Information Sciences 498 (2019) 41–61

Contents lists available at ScienceDirect

Information Sciences

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

Link prediction in temporal networks: Integrating survival analysis and game theory

Zhan Bu[a,*], Yuyao Wang[b], Hui-Jia Li[c], Jiuchuan Jiang[a], Zhiang Wu[a], Jie Cao[b,*]

[a] Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, China
[b] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
[c] School of Science, Beijing University of Posts and Telecommunications, Beijing, China

ARTICLE INFO

Article history:
Received 3 April 2018
Revised 6 May 2019
Accepted 19 May 2019
Available online 20 May 2019

Keywords:
Link prediction
Temporal network
Survival analysis
Game theory
Autonomy oriented computing

ABSTRACT

Link prediction is an important task in complex network analysis and can be found in many real-world applications such as recommendation systems, information retrieval, and marketing analysis of social networks. This paper focuses on studying the evolution mechanism of real-world temporal networks. Specifically, given a set of temporal links during a fixed time window, how to predict the existence of links at any point in the future. To address this problem, we propose a novel semi-supervised learning framework, which integrates both survival analysis and game theory. First, we carefully define the ϵ-adjacent network sequence, and make use of time stamp on each link to generate the baseline network evolution sequence. Next, to capture the law of network evolution, we employ the Cox Proportional Hazard Model (Cox PHM) to study the relative hazard associated with each temporal link, so as to estimate the coefficients of covariates, which are defined as a set of neighborhood based proximity features. To narrow the area of inquiry, we further propose a game theory based two-way selection mechanism to predict the future network topology. We finally encapsulate these two new technologies in a robust Autonomy-Oriented-Computing (AOC) multi-agent system, and propose a paralleled algorithm to conduct the temporal link prediction task. Extensive experiments were applied to real-world temporal networks to demonstrate both effectiveness and scalability of the proposed approach.

1. Introduction

To understand the collective behaviors among human beings, many studies have focused on analyzing the network data extracted from websites, online social platforms, and/or location based online social networks. In such networks; nodes represent actors, researchers or partners; and links indicate the cooperation or friendship among network participants. One of the most challenging tasks is to infer when a pair or even a group of people are in proximity in the near future, which is often referred to as the Link Prediction Problem in the literate [11,22,23]. Link prediction is not only important for understanding and capturing the evolution rules of real-world networks but also useful in many applications, such as recommending collaborators in a scientific network [14], predicting interactions between pairs of proteins [17], or recommending friends in social networks [19].

[*] Corresponding authors.
E-mail addresses: buzhan@nuaa.edu.cn (Z. Bu), caojie690929@163.com (J. Cao).

https://doi.org/10.1016/j.ins.2019.05.050

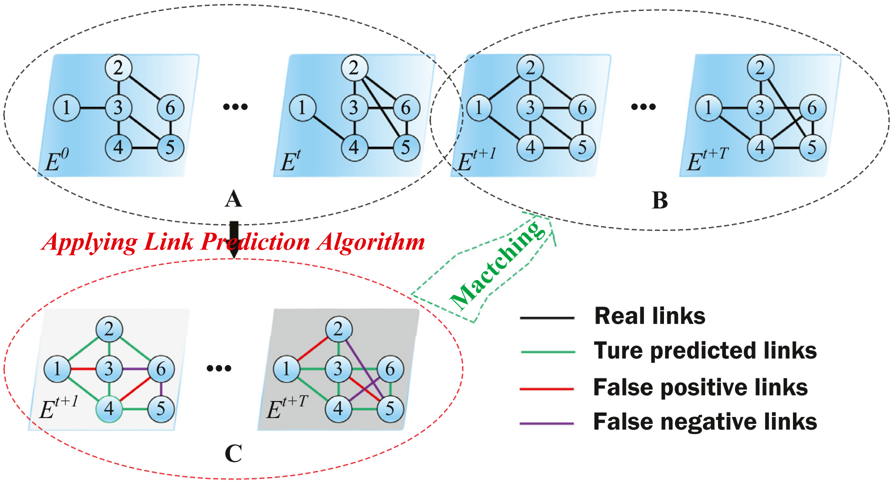

Fig. 1. The motivation of this work.

Broadly speaking, link prediction mainly addresses two types of problems: link existence and future link prediction. The former focuses on analyzing the network structures and gathering information about the links between various nodes. While the latter focuses on the problem of inferring future links in the underlying network [11]. Most traditional link prediction research focused on analyzing static networks, i.e., networks built as the results of aggregation of information over a certain period of time, where the network topology does not change over time. In fact, many real-world networks often have inherent uncertainty that is related to network participants that are constantly changing and evolving [1,2]. In such temporal networks, the underlying temporal information can be denoted by the timestamps of links. By choosing a set of consequent and/or overlapped observation windows, one can obtain a network evolution sequence. The circles *A* and *B* in Fig. 1 show a hypothetical example of such a network evolution sequence, which is constructed by six users (e.g., $\mathbf{V} = \{1, 2, 3, 4, 5, 6\}$) in an online social network. In each observation window, a different network characterizes the structure of interactions among the six participants. For example, link ⟨1, 3⟩ exists in the initial network if and only if users 1 and 3 have interacted with each other during the first time window. By observing the network dynamics from E^t to E^{t+1}, old links could disappear from E^t (i.e., link ⟨2, 5⟩) and new links can be established in E^{t+1} (i.e., link ⟨1, 2⟩). We note that both of the two types of events are common in real-world temporal networks. To perform link prediction in this scenario, we treat the network evolution sequences in circles *A* and *B* as training and testing data, respectively. By analyzing the network evolution sequence in the training data, the underlying link dynamics mechanism among a set of participants can be obtained. Then, we attempt to employ the learned link dynamics rules to predict the future network evolution. For example, circle *C* in Fig. 1 shows a predicted network evolution sequence, where most links are truly predicted (green) and some are falsely predicted (red and purple). To this end, the main goal of this study is to find an effective and efficient link prediction framework, which can bring the predicted network evolution sequence (circle *C*) to match the real one (circle *B*) as closely as possible.

