factor. For each of the similarity indices *CN, RA, AA, KTZ* and *SR*, we test their performances based on the three types of representations with that of our algorithm *TS-RW*. In all experiments, we refer to the methods *CN, RA, AA, KTZ* and *SR* under binary representation as *WCN P, WRA P, WAA P, KTZ P* and *SR P* under the static weighted graph and *WCN Q, WRA Q, WAA Q, KTZ Q* and *SR Q* under dynamic weighted representation.

After each algorithm calculates and ranks the similarities of all node pairs representing all existing and nonexistent links, we use the area-under-curve (AUC) score to evaluate the quality of the results for the algorithms tested. If we randomly choose pairs of an existing link and a nonexistent link, the AUC value is the expected ratio of the existing link with a higher similarity than the missing link. In an iterative way, we randomly choose two links, one from E_{t_0+T} and the other from E'_{t_0+T}, to compare their scores. If among n independent comparisons, n' times existing links have higher scores than nonexistent links and n'' times gain equal scores, the AUC value is given by

$$AUC = (n' + 0.5n'')/n \tag{11}$$

In general, a larger AUC value indicates better performance, and hence, the AUC value of the perfect result is 1.0, while the AUC of the result by a random predictor is 0.5.

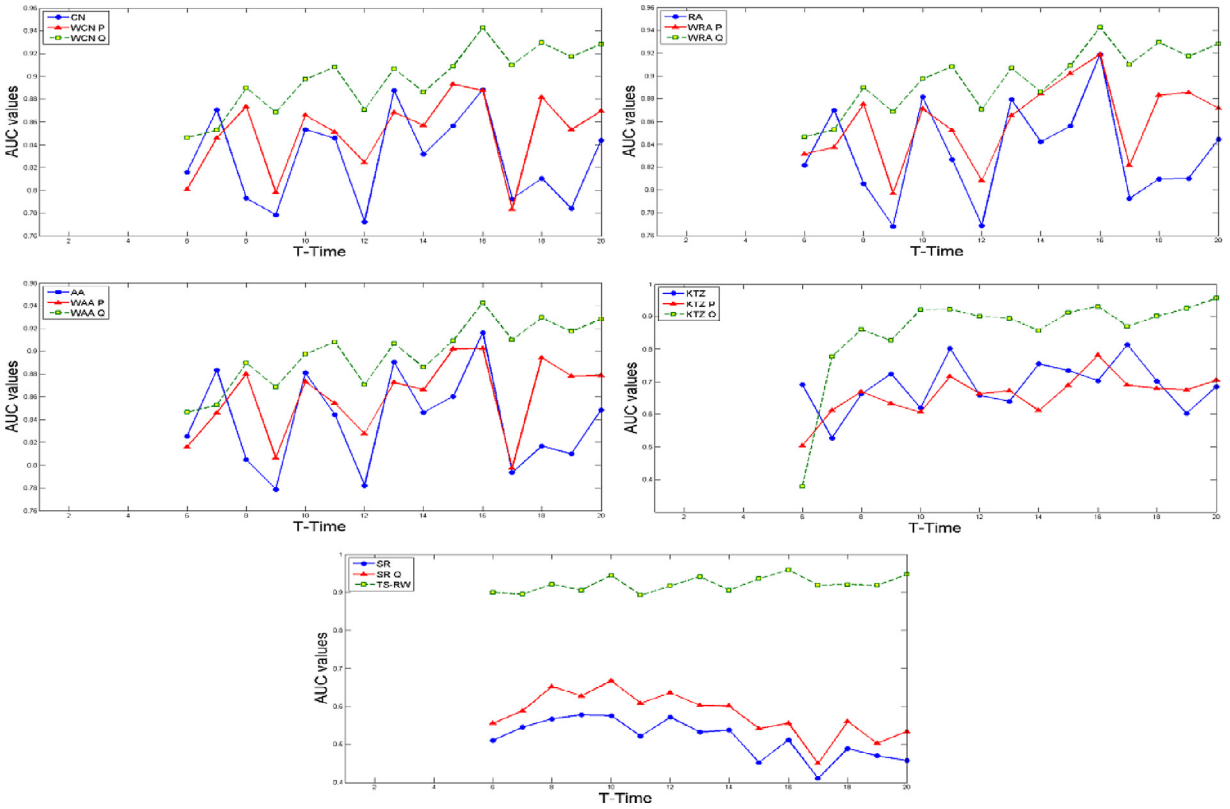

Fig. 6. Performance of different methods using the Gulf-VISIT dataset.

