New approaches for link prediction in temporal social networks

Nahla Mohamed Ahmed [1, 2], Ling Chen [1, 3 *]

[1] *College of Information Engineering, Yangzhou University, Yangzhou China, 225009*

[2] *College of Mathematical Sciences, Khartoum University, Khartoum, Sudan, 115547*

[3] *State Key Lab of Novel Software Tech, Nanjing University, Nanjing China, 210093*

Received 1 March 2014, www.tsi.lv

Abstract

Link prediction in social networks has attracted increasing attention from various domains such as sociology, anthropology, information science, and computer sciences. In this work, efficient approaches to predict potential link in temporal social networks are presented. One approach is based on reduced static graph using a modified reduced adjacency matrix to reflect the frequency of each link. Another approach is based on indices integration, and exploits both the temporal and topological information. The approach integrates the indices in all the time steps, which reflect the topological information, and uses a damping factor to emphasize the importance of more recent links. Experimental results on real datasets show that our approaches can efficiently predict future links in temporal social networks, and can achieve higher quality results than traditional methods.

Keywords: Social networks, temporal networks, link prediction

1 Introduction

Link prediction is an important task in social network analysis. It detects the hidden links from the observed part of the network, or predicts the future links given the current structure of the network. Link prediction has several applications in social network analysis [1-5].

Since relations among social members continuously change over time, links in real world social networks are varying and evolving constantly. Recently, approaches have been advanced to detect potential or future links in such temporal social networks.

Some of such methods are based on the analysis of the topological features of the network [6-8]. H. Kim et al. [8] presented a method to predict future network topology using the nodes' centrality, which can identify the important nodes in the future. Machine learning strategies are also exploited in temporal network link prediction methods [9-12]. Manisha Pujari et al. [9] applied a supervised rank aggregation method for link prediction in temporal complex networks. Some methods for link prediction on temporal network are based on probabilistic model [13-18]. S. Steve Hanneke et al. [13] proposed a family of statistical models for temporal social network link prediction by extending the exponential random graph model. However, such probabilistic model requires a predefined distribution of link appearance, which is difficult to know in advance for a given temporal network.

In this work, we present efficient approaches to predict potential link in temporal social networks. One approach is based on reduced static graph by using a modified reduced adjacency matrix to reflect the frequency of each link. Another approach integrates the similarity indices of the nodes to exploit both the temporal and topological information. Experimental results show that our methods can obtain higher quality results of link prediction in temporal social networks.

2 Reduced static graph approach

2.1 TRADITIONAL REDUCED STATIC GRAPH APPROACH

We use an undirected binary graph to represent the network. Let $V=[v_1,v_2,\ldots,v_N]$ be a set of vertices, $G_1,G_2,\ldots,G_T$ be a sequence of graphs on V at time steps $t=1,2,\ldots,T$. Define symmetric matrices $A_1,A_2,\ldots,A_T$ as the adjacency matrices of graphs $G_1,G_2,\ldots,G_T$ respectively. The binary value of $A_t(i,j)$ indicates the existence of an edge between nodes i and j, $i,j=1,2,\ldots,N$, during the time period t, $t=1,2,\ldots,T$. Given such graph series, the goal of link prediction in temporal social network is to predict the occurrence probabilities of edges at time T+1.

In the traditional algorithms for link prediction in temporal networks, the networks in time series

---

[*] Corresponding author - Tel: +15805271163; fax: +86-514-87887937; E-mail: yzulchen@163.com

$G_1, G_2, \ldots, G_T$ are reduced into a single weighted graph $G_{1,T}$ represented by a reduced adjacency matrix $\tilde{A}_{1,T}$:

$$\tilde{A}_{1,T}(i,j) = \begin{cases} 1 & if \quad \sum_{t=1}^{T} A_t(i,j) \geq 1 \\ 0 & otherwise \end{cases} \quad (1)$$

In the second step, a static network link prediction method is conducted on the reduced static graph $\tilde{G}_{1,T}$, and the result on $\tilde{G}_{1,T}$ is output as the solution of temporal link prediction on G_{T+1} .

2.2 IMPROVED REDUCED STATIC GRAPH APPROACH

The traditional reduced static graph methods for temporal network link prediction have drawbacks of missing some important information. First, since the reduced adjacency matrix $\tilde{A}_{1,T}$ defined in (1) is a binary one, it neglects the frequency of the links in the graphs $G_1, G_2, \ldots, G_T$. It is obvious that if a link occurs more frequently in the temporal network, it will have higher probability to appear in the future. Moreover, the reduced adjacency matrix $\tilde{A}_{1,T}$ ignores the time information, which is also important in link prediction for temporal networks. Since recent links in the network should have more importance in predicting the future links, they should have larger weights in calculating the similarity indexes.