Although plenty of work has been done in the direction of link prediction (see details in Section 2), there are still some issues which require more attention. First, to uncover the link dynamics rules in temporal networks, topological structure and time based information (i.e., timestamps of links) look like two independent heterogeneous data, which are difficult to effectively integrate. Although some time-conscious link prediction models have been proposed in the literature [30,36,41], the choice of priori distributions of the relationship establish time and/or the duration of an interaction in the statistical models requires a nontrivial expertise. Second, existing link prediction algorithms usually have high computational complexity, especially when predicting the potential links in large-scale temporal networks. How to reconcile the forecast accuracy and the computational complexity is still an open problem.

To address the above two challenges, we propose a novel solution framework based on survival analysis and game theory. In our proposed framework, the Cox Proportional Hazard Model (Cox PHM) is employed to uncover the link dynamics rules in temporal networks. When using the Cox PHM to find the likelihood of the existence of potential temporal links, its hazard function can nicely integrate topological features, the establishment time and the duration of the links. Moreover, one only needs to make parametric assumptions about the effects of topological features on the hazard function, but not about the shape of the hazard function itself. To conduct efficient link prediction, we further comprehend the link dynamics as the dynamic game played by two risk enthusiasts in a robust Autonomy-Oriented-Computing (AOC) multi-agent system. In such systems, the searching space of each agent is compressed to a small size as we only consider the local information in the corresponding projection network. The link dynamics identification process is carried out by all involved agents using a two-way selection mechanism. Specifically, any two agents will establish a new link with each other if and only if both of them are the optimal reconnected objects. Unlike existing approaches [20,42] which sort the proximity scores in decreasing order and choose a predefined number of links with top scores, the link dynamics process in our method can be executed by every agent independently. It is also worth noting that the designed link prediction algorithm based on our AOC system can nicely lead itself to parallelization, due to the fact that the topological feature update of every agent is independent of each other. The threefold contributions of this paper are summarized as follows:

- To describe the link dynamics in real temporal networks, we introduced the notion of the ϵ*-adjacent network sequence* that can extract the network evolution sequence from the raw temporal dataset (i.e., a set of links with timestamps). By specifying the duration of the observation window (i.e., δ) and the duration sliding rate (i.e., ϵ), one can generate the baseline network evolution sequences in different time scales.
- To unfold the link dynamics rules in real temporal networks, we associated each temporal link with a set of neighborhood based proximity scores and a survival time (i.e., measuring how long such a link exists in the future). By using the Cox PHM, the probability that a pair of two nodes will be connected or disconnected in the future can be computed. To our best knowledge, this is the first work to study the evolution mechanism of real temporal networks from the perspective of survival analysis.
- To simulate the link dynamics in real temporal networks, we proposed a dynamic game model. Specifically, during each period, two payoff-matrices are locally updated, the element of which indicates the probability that a pair of nodes will be connected or disconnected in the future. By searching the pure Nash equilibrium in these two matrices, one can forecast the network topology in the near future. Experiments showed that the proposed link prediction approach performs well by measuring the alignment between the predicted network evolution sequence and the real one.

The remainder of this paper is organized as follows. In Section 2, we review some related work on link prediction. Section 3 gives the problem definition, and some background on survival analysis and game theory. Section 4 provides our solution framework for link prediction. In Section 5, we present an Autonomy-Oriented-Computing (AOC) system and the algorithmic details to predict the network evolution sequence. Experiments are conducted in Section 6 to evaluate performance of our approach. We, finally, conclude this paper in Section 7.

2. Related work

Due to its numerous applications in various network-centered domains, the problem of link prediction has been extensively studied in recent years, which can be classified into four main categories, namely, 1) unsupervised methods, 2) supervised methods, 3) probabilistic model based methods, and 4) latent space model based methods. Please refer to articles [11,22,23] for a more complete review in this field.

Unsupervised Methods: Unsupervised methods, also known as scoring approaches, quantify the likelihood of the potential links by utilizing the predefined proximity function to score the proximity for each pair of nodes. The proximity score can be computed using various types of topological similarity metrics (see details in [22]). Many of the unsupervised methods are based on the idea that two nodes are similar and more likely to form a link in the future if they are connected to similar neighbors. Apart from Common Neighbors, there are many other measures based on the neighborhood of the nodes, such as Jaccard Coefficient, Salton Cosine Index, Sorensen Index, and etc. Some other proximity metrics, i.e., Rooted PageRank, Katz, PropFlow and Local Path, are based on the path information, they assume that two nodes can be similar if they have less distance between them. Once the proximity scores are calculated, the prediction can be made by sorting the proximity scores in decreasing order and choosing a pre-defined number of links with top scores. For example, Liben-Nowell and Kleinberg [20] proposed one of the earliest topology-based prediction models, in which some proximity metrics, including Common Neighbors, Preferential Attachment, Katz, etc., are tested. In the work of Zhou et al. [42], nine neighborhood based proximity metrics are compared and the results show that the Common Neighbors possesses the best performance. As mentioned by Lee et al. [18], any scoring approach can be used alone or combined with other unsupervised methods. In many cases, the scoring approaches may serve as a starting point for more sophisticated link prediction approaches.