6.3. Experimental results and analysis

In our experiments, results were obtained by fixing window size $T = 5$ for all datasets. Because the changing rates of the edge probabilities in the datasets are different, the value of damping factor γ was also varied to adapt the changing rates of the edge probabilities in the network. Table 2 shows the values of γ we used, which gives the best results in different datasets.

In the first part of our experiments, we tested and compared the performance of proposed time series method *TS-RW* with that of methods *CN, RA, AA, KTZ* and *SR* on the six datasets. Fig. 1 shows the AUC values of the results by *TS-RW* (solid line) in every time t (t = 1,2,…,20) and the AUC values of the results by other methods (dashed lines). From the figure, we can see clearly that our *TS-RW* model achieves the highest performance among all of the methods including *KTZ* and *SR* on all datasets.

Fig. 2 illustrates the average AUC scores of the results by *TS-RW* and the methods *CN, RA, AA, KTZ* and *SR* on the six datasets. It can be seen from Fig. 2 that the general *TS-RW* achieves the best average performance on all datasets.

We also tested our method by varying the value of the damping factor γ, and the results are shown in Fig. 3. From the figure, we see that when $\gamma = 0$, the method achieved its lowest performance on most of the datasets. If $\gamma = 0$, only the most recent snapshot is used to predict its next graph; in other words, the window size T is set as 1. This shows the importance of historical information because the performance improves sharply when the historical information is introduced. In all datasets, after the performance has reached the maximum, it deteriorates when the value of γ continues to increase. This shows that the more recent data is more important. Because the window size in all our experiments is 5, which is quite small, all information included is somewhat new, which explains the slight deterioration. The average optimal values of γ for all datasets are given in Table 2; these are used for all our experiments.

In the second part of our experiments, we test the performances of the algorithms *CN, RA, AA, KTZ* and *SR* based on the three types of graph representations, namely, the static binary graph, static weight graph, and proposed dynamic weighted graph, and compare their performances with that of our algorithm *TS-RW*. Figs. 4, 5, 6, 7, 8 and 9 show the AUC values on the HS-2011, HS-2012, Gulf-VISIT, Gulf-CRIT, Balkan-REC, and Balkan-MAG datasets, respectively. Based on the figures, we observe that our algorithm *TS-RW* achieves the highest AUC values on all datasets compared with the other methods. The algorithm *TS-RW* achieves the highest quality results because the time series random walk strategy can integrate time and global topology information efficiently.