A common feature of existing link prediction approaches under the static graph representation is that the occurrence probability of a link (i,j) is solely determined by other links related to it disregarding the temporal information. The Common Neighbour, Jaccard, Adamic/Adar indices rely on the number of occurrences of link pairs of the form $((i,k),(k,j))$, and ignore previous occurrences of link (i,j) itself. This is the main reason why some predictors based on Common Neighbour index fail to have a high performance in temporal networks with repeated links.

In our model proposed, we first add a self-loop (an edge form node i to j to each node in the graph representation of the social network. Thus if link (i,j) has appeared previously, the link occurrence probability of (i,j) exploits the occurrences of two link pairs $((i,i),(i,j))$, and $((i,j),(j,j))$. In addition, a damping factor is used to assign more importance to the more recent topological information. Therefore, the reduced adjacency matrix $A^*_{1,T}$ is defined as:

$$A^*_t(i,j) = \begin{cases} 1 & if \quad i = j \\ \sum_{t=1}^{T} \delta^{T-(t-1)} A_t(i,j) & otherwise \end{cases} \quad (2)$$

Here, $0 < \delta < 1$ is a damping factor. The base fact we considered in this model is that resent links have more accurate information than the old ones. In this model, the weight of each edge is reduced by the damping factor at each time. From (2), we also can see that value of $A^*_{1,T}(i,j)$ reflects the frequency of the appearance of link (i,j). Since more frequently appeared links have higher probability to emerge in the future, and existing links have higher probability to appear in the future than the missing links, the reduced static graph with adjacent matrix $A^*_{1,T}$ is more reliable for temporal link prediction. In our modified method, we take both the frequency and the time of the link appearance into consideration, the more frequently and recently a link (i,j) appears the higher weight is assigned to it according to (2).

As we will see later from the experimental results, our approach demonstrates higher performance and obtains more accurate results on future link prediction.

3 Indices integration algorithm for temporal link prediction

3.1 FRAMEWORK OF THE ALGORITHM

In this section, we present an indices integration method for temporal link prediction. In the method, we first calculate the indices of all the node pairs in the given temporal graphs $G_1, G_2, \ldots, G_T$. Such indices can be a commonly used similarity measurement such as Common Neighbour, Jaccard, Adamic/Adar, Preferential Attachment or Katz. Let the matrices of the indices be $S_1, S_2, \ldots, S_T$ for graphs $G_1, G_2, \ldots, G_T$, respectively. Then we integrate these matrices to construct $S_{1,T}$, which is a matrix consisting of the indices of probabilities for future links.

$$S_{1,T} = \sum_{t=1}^{T} \delta^{T-(t-1)} S_t \quad (3)$$

In (3), $0 < \delta < 1$ is a damping factor, which is used to assign more importance to the more recent topological information. The probability matrix $S_{1,T}$ carries both time and topological information.

To take the previously appeared links into consideration, in calculation of the indices for each graph G_t , we use the augmented adjacency matrix A^*_t instead of the traditional adjacency matrix A_t . The augmented adjacency matrix A^*_t is defined as:

$$A^*_t(i,j) = \begin{cases} 1 \\ number\ of\ connections\ between\ v_i \end{cases} \quad (4)$$

Figure 1 shows the framework of our indices integration algorithm for temporal link prediction.

Algorithm (Indices Integration Algorithm for Temporal Link Prediction)

Input:

$A_1^*, A_2^*, ..., A_T^*$: augmented adjacency matrix of $G_1, G_2, ..., G_T$;

$f : A^* \rightarrow S$, static graph link prediction algorithm where A^* is an augmented adjacency matrix, and S is a link occurrence probability score matrix;

δ : the damping factor $0 < \delta < 1$

Output:

$S_{1,T}$: $S_{1,T}(i, j)$ gives the probability score of edge (i, j) at time $T+1$;

begin

Initialize a zero $N*N$ matrix $S_{1,T}$;

Compute static graph link prediction matrices

For $t=1$ *to* T do

$S_t = f(A_t^*)$;

Estimate the time series link prediction model

For $t=1$ *to* T do

$S_{1,T} = S_{1,T} + \delta^{T-(t-1)} S_t$

Output $(S_{1,T})$

end

FIGURE 1 Indices Integration Algorithm for Temporal Link Prediction

3.2 COMPUTATION OF THE INDICES

Since the indices of Common Neighbour, Jaccard, Adamic Adar and Preferential Attachment only consider the direct neighbours for each node, they require computation time less than $O(N^2)$. However, since Katz index considers all existing paths in the graph, its computation time cost is quite expensive. In this section, we provide an efficient way for calculating the Katz index of graphs G_t, $t = 1,2,\dots,T$.