Supervised Methods: As the baseline of existing links is always available from the history of the network, link prediction can be treated as a binary classification problem in which each pair of nodes is an instance, and being positive or being negative is a class label which indicates whether the pair of nodes are connected or not. There exist a plethora of classification models for link prediction, such as Logistic Regression, SVM-Light, Decision Tree, Multilayer Perceptron Neural Network and Naive Bayes. Most of the existing attempts were made to combine the efforts of different topological similarity metrics in order to enhance the overall prediction performance of the approach. For example, Sharan and Neville [34] proposed a time varying relational classification model, which is basically a relational model. It can exploit information of dynamic relationships between entities to improve classification and prediction accuracy. Rossi and Naville [32] proposed an ensemble method for predicting temporal relational dependency for classification, which outperformed competing models that ignored temporal information. By using techniques of data mining and machine learning, O' Madadhain et al. [28] presented a link prediction method for event-based temporal networks. Although supervised methods are powerful since they can utilize various features and measures, how to deal with the class imbalance problem is still a challenging hurdle. Compared with unsupervised methods, supervised approaches also have the difficulties in feature selection and are limited by computational cost and limitation of capacity, therefore they are not suitable for large-scale or dynamic networks.

Probabilistic Model based Methods: Generally, the structural features of a network can be represented by a probabilistic function of the uncertainty on the links and the topology features of the network. Along this line, some researchers have considered the use of various probabilistic models, such as Probabilistic Relational Model (PRM) [30], Local Probabilistic Model (LPM) [36] and Stochastic Relational Model (SRM) [41], to describe the uncertainty features of temporal networks. Tylenda et al. [35] developed a graph based link prediction approach, which extends the LPM to include time awareness. Hanneke et al. [15] proposed a family of statistical models for dynamic link prediction by extending the exponential random graph model. Gao et al. [13] proposed a model that exploits multiple information sources in the dynamic network to obtain link occurrence probabilities. Barbieri et al. [4] presented a stochastic link prediction model on directed attributed graphs. In addition to predicting links, their model can also provide explanations for the detected links. A recent work proposed by Ahmed and Chen [1] transformed the link prediction problem in temporal uncertain network to a random walk in a deterministic network. By integrating the temporal and global topological information in temporal uncertain networks, their approach can achieve higher quality results than other similar methods.

Latent Space Model based Methods: Recent work has explored static latent space modeling for link prediction, which assumes that each node lies in an unobserved latent space, and interactions are more likely to occur between similar nodes in the latent space representation. To infer the latent space, matrix factorization approaches are often applied to observed networks to learn the low-dimension latent space representation. For example, Yang et al. [39] proposed a multiplicative algorithm with a random walk graph to solve the symmetric graph factorization problem. However, their approach performs poorly as the density of the input graph increases. Moreover, it does not scale well due to the high computation cost in each iteration. Zhu et al. [43] applied the symmetric matrix factorization approaches directly on the observed networks to infer low-rank latent spaces; in their work, a scalable block coordinate gradient descent algorithm was proposed. In addition to matrix factorization approaches, Yin et al. [40] proposed an efficient stochastic variational inference algorithm and a parsimonious triangular model to infer the latent spaces of large networks. Xie et al. [38] proposed a label propagation-based approach that incrementally detects latent positions at each period with conditional updates on a set of affected nodes. Sarkar and Moore [33] generalized the classical multidimensional scaling to get an initial estimate of the positions in the latent space, and then applied nonlinear optimization directly to obtain the latent space representation.

3. Preliminaries

In this section, we will first give the problem definition, and then revisit *Scoring Methods, Survival Analysis* and *Game Theory* for explaining some basic concepts.

3.1. Problem definition

Many real-world systems, such as those arising from biological and social interactions, can be modeled by networks. In such networks, each node represents a participant, and the links represent the existence of connections between nodes. Usually, real-world networks are not static as the links between objects dynamically change over time. Such *temporal networks* can be represented by a series of *temporal links* [16]. Formally, a temporal network $\mathcal{T}$ is a collection of tuples $\langle i_p, j_p, \tau_p\rangle$, $p = 1, 2, \cdots, M$, where a specific $\langle i_p, j_p, \tau_p\rangle$ tuple can be viewed as a *temporal link*, indicating that nodes i_p and j_p are connected at timestamp τ_p.

Definition 1. Induced Static Network: For a given collection of temporal links in the time window $[t_s, t_e]$, where t_s and t_e are the start and end timestamps of such window. The induced static network $\mathcal{G}^t$ is created by ignoring timestamps and duplicate links.

In other words, a link $\langle i, j\rangle$ exists in the static network $\mathcal{G}^t$ if and only if there is a temporal link $\langle i_p, j_p, \tau_p\rangle$ in the time window $[t_s, t_e]$ (i.e., $t_s < \tau_p < t_e$). Inspired by the recently proposed methodology for analyzing temporal networks [29], the *δ-temporal motif* can be defined as follows:

Definition 2. δ-Temporal Motif: A δ-temporal motif during period t is a sequence of time ordered temporal links $\mathbb{M}^t(\delta) = \langle i_{t_s}, j_{t_s}, \tau_{t_s}\rangle, \cdots, \langle i_{t_e}, j_{t_e}, \tau_{t_e}\rangle$, within a δ-duration, i.e., $\tau_{t_s} \le \cdots \le \tau_{t_e}$ and $\delta = \tau_{t_e} - \tau_{t_s}$, such that the induced static δ-network $\mathcal{G}^t(\delta) = (V^t(\delta), E^t(\delta)), E^t(\delta) \subseteq V^t(\delta) \times V^t(\delta)$ is connected.