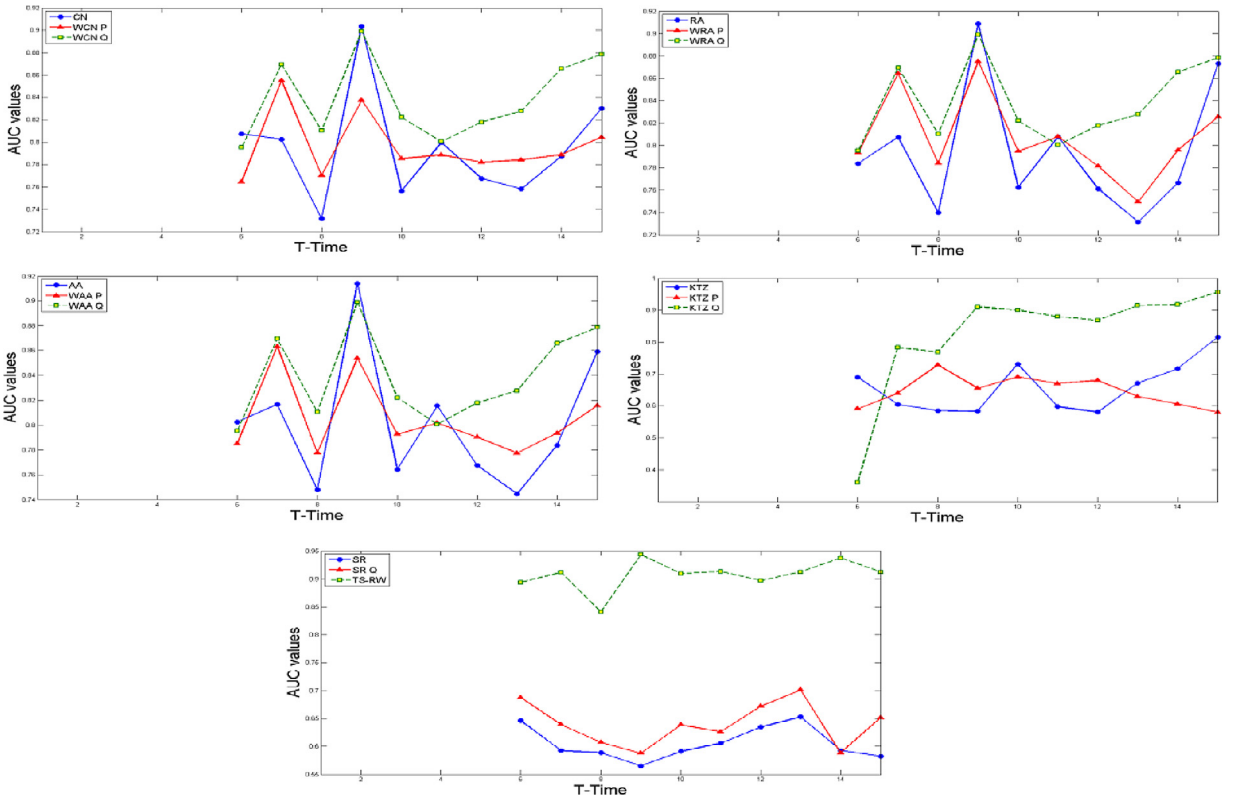

Fig. 7. Performance of different methods using the Gulf-CRIT dataset.