The Katz index is defined as follows:

$$S^{KZ} = (I - \beta A)^{-1} - I = \sum_{l=1}^{\infty} \beta^l A^l . \quad (5)$$

Here, A is the adjacency matrix, and $\beta \in (0,1)$ is a factor for emphasising the importance of the short links. S^{KZ} is the matrix consisting of the indices of the node pairs. From (5), we can see that the key challenge in the computation of Katz index is the matrix inversion, which requires $O(N^3)$ time. It is computationally prohibitive for large scale networks. To address this challenge, we approximate the elements in $Q = (I - \beta A)^{-1} - I$ by rewriting the equation of Q as

$$Q = (I - \beta A)^{-1} - I \approx \sum_{l=1}^{lm} \beta^l A^l . \quad (6)$$

Here, l_m is a positive integer, $A^l(i, j)$ is the number of paths of length *l* between nodes *i* and *j*. Since the long paths have less influence on the Katz index, we only consider the paths of lengths less than l_m.

In general, the temporal social network evolves smoothly over time, the corresponding sequence of adjacency matrices will not change rapidly. In the other words, the difference matrix $\Delta_t = A_{t+!} - A_t$ is likely to be a sparse one. Under this consideration, we are able to make a quick calculation of Q_{t+1} based on the second-order approximation:

$$Q_{t+1} = Q_t + \beta\Delta_t + \beta^2(\Delta_t A_t + A_t\Delta_t + \Delta_t^2) . \quad (7)$$

Equivalently,

$$S_{t+1}^{KZ} = S_t^{KZ} + \beta\Delta_t + \beta^2(\Delta_t A_t + A_t\Delta_t + \Delta_t^2) \quad (8)$$

Since Δ_t has few non-zero elements, terms $\Delta_t A_t$, $\Delta_t\Delta_t$ and Δ_t^2 in (8) are sparse matrix multiplications. By exploiting the sparseness of matrix Δ_t, we can compute the value of $Q_{t+!}$ very efficiently using incremental proximate updating [22]. As a result, we can compute S_{t+1}^{KZ} in $O(n_t)$ time, where n_t is the number of non-zero elements in Δ_t.

Experiments

To evaluate our proposed methods for link prediction in temporal social network, we test them by a series of experiments on several temporal social networks. All the experiments are performed on a Pentium IV computer running Windows XP, with 1.7G memory, and using VC++ 6. 0. First, we introduce the 6 datasets we used in the experiments, and introduce the experimental setup.

4.1 THE TEST DATASETS

4.1.1 Enron Email dataset

Enron Email dataset represents an email communication network of Enron employees during 38 months from May 1999 to June 2002. In the network, nodes represent employees. If there has been at least one email communication between two employees, an edge will link their nodes in the network. We performed the link prediction analysis on the monthly email graphs.

4.1.2 Infectious SocioPatterns (ISP) dataset

Infectious SocioPatterns dataset represents a network reflecting human mobility. It contains the daily cumulated networks represented in the Infectious SocioPatterns visualization system. The nodes represent visitors of the Science Gallery, while the edges represent close-range face-to-face proximity during 20 seconds interval between the concerned visitors.

After pre-processing, we choose the dataset of May first, which includes 8 active hours partitioned into 93 periods. Length of each of period is 5 minutes, therefore, we perform the link prediction analysis on periodically 5 minutes graphs.

4.1.3 Autonomous Systems (AS) dataset

Autonomous Systems dataset represents online communication network of who-talks-to-whom. It was collected by the University of Oregon Route Views Project - Online Data and Reports. The graph consists of routers in the Internet, and is organized into sub-graphs called Autonomous Systems (AS). The nodes represent ASs, and edges represent AS exchanging traffic flows between corresponding nodes. The dataset contains 733 daily instances which span an interval of 785 days from November 8, 1997 to January 2, 2000.

After pre-processing, we chose the first 30 days in the given data, and 470 most active Autonomous Systems. The link prediction algorithm is performed on daily AS communication graphs.

4.1.4 Nodobo dataset

Nodobo dataset represents mobile phone calls network of high-school students, from September 2010 to February 2011. The nodes represent students, and links represents phone calls between corresponding students.

After pre-processing, we choose 61 days starting from the first of October to the 30th of November. The link prediction algorithm is performed on daily phone call graphs.

4.1.5 Manufacturing Emails (ME) dataset

Manufacturing Emails dataset is the internal email communication network between employees of a mid-sized manufacturing company. The nodes represent employees, and an edge between two nodes represent their email communication.