The above definition provides a template for a particular pattern, an elusive question is - how those node pairs $\forall(i, j) \in V^t(\delta) \times V^t(\delta)$ evolve in the future. To answer this question, a novel concept of *ϵ-adjacent* is introduced as follows:

Definition 3. ϵ-Adjacent: Given two overlapping δ-durations, i.e., $[t_s, t_e]$ and $[t'_s, t'_e]$, such that $\tau_{t_e} - \tau_{t_s} = \tau_{t'_e} - \tau_{t'_s} = \delta$ and $t_s \le t'_s \le t_e \le t'_e$, the duration sliding rate, denoted by ϵ, is defined as $\epsilon = \frac{t'_s - t_s}{\delta}$. In this context, we say that the static network $\mathcal{G}^t$ is ϵ-adjacent to $\mathcal{G}^{t'}$ (Note that $\mathcal{G}^t$ and $\mathcal{G}^{t'}$ may be unconnected).

Given a snapshot of the static network $\mathcal{G}^t$, the dynamics from $\mathcal{G}^t$ to its ϵ-adjacent network $\mathcal{G}^{t'}$ can be described by three aspects, as shown in Fig. 2: 1) the missing links from $\mathcal{G}^t$; 2) the reconnected links between any two nodes in $\mathcal{G}^t$; and 3) the new links ended by at least one external unknown node. We remark that the last aspect is unpredictable in many real-world applications, due to the great randomness of the external information. In this paper, we assume that nodes present across all periods are fixed. Formally, given an initial static network derived by some δ-temporal motif, the *ϵ-adjacent network sequence* is defined as follows:

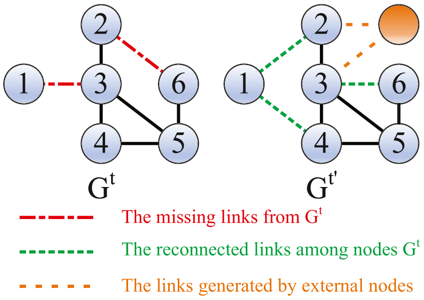
2
1
3
6
4
5
G^t
2
1
3
6
4
5
$G^{t'}$
The missing links from G^t
The reconnected links among nodes G^t
The links generated by external nodes

Fig. 2. The network dynamics from the toy network ($\mathcal{G}^t$) to its ϵ-adjacent network ($\mathcal{G}^{t'}$), where red dotted links are disappeared from $\mathcal{G}^t$, green dotted links are reconnected by existing nodes in $\mathcal{G}^t$, while yellow dotted links are formed by at least one unknown external node. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Definition 4. ϵ-Adjacent Network Sequence: The ϵ-adjacent network sequence, starting from an initial δ-network $\mathcal{G}^0(\delta) = (\mathbf{V}, E^0)$, is a sequence of coupled static networks, i.e., $\Gamma^\epsilon(\mathcal{G}^0(\delta)) = \{\mathcal{G}^1, \mathcal{G}^2, \cdots\}$, such that for every $t \geq 1$: 1) $\mathcal{G}^{t-1}$ is ϵ-adjacent to $\mathcal{G}^t$; 2) $V^t \subseteq \mathbf{V}$, $E^t \subseteq \mathbf{V} \times \mathbf{V}$.

In the following sections of this paper, we will sometimes write a network snapshot $\mathcal{G}^t$ as E^t to emphasize node connection during period *t*. To further emphasize network dynamics from $\mathcal{G}^{t-1}$ to $\mathcal{G}^t$, the missing and reconnected links w.r.t. $\mathcal{G}^{t-1}$ are denoted by M^t and R^t, respectively. With these notations, we have $M^t = E^{t-1} - E^t$ and $R^t = E^t - E^{t-1}$. Thus, the problem of *link prediction* is formally defined as follows:

Problem 1 (Link Prediction)**.** Given an initial static network derived by some δ-temporal motif $\mathcal{G}^0(\delta)$, the problem of link prediction is to generate a predicted network evolution sequence, i.e., $\widetilde{\Gamma}(\mathcal{G}^0(\delta)) = \{\widetilde{E}^1, \widetilde{E}^2, \cdots\}$, such that for every $t \geq 1$, the predicted $\widetilde{E}^t$ can match some snapshot network in the real ϵ-adjacent network sequence (e.g., $E^{t'} \in \Gamma^\epsilon(\mathcal{G}^0(\delta))$) as well as possible.

3.2. Scoring methods

The most straightforward approaches for solving the link prediction problem are the so-called *scoring methods*. In scoring methods, a number of proximity functions that measure the similarity (or proximity) between the participants of the network are defined. Along this line, each node pair of the network is assigned with a certain score by a given proximity function, and the score itself can represent the probability of the existence of the link. In the literature, the proximity functions can be defined in various ways, with each method designed to reflect a specific aspect of the network topology, such as the number of neighbors, the distance, and/or the communities. Once the link scores are calculated, the prediction can be made by sorting the link scores in decreasing order and choosing a predefined number of links with top scores.