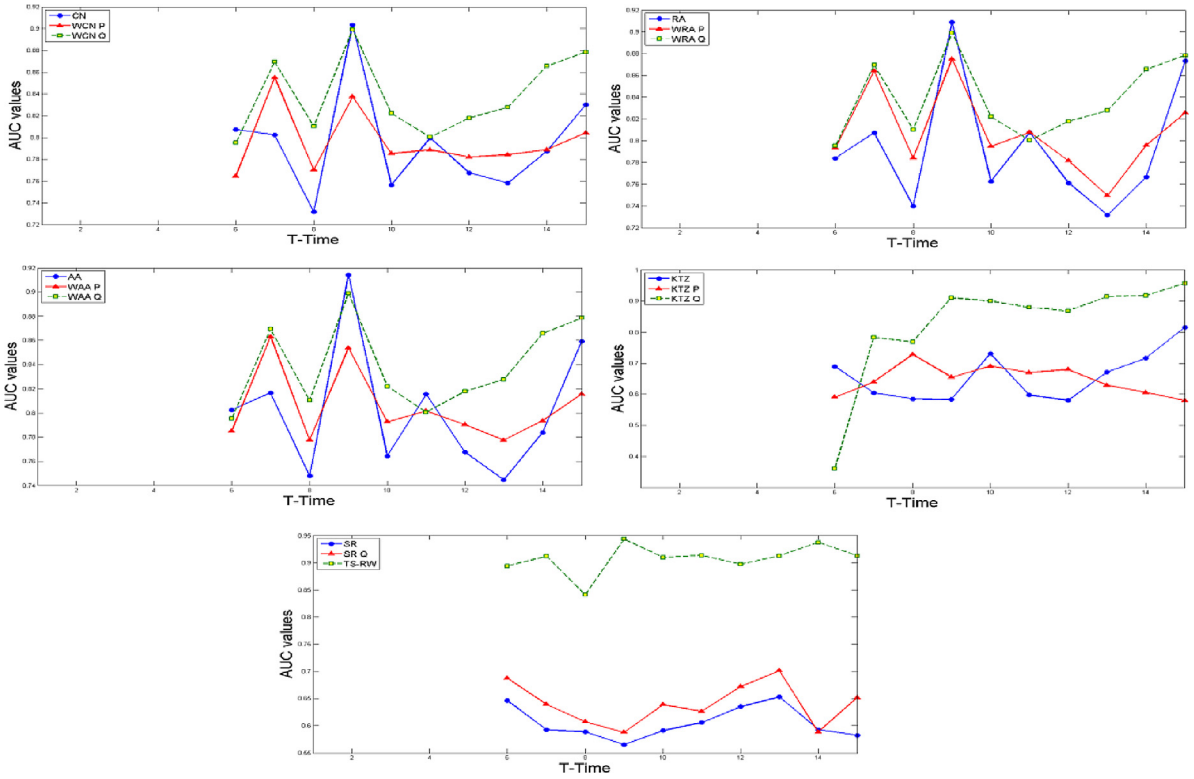

Fig. 8. Performance of different methods using the Balkan-REC dataset.

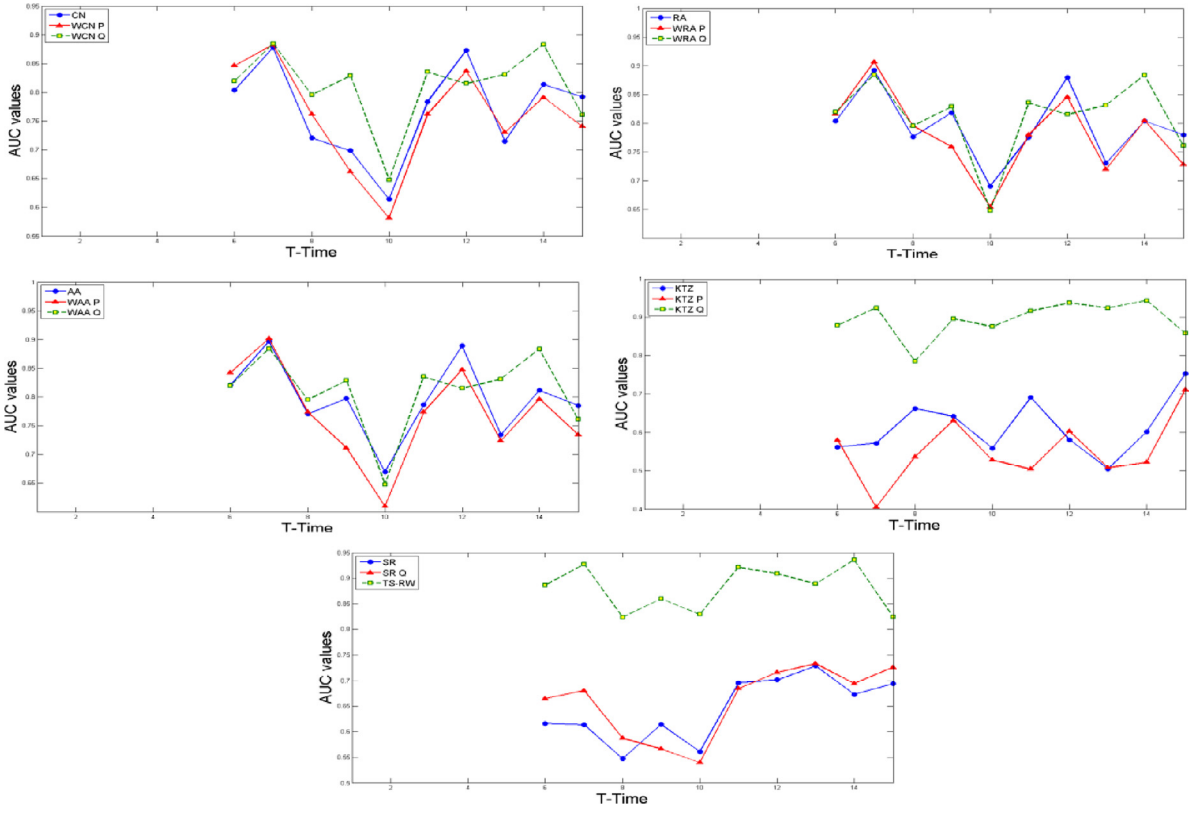

Fig. 9. Performance of different methods using the Balkan-MAG data.