4.1.6 Southern Women (SW) dataset (http://konect.uni-koblenz.de/networks/opsahl-southernwomen)

Southern Women dataset shows the participations of 18 women in 14 social events over a nine-month period. The data was collected in the southern United States of America in the 1930s. Originally, this dataset represents a bipartite network where each edge connects a woman with the event she participates in. We transform this dataset to unipartite temporal networks where the nodes represent women, edges between two nodes in the *t*-th graph shows that these two women participate in the same event at time *t*.

Table 1 shows the main features of the 6 datasets, including number of nodes (#Nodes), total number of links (#Links), total number of unique links(# Unique Links), by which repeated links between the same node pair are counted only once, length of the time series sequence (T), total number of unique links at each time step (#Unique Links-T).

TABLE 1 The main features of the datasets

Datasets	#Nodes	#Links	#Unique Links	T	#Unique Links-T
Enron	145	26092	1101	38	3960
ISP	200	5943	714	93	1733
AS	470	92348	1842	30	46174
Nodobo	395	5453	453	61	2016
ME	167	82926	3250	117	29847
SW	14	214	66	18	214

4.2 EXPERIMENT SETUP

For every dataset, we take $T+1$ snapshot graphs $G_1, G_2, \ldots, G_{T+1}$. In each time step t, $t = 1,2,\ldots,T$, we use the first t graphs, $G_1, G_2, \ldots, G_T$, to test static and time series link prediction methods to predict links in G_{t+1}. In static graph representation tests, we combine these t graphs to build $G_{1,t}$ as shown in Section 3.

We test our methods based on reduced static graph and on indices integration, and compare the quality of their results with the traditional methods based on reduced static graph. In those traditional methods, we use the similarity measurements of Common Neighbour, Jacard, Adamic/Adar, Katz, and Preferential Attachment; we denote them as CN, JC, AA, KZ and PA respectively. We also test the methods based on the modified reduced adjacency matrix $A^*_{1,T}$ defined in (2) using similarity measurements of Common Neighbour, Jacard, and Adamic/Adar, we denote them as CN*, JC*, AA*, respectively. For the methods based on indices integration, we use both adjacent matrix A_t and augmented adjacency matrix A^*_t. When using adjacent matrix A_t, we use the similarity measurements of Katz and Preferential Attachment, the algorithms are denoted as TKZ, TPA respectively. When using augmented adjacency matrix A^*_t, we use the similarity measurements of Common Neighbour, Jacard, and Adamic/Adar, the algorithms are denoted as TCN*, TJC* and TAA* respectively. Table 2 lists the names of the algorithms, along with their reduction methods and adjacent matrixes used.

TABLE 2 Link prediction methods tested

Reduction method	Adjacent matrix	Common neighbour	Preferential Attachment	Jaccard	Adamic/Adar	Katz
Reduced static graph	$\tilde{A}_{1,T}$ defined in (1)	CN	PA	JC	AA	KZ
	$A^*_{1,T}$ defined in (2)	CN^*	-	JC^*	AA^*	-
Indices Integration	A_t	-	TPA	-	-	TKZ
	A^*_t defined in (4)	TCN^*	-	TJC^*	TAA^*	-

We use AUC (Area Under Curve) scores to evaluate the quality of the results by the algorithms tested. After the algorithms calculating and ranking the similarities of the all the node pairs, which represent all the existent and the non-existent links, the AUC value can be interpreted as the probability that a randomly chosen existent link is given a higher score than a randomly chosen non-existent link. At each time we randomly pick an existent link and a non-existent link to compare their scores, if among n independent comparisons, there are n' times the existent link having a higher score and n'' times they have the same score, the AUC value is

$$AUC = (n' + 0.5n'')/n \tag{9}$$

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

4.3 EXPERIMENTAL RESULTS AND ANALYSIS

4.3.1 Tests of the methods based on reduced static graph

First we compare the performance of the reduced static graph methods CN, JC and AA using traditional reduced adjacency matrix $\tilde{A}_{1,T}$ defined in (1) with the methods CN*, JC* and AA* using the modified reduced adjacency matrix $A^*_{1,T}$ defined in (2). Figure 2 shows the AUC values of the results by CN, JC and AA using solid lines, and the AUC values of the results by CN*, JC* and AA* using dash lines. The figure shows a high improvement of AUCs by methods CN*, JC* and AA* implementing modified reduced adjacency matrix A*1T. The reason for CN*, JC* and AA* getting higher quality results is that the modified reduced adjacency matrix $A^*_{1,T}$ integrates the information of both the frequency and the time of the link appearance. Since the network link connections accumulate over time, the probability of repeated links increases gradually. This explains why the solid lines and dashed lines in Figures 2(A) and 2(B) are close to each other when t is small and run far away when t becomes larger. Figure 2 clearly shows that methods CN*, JC* and AA* can achieve much better performance than other methods.