Consider a general undirected and unweighted attributed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{L})$, where each node $i \in \mathbf{V}$ is associated with a set of neighbors (e.g., $N_i = \{j : (i, j) \in \mathbf{E}\}$). We expect that the higher two nodes' topological similarity index is, the more likely they may exhibit a future link. In existing literature, topological similarity indices may be characterized by local, quasi-local, or global measures. Since global similarity measures are computationally laborious for large networks, in this paper, we only consider the neighborhood-based similarities:

Definition 5. Neighborhood based Proximity Functions: Our proximity functions are chosen from some well known neighborhood-based similarities in [22]:

- ***Common Neighbor Index (CN):*** $\lambda_{ij}^{[CN]} = |N_i \cap N_j|$.
- ***Jaccard Coefficient (JC):*** $\lambda_{ij}^{[JC]} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$.
- ***Salton Index (Sa):*** $\lambda_{ij}^{[Sa]} = \frac{|N_i \cap N_j|}{\sqrt{|N_i||N_j|}}$.
- ***Sorensen Index (So):*** $\lambda_{ij}^{[So]} = \frac{2|N_i \cap N_j|}{|N_i|+|N_j|}$.
- ***Hub Promoted Index (HP):*** $\lambda_{ij}^{[HP]} = \frac{|N_i \cap N_j|}{\min(|N_i|,|N_j|)}$.
- ***Hub Depressed Index (HD):*** $\lambda_{ij}^{[HD]} = \frac{|N_i \cap N_j|}{\max(|N_i|,|N_j|)}$.
- ***Leicht-Holme-Newman Index (LN):*** $\lambda_{ij}^{[LN]} = \frac{|N_i \cap N_j|}{|N_i||N_j|}$.
- ***Preferential Attachment Index (PA):*** $\lambda_{ij}^{[PA]} = |N_i||N_j|$.
- ***Adamic-Adar Index (AA):*** $\lambda_{ij}^{[AA]} = \sum_{k \in N_i \cap N_j} \frac{1}{|N_k|}$.

As mentioned by Lee et al. [18], any proximity function can be used alone or combined with other scoring methods. Typically, the scoring method may serve as a starting point for more sophisticated link prediction approaches.

3.3. Survival analysis

Survival analysis [24] models the time it takes for events to occur. The prototypical event is “death” in the medical field, from which the notion of “survival analysis” derives, but the applications of survival analysis are much broader. Essentially the same methods are employed in a variety of disciplines under various rubrics, i.e., “event-history analysis” in sociology and “failure-time analysis” in engineering.

Let T denote the survival time, which can be regards as a random variable with cumulative distribution function $P(t) = Pr(t \le T)$. The more optimistic survival function $S(t)$ is the complement of the distribution function, i.e., $S(t) = Pr(t > T) = 1 - P(t)$. The representation of the distribution of survival times is the so-called *hazard function*, which assesses the instantaneous risk of demise at the time interval $[t, t + \Delta t]$, conditional on survival to or beyond t, which is defined as

$$h(t) = \lim_{\Delta t \to 0} \frac{Pr(t \le T \le t + \Delta t | T \le t)}{\Delta t}. \tag{1}$$

Survival analysis typically examines the relationship of the survival distribution to a set of features (or covariates). Most commonly, this examination entails the specification of a linear-like model for the log hazard. In the literature, the most commonly used model for survival analysis is the *Cox Proportional Hazard Model (Cox PHM) [21]*. Formally, let $\Omega = \{\Omega_p : p \in [1, N]\}$ be a typical survival dataset that consists of N observation samples (or subjects). Each sample in Ω can be represented by a three-tuple, i.e., $\Omega_p = \langle T_p, \delta_p, \vec{\mathbf{F}}_p \rangle$, where T_p denotes the survival time for the pth subject, δ_p is the $0-1$ binary censoring indicator, and $\vec{\mathbf{F}}_p = (\mathbf{f}_{p1}, \mathbf{f}_{p2}, \cdots, \mathbf{f}_{pd})^{\mathrm{T}} \in \mathbb{R}^d$ is the feature vector associated with the pth subject, respectively. The Cox PHM formulates the hazard function $h(t|\vec{\mathbf{F}}_p)$ for the pth subject as

$$h(t|\vec{\mathbf{F}}_p) = h_0(t) \exp(\vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_p), \tag{2}$$

where $h_0(t)$ is the unspecified baseline hazard function, representing how the hazard changes with time; $\vec{\mathbf{W}} = (\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_d)^{\mathrm{T}} \in \mathbb{R}^d$ is the vector of unknown regression coefficients. Usually, the coefficient vector $\vec{\mathbf{W}}$ can be estimated by maximizing the partial likelihood as follows

$$\mathfrak{L}(\vec{\mathbf{W}}) = \prod_{p=1}^{N} \left(\frac{\exp \vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_p}{\sum_{q \in \mathbf{R}(t_p)} \vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_q} \right)^{\delta_p}, \tag{3}$$

where t_p is the time unit that the pth subject suddenly “dies” (in different application scenarios, the term “dies” can be also interpreted as some event suddenly happens); $\mathbf{R}(t_p)$ is the set of subjects which are observed to be suddenly “died” at the time unit t_p. Taking the logarithm, we have

$$\log \mathfrak{L}(\vec{\mathbf{W}}) = \sum_{p=1}^{N} \delta_p \left(\vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_p - \log \sum_{q \in \mathbf{R}(t_p))} \vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_q \right). \tag{4}$$

Obviously, $\log \mathfrak{L}(\vec{\mathbf{W}})$ only considers probabilities for subjects who fail; but does not consider probabilities for censored subjects explicitly. Similar to the Logistic regression, one can solve the $\log \mathfrak{L}(\vec{\mathbf{W}})$ maximization problem by using the Newton–Raphson method. The resultant maximum partial log-likelihood estimator, i.e., $\vec{\mathbf{W}}^*$ can be learned by

$$\vec{\mathbf{W}}^* = \arg_{\vec{\mathbf{W}} \in \mathbb{R}^d} \max(\log \mathfrak{L}(\vec{\mathbf{W}})). \tag{5}$$