With respect to methods other than *TS-RW*, the static weighted graph shows much higher results than those of the binary graph for the KTZ index using the HS-2011, and HS-2012 datasets; the dynamic weighted graph shows even higher quality results. In the Gulf-VISIT, Gulf-CRIT, Balkan-REC, and Balkan-MAG datasets, the weighted static graph shows better performance than does the binary graph with the SR index, while the dynamic weighted graph shows even better performance.

7. Conclusions and further work

We investigated the problem of link prediction in probabilistic temporal uncertain networks. In this work, we achieved higher quality link prediction results by designing a new method based on a random walk in temporal uncertain networks. Our method transforms the link prediction problem in uncertain networks to a random walk in a deterministic network. To reduce the computational time, the similarity scores between a node and its neighbors are computed within a sub-graph around this node. We also proposed a method to integrate temporal and global topological information to obtain more accurate results. Experimental results on real social networks show that our method can predict future links efficiently in temporal uncertain social networks and achieves higher quality results than other methods.

In real-world applications, the occurrence probability of each edge in the network may not be fixed and may change over time. To predict the potential future links in such networks, we must take the probabilities in the history of the edges into account. However, this requires a huge amount of computation time and memory space for large-scale uncertain dynamic networks. In future work, we intend to find an efficient approach for link prediction in uncertain dynamic networks where the occurrence probability of each edge in the network changes over time.

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 the Education Department of Jiangsu Province under contracts 12KJB520019, 13KJB520026, and 09KJB20013.

References

[1] E. Adar, C. Re, Managing uncertainty in social networks, IEEE Data Eng. Bull. 30 (2) (2007) 15–22.
[2] C.C. Aggarwal, Managing and Mining Uncertain Data editor, Advances in Database Systems, Springer, 2009 editor.
[3] 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 Sens. Intell. Syst. 6 (4) (2013) 1317–1342 September.
[4] L.M. Aiello, A. Barrat, R. Schifanella, et al., Friendship prediction and homophily in social media, ACM Trans. Web (TWEB) 6 (2) (2012) 9.
[5] S. Asthana, O.D. King, F.D. Gibbons, F.P. Roth, Predicting protein complex membership using uncertain network reliability, Genome Res. 14 (6) (2004) 1170–1175.
[6] M.O. Ball, Computational complexity of network reliability analysis: an overview, IEEE Trans. Reliab. 35 (1986) 230–239.
[7] Bao Zhifeng, Y Yong Zeng, C. Tay, sonLP: 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.
[8] Barbieri Nicola, Francesco Bonchi, Giuseppe 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.
[9] Vladimir Batagelj and Andrej Mrvar (2006): Pajek datasets. URL: http://vlado.fmf.uni-lj.si/pub/networks/data/.
[10] A Catherine, Morgan Bliss, R. Frank, Christopher M Danforth, Peter Sheridan Dodds, An evolutionary algorithm approach to link prediction in dynamic social networks, J. Comput. Sci. 5 (5) (2014) 750–764 September.
[11] B. Bringmann, M. Berlingerio, F. Bonchi, A. Gionis, Learning and predicting the evolution of social networks, IEEE Intell. Syst. (2010) 26–34.
[12] J. Fournet, A. Barrat, Contact patterns among high school students, PLoS One 9 (9) (2014) e107878.
[13] S. Gao, L. Denoyer, P. Gallinari, et al., Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks, J. China Univ. Posts Telecommun. 19 (2012) 172–181.
[14] S. Gao, L. Denoyer, P. Gallinari, Temporal Link Prediction by Integrating content and structure information, CIKM'11, Glasgow, Scotland, UK, 2011, pp. 1169–1174. October.
[15] J. Ghosh, H.Q. Ngo, S. Yoon, and C. Qiao. On a Routing Problem within Probabilistic Graphs and its Application to Intermittently Connected Networks. In INFOCOM'07, (2007), 1721–1729.
[16] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In WWW'04, (2004), 403–412.
[17] S. Hanneke, W.J. Fu, E.P. Xing, Discrete temporal models of social networks, Electron. J. Stat. 4 (2010) 585–605.
[18] He Yu-lin, JamesN.K. Liu, Yan-xing Hu, Xi-zhao Wang, OWA operator based link prediction ensemble for social network, Expert Syst. Appl. 42 (1) (2015) 21–50 January.
[19] P. Hintsanen, H. Toivonen, Finding reliable sub-graphs from large probabilistic graphs, Data Min. Knowl. Discov. 17 (1) (2008) 3–23.
[20] F.Y. Hu, H.S. Wong, Labelling of human motion based on CBGA and probabilistic model, Int. J. Smart Sens. Intell. Syst. 6 (2) (2013) 583–609 April.
[21] X. Jia, F. Xin, W.R. Chuan, Adaptive spray routing for opportunistic networks, Int. J. Smart Sens. Intell. Syst. 6 (1) (2013) 95–119 February.
[22] C. Kang, A. Pugliese, J. Grant, V.S. Subrahmanian, STUN: querying spatio-temporal uncertain (social) networks, Springer-Verlag Wien, 2014.
[23] M. Potamias, et al., K-nearest neighbors in uncertain networks, Proc. VLDB (2010) 997–1008.
[24] D. Kempe, J.M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In KDD 2003, (2003), 137–146.
[25] H. Kim, J. Tang, R. Anderson, C. Mascolo, Centrality prediction in dynamic human contact networks, Comput. Netw. 56 (2012) 983–996.
[26] Z.H. Liu, J.F. Ma, Y. Zeng, Secrecy transfer for sensor networks: from random graphs to secure random geometric graphs, Int. J. Smart Sens. Intell. Syst. 6 (1) (2013) 77–94 February.
[27] Ji Liu, G Denga, Link prediction in a user_object network based on time-weighted resource allocation, Physica A 388 (2009) 3643–3650.
[28] W. Liu, L. Lü, Link prediction based on local random walk, Europhys. Lett. 89 (5) (2010) 58007.
[29] 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.
[30] L.Y. Lü, T. Zhou, Link prediction in complex networks: A survey, Physica A 390 (2011) 1150–1170.
[31] T. Murata, S. Moriyasu, Link prediction based on structural properties of online social networks, N. Generat. Comput. 26 (3) (May 2008) 245–257.
[32] 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.