Table 3 illustrates the average AUC scores of the results by the methods shown in Figure 2. It can be seen from Table 3 that method JC* achieves the best performance on Enron, ISP and Manufacturing Emails datasets, while AA* and CN* obtain the best AUC values on AS and Nodobo datasets respectively. We also can see from the table that methods CN*, JC* and AA*, which use the modified reduced adjacency matrix $A^*_{1,T}$, have higher AUC values than all the other methods on the datasets.

Figure 3 compares the average AUC values by the methods CN*, JC* and AA* with those by PA and KZ. From the figure, we can see that methods CN*, JC* and AA* have better performances than PA, and even KZ. Although method KZ has a deep consideration on common paths, methods CN*, JC*, AA* get even better results.

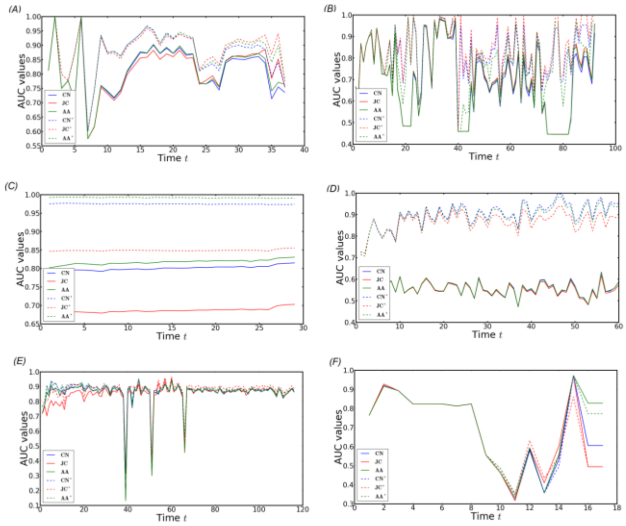

FIGURE 2 Testing results by methods CN, JC, AA, CN*, JC* and AA* on datasets:(A) Enron, (B) ISP, (C) AS, (D) Nodobo, (E) ME, (F) SW

Table 4 presents the average AUC values of CN*, JC*, AA*, PA and KZ on the 6 datasets. It can be seen from Table 4 that method JC* achieves the best performance on Enron and ISP datasets, while AA* and CN* obtain the best AUC values on AS and Nodobo datasets respectively. It is obvious that methods CN*, JC* and AA*, which use modified reduced adjacency matrix $A^*_{1,T}$, have higher AUC values than the other methods on most of the datasets. This demonstrates that using the modified reduced adjacency matrix is helpful for increasing the quality of link prediction in temporal social networks.

TABLE 3 Average AUC values by methods CN, JC, AA, CN*, JC* and AA* on 6 datasets

Algorithm	Enron	ISP	AS	Nodobo	ME	SW
CN	0.8095	0.6966	0.8006	0.5546	0.857	0.6867
JC	0.8133	0.7064	0.6861	0.5505	0.8441	0.682
AA	0.8148	0.7029	0.8174	0.5543	0.8587	0.7153
CN*	0.8735	0.8242	0.9748	0.9004	0.8725	0.6884
JC*	0.8827	0.8423	0.8492	0.868	0.8772	0.6776
AA*	0.8797	0.7969	0.9918	0.8917	0.8748	0.7116

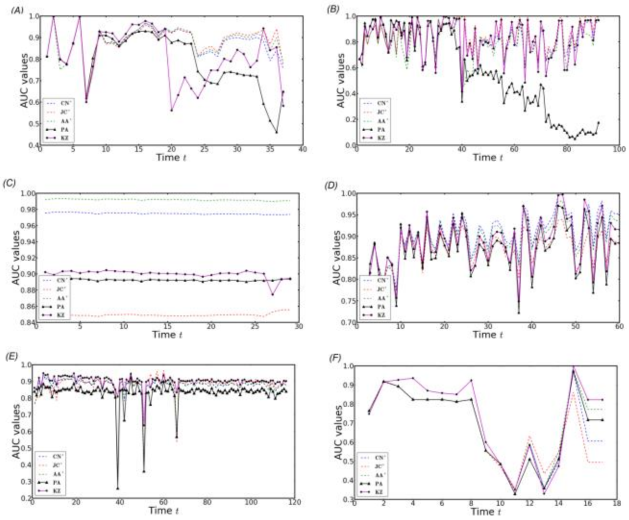

FIGURE 3 Testing results by methods CN*, JC*, AA*, and PA, KZ on datasets: (A) Enron, (B) ISP, (C) AS, (D) Nodobo, (E) ME, (F) SW