The Cox PHM is often called semi-parametric prediction: one only needs to make parametric assumptions about the effects of the features on the hazard function, but not about the shape of the hazard function itself. This is an important and appealing feature of the Cox PHM. If we were interested only in the hazard ratios, we could disregard distributional assumptions about the event times. In other words, we assume that any two hazard rates predicted by the model are proportional over time. More formally, the relative hazard for any two observations p and q must obey the following relationship:

$$\frac{h(t|\vec{\mathbf{F}}_p)}{h(t|\vec{\mathbf{F}}_q)} = \frac{h_0(t) \exp(\vec{\mathbf{W}}^{*\mathrm{T}} \vec{\mathbf{F}}_p)}{h_0(t) \exp(\vec{\mathbf{W}}^{*\mathrm{T}} \vec{\mathbf{F}}_q)} = \frac{\vec{\mathbf{W}}^{*\mathrm{T}} \vec{\mathbf{F}}_p}{\vec{\mathbf{W}}^{*\mathrm{T}} \vec{\mathbf{F}}_q}. \tag{6}$$

Therefore, the relative risk, i.e., a proportionate increase or decrease in risk, w.r.t. subject Ω_p with the feature vector $\vec{\mathbf{F}}_p$, can be defined as

$$\gamma(\vec{\mathbf{W}}^*, \vec{\mathbf{F}}_p) = \vec{\mathbf{W}}^{*\mathrm{T}} \vec{\mathbf{F}}_p. \tag{7}$$

3.4. Basic concepts of game theory

Consider a standard game with K-players, i.e., $\mathbf{P} = \{1, 2, \cdots, K\}$, let $\mathcal{S}_\alpha = \{s_\alpha^1, s_\alpha^2, \cdots, s_\alpha^{D_\alpha}\}$ denote the set of D_α pure strategies associated with player α, where s_α^g, $g \in \{1, 2, \cdots, D_\alpha\}$ is the gth pure strategy of player α. In a standard

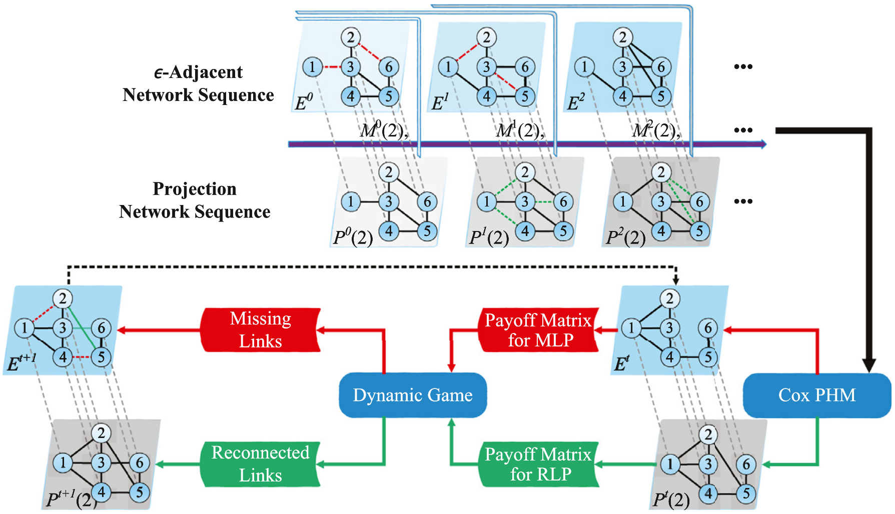

Fig. 3. The framework for link prediction in temporal networks: (Top side) During each period $t = 0, 1, 2,$ a different network characterizes the structure of interactions between six participants (e.g., $\mathbf{V} = \{1, 2, 3, 4, 5, 6\}$); the set of corresponding projection networks where each different colored network is the projection of the latest two networks. (Bottom side) The problem of link prediction in temporal networks is translated into two sub-problems, i.e., missing link prediction (MLP) and the reconnected link prediction (RLP). To address these two sub-problems, the Cox PHM and the dynamic game are combined.

game, each player has a utility function $u_\alpha(\cdot) : \mathbf{S} \to \mathbb{R}$ (to be maximized), where $\mathbf{S} = \prod_{\alpha=1}^{K} \mathcal{S}_\alpha$ ($\prod$ is the Cartesian product) denotes the set of joint strategies played by all players. A strategy point $\mathbf{s} \in \mathbf{S}$ is a K-tuple, i.e., $\mathbf{s} = \langle s_1, s_2, \cdots, s_K \rangle$, with each item $s_\alpha \in \mathcal{S}_\alpha$ associated with a pure strategy of α. While $\mathbf{s}_{-\alpha} \in \mathbf{S}_{-\alpha}$ denotes the other players' strategies, i.e., $\mathbf{s}_{-\alpha} = \langle s_1, \cdots, s_{\alpha-1}, s_{\alpha+1}, \cdots, s_K \rangle$ and $\mathbf{S}_{-\alpha} = \prod_{\beta \neq \alpha} \mathcal{S}_\beta$. With this notation, we will sometimes write a strategy point $\mathbf{s}$ as $(s_\alpha, \mathbf{s}_{-\alpha})$ to emphasize player α's strategy.

Definition 6. Nash equilibrium problem (NEP): The NEP is the problem of finding a strategy point $\mathbf{s}^* \in \mathbf{S}$ such that no player can unilaterally optimize the utility by changing her strategy, i.e.,

$$u_i(s_\alpha^*, \mathbf{s}_{-\alpha}^*) \geq u_i(s_\alpha, \mathbf{s}_{-\alpha}^*), \forall s_\alpha \in \mathcal{S}_\alpha, \tag{8}$$

Such a point $\mathbf{s}^*$ is called a **pure Nash equilibrium (PNE)**.