[33] M. Potamias, F. Bonchi, A. Gionis, G. Kollios, k-nearest neighbors in uncertain graphs, PVLDB 3 (1) (2010) 997–1008.
[34] M. Pujari, R. Kanawati, Supervised rank aggregation approach for link prediction in complex networks, WWW 2012 Companion, Lyon, France, 2012, pp. 1189–1196. April 16–20.
[35] S. Purnamrita, D. Chakrabarti, and M. Jordan. Nonparametric link prediction in dynamic networks, 2012, arXiv:1206-6394.
[36] G. Rubino. Network reliability evaluation. In Network performance modeling and simulation, pages 275–302. 1999.
[37] E. Sherkat, M. Rahgozar, M. Asadpour, Structural link prediction based on ant colony approach in social networks, Phys. A: Stat. Mech. Appl. 419 (1) (2015) 80–94.
[38] 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 80 (1) (2009) 017101-1–017101-4.
[39] L.G. Valiant, The complexity of enumeration and reliability problems, SIAM J. Comput. 8 (3) (1979) 410–421.
[40] 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.
[41] L.T. Yang, S. Wang, H. Jiang, Cyclic temporal network density and its impact on information diffusion for delay tolerant networks, Int. J. Smart Sens. Intell. Syst. 4 (1) (March 2011) 35–52.
[42] Y. Yuan, L. Chen, and G. Wang. Efficiently answering probability threshold-based shortest path queries over uncertain graphs. In DASFAA, pages 155–170, 2010.
[43] 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.
[44] T. Zhou, L. Lü, Y.C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (4) (2009) 623–630.
[45] Z. Zou, H. Gao, and J. Li. Discovering frequent sub-graphs over uncertain graph databases under probabilistic semantics. In KDD 2010, pages 633–642, 2010.
[46] Z. Zou, J. Li, H. Gao, and S. Zhang. Finding top-k maximal cliques in an uncertain graph. In ICDE2010, pages 649–652, 2010.
[47] Z. Zou, J. Li, H. Gao, S. Zhang, Mining frequent sub-graph patterns from uncertain graph data, IEEE Trans. Knowl. Data Eng. 22 (9) (2010).