TABLE 4 The average of AUC values of CN*, JC*, AA*, PA, and KZ

Algorithm	Enron	ISP	AS	Nodobo	ME	SW
CN*	0.8735	0.8242	0.9748	0.9004	0.8725	0.6884
JC*	0.8827	0.8423	0.8492	0.868	0.8772	0.6776
AA*	0.8797	0.7969	0.9918	0.8917	0.8748	0.7116
PA	0.7993	0.5521	0.8923	0.8673	0.8301	0.6964
KZ	0.8194	0.8305	0.8998	0.8822	0.9019	0.7349

4.3.2 Tests of the methods based on indices integration

We also test our indices integration methods TCN*, TJC*, TAA*, TPA and TKZ on the 6 data sets. Figures 4 to 9 show the AUC values of the results on the datasets of Enron, ISP, AS, Nodobo, ME and SW respectively. We also compare the AUC values of the results by methods TCN*, TJC*, TAA*, TPA, and TKZ with their corresponding reduced static graph methods CN, JC, AA, PA and KZ. From the figures, we can see that AUC values of the results by methods TCN*, TJC*, TAA*, TPA and TKZ are all higher than their corresponding static graph methods on all the data sets. For instance, a surprising result we achieved is the amazing improvement of AUC value by TPA over PA on Infectious SocioPatterns dataset. When t is greater than 40, AUC value TPA increases rapidly, and becomes 3 to 10 times higher than that of PA. Our methods based on indices integration can achieve higher quality results because they consider both temporal and topological information, while static graphs methods lose time serial information. In our methods, temporal representation provides better improvement on the results of link prediction.

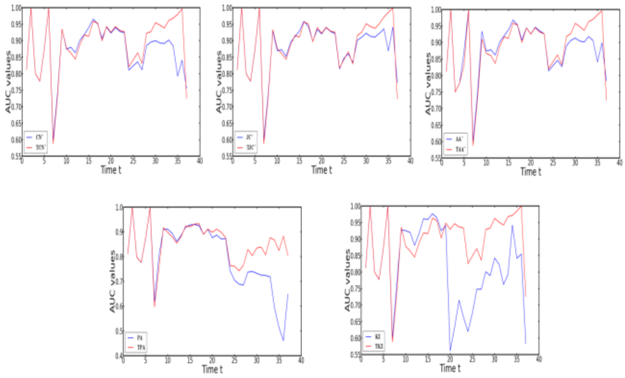

FIGURE 4 Testing results by methods based on indices integration on Enron dataset

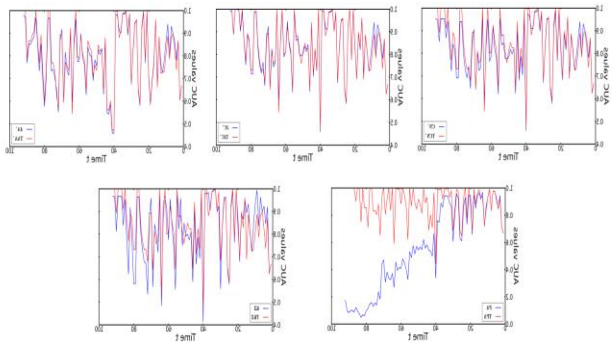

FIGURE 5 Testing results by methods based on indices integration on Infectious SocioPatterns dataset

Table 5 shows the experimental results by our indices integration based methods TCN*, TJC*, TAA*, TPA, TKZ and their counterparts based on static graph. The reported values in the table are the average AUC scores of tests at 60 time steps on the 6 datasets. It shows that indices integration based Katz method, TKZ, achieves AUC values 0.9845, 0.9014, 0.9359 and 0.7965 on Enron, Nodobo, and Manufacturing Emails and South Women datasets respectively. These are the best AUC values among all the methods on those datasets. The indices integration based Preferential Attachment method, TPA, has the highest AUC score 0.8909 for Infectious SocioPatterns dataset, while the indices integration based Jacard method, TJC*, obtains the highest AUC score 0.9932 on Autonomous Systems dataset.

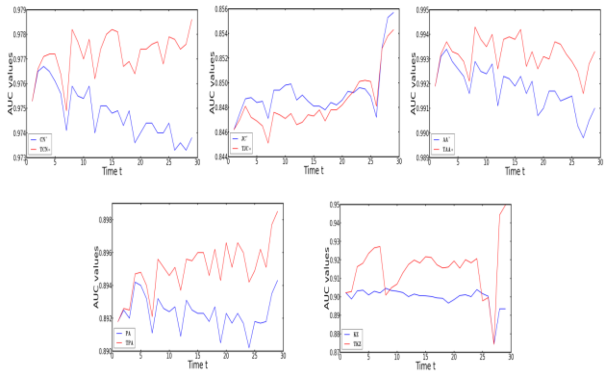

FIGURE 6 Testing results by methods based on indices integration on Autonomous systems dataset