We refer the interested readers to [25,27] for a state-of-the-art introduction on standard game theory.

4. The framework of our solution

In this section, we give the details about how to combine the survival analysis and the game theory to solve the link prediction problem as introduced in Problem 1. The overall solution framework is shown in Fig. 3. We first employ the ϵ-*adjacent network sequence* (see Definition 4) to extract the baseline network evolution sequence from real temporal network dataset. Then, we associate each temporal link with a set of neighborhood based proximity scores (see Definition 5); and further employ the Cox PHM to learn the relative hazard associated with each node pair, such relative hazard can be interpreted as the probability that a pair of two nodes will be connected or disconnected in the future. Finally, to reconcile the prediction effectiveness and the computational cost, we formulate the link prediction problem as a dynamic game, by locally searching the pure Nash equilibrium in the induced payoff-matrices, one can forecast the future network topology based on the current network snapshot and the corresponding projection network (see Definition 7) in a given time window.

4.1. Novel concepts w.r.t. link prediction

Although many works have been proposed during the last decade to predict the existence of a link in temporal networks, most of the existing approaches either focus on predicting the missing links or focus on the problem of reconnected link prediction. To effectively handle the two sub-tasks of link prediction, we introduce several novel concepts with respect to Problem 1 as follows.

Definition 7. Multiplex Network Model: The multiplex network w.r.t. the tth snapshot of an ϵ-adjacent network sequence $\Gamma^\epsilon(\mathcal{G}^0(\delta)) = \{E^1, E^2, \cdots\}$, is a special type of multilayer network [26] with two layers:

$$\mathcal{M}^t(L) = \{E^t, P^t(L)\}, \tag{9}$$

where $E^t \subseteq \mathbf{V} \times \mathbf{V}$ is the link set during period t, and $P^t(L) \subseteq \mathbf{V} \times \mathbf{V}$ denotes the link set of the corresponding projection network [12] of the latest L networks ended by $\mathcal{G}^t$, such that

$$P^t(L) = \begin{cases} \cup_{l=0}^{t} E^l & 0 \le t < L-1 \\ \cup_{l=1}^{L} E^{t-l+1} & t \ge L-1 \end{cases}, \tag{10}$$

where $L(L \ge 1)$ is the time-window length, which can be user-specified. Thus the corresponding **multiplex network sequence** is represented as $MNS(L) = \{\mathcal{M}^0(L), \mathcal{M}^1(L), \cdots\}$ (see the schematic illustration of Fig. 3).

Definition 8. Missing Link Sequence: The missing link sequence w.r.t. a given ϵ-adjacent network sequence $\Gamma^\epsilon(\mathcal{G}^0(\delta))$ is defined as $\Gamma^M(\mathcal{G}^0(\delta)) = \{M^1, M^2, \cdots\}$, such that $\forall t \ge 1$, $M^t = E^{t-1} - E^t$ (see the red dot dash lines in Fig. 3).

Definition 9. Reconnected Link Sequence (RLS): The reconnected link sequence w.r.t. a given ϵ-adjacent network sequence $\Gamma^\epsilon(\mathcal{G}^0(\delta))$ is defined as $\Gamma^R(\mathcal{G}^0(\delta)) = \{R^1, R^2, \cdots\}$, such that $\forall t \ge 1$, $R^t = E^t - E^{t-1}$ (see the green dash lines in Fig. 3).

Based on the three important notions above, we remark that Problem 1 can be formulated as a semi-supervised classification problem: Given an ϵ-adjacent network sequence $\Gamma^\epsilon(\mathcal{G}^0(\delta))$, how to make use of the labeled instances in the related $\Gamma^M(\mathcal{G}^0(\delta))$ and $\Gamma^R(\mathcal{G}^0(\delta))$ to infer the classification rules for *Missing Link Prediction (MLP)* and *Reconnected Link Prediction (RLP)*, respectively. As the entire training data could be timing-dependent, we attempt to employ the Cox PHM for the analysis of such early-failure data associated with time-varying networks.

4.2. Covariate Learning based on the Cox PHM

Given a specific link in the snapshot of network during period t, i.e., $(i, j)^t \in E^t$, we associate it with a set of features (or covariates) derived by the neighborhood based proximity functions (see Definition 5) over the corresponding static/projection network (e.g., E^t or $P^t(L)$); we then observe how long (e.g., the number of periods) the two related nodes (i and j) can stay connected with each other in the future. In this way, we can generate a set of survival (or failure) data for missing link prediction. On the contrary, if we focus our attention on those unconnected node pairs, i.e., $(i, j)^t \notin E^t$; we can also collect a set of survival (or failure) data for reconnected link prediction. Finally, we can apply the Cox PHM on each dataset respectively, to analyze all the selected features and further identify those which have significant effects on the disappearance and/or appearance of a specific link. We first introduce two survival data generation models as follows.

Definition 10. Survival Data Generation Model for Missing Link Prediction (SDGM-MLP): Given a multiplex network sequence $MNS(L) = \{\mathcal{M}^0(L), \mathcal{M}^1(L), \cdots\}$ and the corresponding missing link sequence $\Gamma^M(\mathcal{G}^0(\delta)) = \{M^1, M^2, \cdots\}$, the survival data for MLP is defined as $\Omega^M = \{\langle T_p, \vec{\mathbf{F}}_p^M \rangle : p \in [1, \sum_t |M^t|]\}$. Each sample in Ω^M characterizes one missing link in $\Gamma^M(\mathcal{G}^0(\delta))$, i.e., $(i, j)_p \in M^t$, which can be represented by a two-tuple $\langle T_p, \vec{\mathbf{F}}_p^M \rangle$, where T_p indicates the survival time of the corresponding connected node pair $(i, j)_p$, such that $(i, j)_p \in E^l$, $0 \le t - T_p^M \le l \le t$; and $\vec{\mathbf{F}}_p^M$ is the feature vector associated with $(i, j)_p$, which is calculated according to the neighborhood based proximity functions (see Definition 5) on the static network E^{t-T_p}.