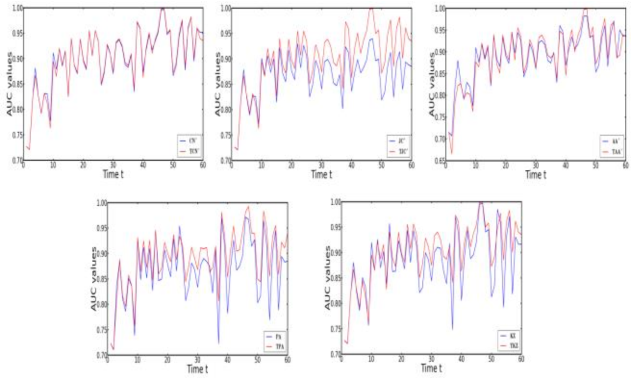

FIGURE 7 Testing results by methods based on indices integration on Nodobo dataset

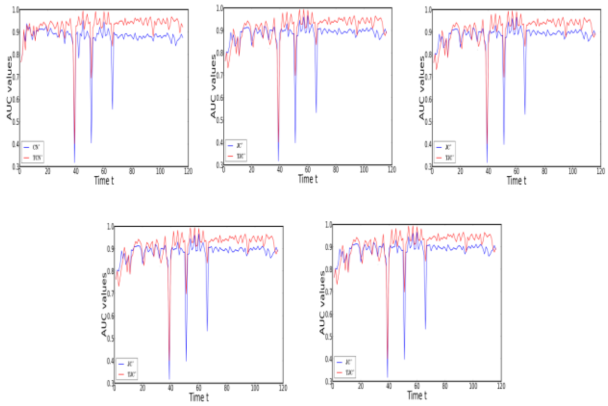

FIGURE 8 Testing results by methods based on indices integration on Man.Email dataset

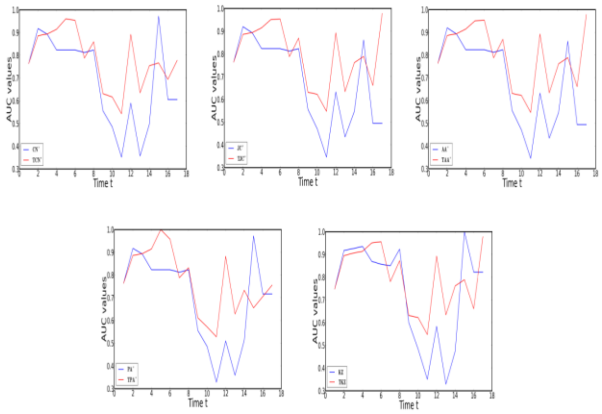

FIGURE 9 Testing results by methods based on indices integration on SW dataset

It is worth mentioning that the best AUC values we achieved in all experiments on Enron, ISP, AS, Nodobo and Manufacturing Email datasets are 0.8945, 0.8909, 0.9932, 0.9014 and 0.9539 respectively. These significantly high AUC values are all obtained by our indices integration based methods. By using the augmented adjacency matrix, we can improve the quality of the results of link prediction in temporal social networks. Moreover, we reach even better improvement by the methods using indices integration, which can integrate the temporal and the topological information of the temporal social networks.

TABLE 5 Average AUC values for static and time series graph representations

Algorithm	Enron	ISP	AS	Nodobo	ME	SW
CN*	0.8735	0.8242	0.9748	0.9004	0.8725	0.6884
JC*	0.8827	0.8423	0.8492	0.868	0.8772	0.6776
AA*	0.8797	0.7969	0.9918	0.8917	0.8748	0.7116
PA	0.7993	0.5521	0.8923	0.8673	0.8301	0.6964
KZ	0.8194	0.8305	0.8998	0.8822	0.9019	0.7349
TCN*	0.8914	0.8422	0.9772	0.9008	0.9253	0.7843
TJC*	0.8902	0.8424	0.8483	0.9005	0.9092	0.7964
TAA*	0.8877	0.8034	0.9932	0.8924	0.9251	0.7775
TPA	0.8524	0.8909	0.895	0.8938	0.8749	0.7712
TKZ	0.8945	0.8423	0.9155	0.9014	0.9359	0.7965

5 Conclusions

We investigate the problem of link prediction in temporal social networks. In this work, we achieve higher quality link prediction results by providing larger weights for frequently occurred links. We also present a time series model that exploit temporal information on evolving social network for link prediction. We advanced a method based on indices integration6 which exploits both the temporal and topological information of the temporal networks. To take the previously appeared links into consideration, we define an augmented adjacency matrix in calculating the indices at each time step. We provide a fast algorithm for efficiently calculating the indices involving matrix computation such as Katz index. We conduct extensive experimental evaluation of our methods on benchmark datasets. Experimental results show that our methods can obtain higher quality results of link prediction in temporal social networks.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (61379066, 61070047, 61070133), Natural Science Foundation of Jiangsu Province (BK21010134, BK20130452) Natural Science Foundation of Education Department of Jiangsu Province (09KJB20013).

References

[1] Lü L Y, Zhou T 2011 *Physica A* **390** 1150–1170

[2] Ahmed H S, Faouzi B M, Caelen J 2013 Detection and classification of the behaviour of people in an intelligent building by camera *International Journal on Smart Sensing and Intelligent Systems* **6**(4) 1317-1342

[3] Yang L T, Wang S, Jiang H 2011 Cyclic Temporal Network Density and its impact on Information Diffusion for Delay Tolerant Networks *International Journal on Smart Sensing and Intelligent Systems* **4**(1) 35-52

[4] Liu Z H, Ma J F, Zeng Y 2013 Secrecy transfer for sensor networks: from random graphs to secure random geometric graphs *International Journal on Smart Sensing and Intelligent Systems* **6**(1) 77-94

[5] Jia X, Xin F, Chuan W R 2013 Adaptive spray routing for opportunistic networks *International Journal on Smart Sensing and Intelligent Systems* **6**(1) 95-119

[6] Purnamrita S, Chakrabarti D, Jordan M 2012 Nonparametric link prediction in dynamic networks *arXiv preprint arXiv* 1206-6394

[7] Murata T, Moriyasu S 2008 Link Prediction based on Structural Properties of Online Social Networks *New Generation Computing* **26**(3) 245-257

[8] Kim H, Tang J, Anderson R, Mascolo C 2012 Centrality prediction in dynamic human contact networks *Computer Networks* **56** 983-996

[9] Pujari M, Kanawati R 2012 *Supervised Rank Aggregation Approach for Link Prediction in Complex Networks* Companion, Lyon, France pp. 1189-1196 April 16–20, 2012

[10] Vu D Q, Asuncion A U, Hunter D R, Smyth P 2011 Continuous-time regression models for longitudinal networks *Advances in Neural Information Processing Systems 24: Proceedings of the 25th Annual Conference on Neural Information Processing Systems* 1-9

[11] Bringmann B, Berlingerio M, Bonchi F, Gionis A 2010 Learning and Predicting the Evolution of Social Networks *IEEE Intelligent Systems* 26-34

[12] O'Madadhain J, Hutchins J, Smyth P 2005 Prediction and ranking algorithms for event-based network data *In Proceeding of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 23-30

[13] Hanneke S, Fu W J, Xing E P 2010 Discrete temporal models of social Networks *Electronic Journal of Statistics* **4** 585–605

[14] Ji Liu, Denga G 2009 Link prediction in a user_object network based on time-weighted resource allocation *Physica A* **388** 3643-3650

[15] Gao S, Denoyer L, Gallinari P 2011 Temporal Link Prediction by Integrating Content and Structure Information *CIKM'11, Glasgow, Scotland, UK* 1169-1174

[16] Potgieter A, April K, Cooke R, Osunmakinde I O 2006 Temporality in link prediction: Understanding social complexity *Journal of Transactions on Engineering Management* **11**(1) 83-96

[17] Hu F Y, Wong H S 2013 Labelling of Human Motion Based on CBGA and Probabilistic Model *International Journal on Smart Sensing and Intelligent Systems* **6**(2) 583-609

[18] Jianhan Zhu, Jun Hong, Hughes J G 2002 Using Markov Chains for Link prediction in Adaptive Web Sites D. Bustard, W. Liu, and R. Sterritt (Eds.): *Soft-Ware 2002, LNCS* **2311** 60–73

Authors

Nahla Mohamed Ahmed, born in July, 14, 1987, Khartoum, Sudan

Current position, grades: PhD student, MS
University studies: Khartoum University, Khartoum, Sudan
Scientific interest: Mathematics
Experience: She got her master degree in mathematics in Institute of Mathematical Sciences, Cape town, South Africa. She is currently a lecturer in College of Mathematical Sciences, Khartoum University, Sudan. Her research interest is in complex network analysis.

Ling Chen, born in September, 10, 1951, Jiangsu, China

Current position, grades: Professor
University studies: Yangzhou University, China
Scientific interest: Computer Science
Publications: Has published more than 200 journal and conference papers in computer science.
Experience: He is currently professor of computer science, and the dean of Information Technology College, Yangzhou University, Jiangsu Province, P.R. China. He is also the chair of Yangzhou Computer Society, vice chair of Jiangsu Computer Society, senior member of the Chinese Computer Society, member of IEEE Computer Society. His research interests include parallel processing, computer architecture, algorithms of optimization.