Definition 11. Survival Data Generation Model for Reconnected Link Prediction (SDGM-RLP): Given a multiplex network sequence $MNS(L) = \{\mathcal{M}^0(L), \mathcal{M}^1(L), \cdots\}$ and the corresponding reconnected link sequence $\Gamma^R(\mathcal{G}^0(\delta)) = \{R^1, R^2, \cdots\}$, the survival data for RLP is denoted by $\Omega^R = \{\langle T_q, \vec{\mathbf{F}}_q^R \rangle : q \in [1, \sum_t |R^t|]\}$, where each sample characterizes one reconnected node pair in Γ^R, i.e., $(i, j)_q \in R^t$. Such node pair can be also represented by a two-tuple like $\langle T_q, \vec{\mathbf{F}}_q^R \rangle$, where T_q indicates the survival time of the corresponding unconnected node pair $(i, j)_q$, such that $(i, j)_q \notin E^l$, $0 \le t - T_q^R \le l \le t$; and $\vec{\mathbf{F}}_q^R$ is the feature vector associated with $(i, j)_q$, over the projection network $P^{t-T_q}(L)$.

In this paper, we focus our attention on the relative importance of covariates on the hazard, and employ the Cox PHM to find the maximum partial log-likelihood estimator. As the Cox PHM only considers probabilities for subjects who fail, but does not consider probabilities for censored subjects explicitly (see Eq. (4)); we thus ignore the censored subjects in Definitions 10 and 11. Therefore, given the survival datasets Ω^M and Ω^R, the resultant maximum partial log-likelihood estimators for MLP ($\vec{\mathbf{W}}^{M*}$) and RLP ($\vec{\mathbf{W}}^{R*}$) can be learned by

$$\vec{\mathbf{W}}^{M*} = \arg_{\vec{\mathbf{W}} \in \mathbb{R}^d} \max \sum_p \left(\vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_p^M - \log \sum_{o \in \mathbf{R}(t_p))} \vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_o^M \right) \tag{11}$$

and

$$\vec{\mathbf{W}}^{R*} = \arg_{\vec{\mathbf{W}} \in \mathbb{R}^d} \max \sum_q \left(\vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_q^R - \log \sum_{o \in \mathbf{R}(t_q))} \vec{\mathbf{W}}^{\mathrm{T}} \vec{\mathbf{F}}_o^R \right), \tag{12}$$

respectively. Formally, let $\mathcal{A}^t = [A_{ij}^t] \in \mathbb{R}^{n \times n}$ denote the adjacency matrix of network $\mathcal{G}^t$, where A_{ij}^t equals 1 if $(i, j) \in E^t$ and 0 otherwise. For the sake of notational simplicity, let $\vec{\mathbf{F}}_{ij}^{t[M]}$ and $\vec{\mathbf{F}}_{ij}^{t[R]}$ denote the feature vectors associated with node pair (i, j)

w.r.t. E^t and $P^t(L)$ respectively. Based on the maximum partial log-likelihood estimators (see Eqs. (11) and (12)); the relative risks associated with node pair (i, j) being unconnected/connected in period $t+1$, are defined as

$$\theta_{ij}^t = A_{ij}^t (\vec{\mathbf{W}}^{M*})^{\mathrm{T}} \vec{\mathbf{F}}_{ij}^{t[M]}, \tag{13}$$

and

$$\phi_{ij}^t = (1 - A_{ij}^t - \varrho_{ij})(\vec{\mathbf{W}}^{R*})^{\mathrm{T}} \vec{\mathbf{F}}_{ij}^{t[R]}, \tag{14}$$

respectively, where ϱ_{ij} is 1 if $i = j$, and 0 otherwise.

4.3. Link prediction based on game theory

In this paper, we formulate the link prediction problem as a dynamic game, which is played by two risk enthusiasts, i.e., α and β, in a discrete-time dynamic system (indexed by $t = 0, 1, \cdots$). During every period t, each speculator has the same strategy space, i.e., $\mathcal{S}_\alpha = \mathcal{S}_\beta = \mathbf{V} = \{1, 2, \cdots, n\}$. To emphasize each speculator's selection, let $i \in \mathcal{S}_\alpha$ and $j \in \mathcal{S}_\beta$ denote the strategies of α and β, respectively. In this context, a strategy point, i.e., $\mathbf{s}^t = \langle i, j\rangle \in \mathbf{V} \times \mathbf{V}$, can be also viewed as a node pair $(i, j)^t$ in network $\mathcal{G}^t$; and the payoff-matrix during period t, denoted by Ψ^t, can be written as

$$\Psi^t = \begin{pmatrix} u_\alpha^t(1,1)\ u_\beta^t(1,1) & \cdots & u_\alpha^t(1,n)\ u_\beta^t(1,n) \\ \cdots & \cdots & \cdots \\ u_\alpha^t(n,1)\ u_\beta^t(n,1) & \cdots & u_\alpha^t(n,n)\ u_\beta^t(n,n) \end{pmatrix} \tag{15}$$

We further assume that for a given strategy point $\langle i, j\rangle$, the two super speculators α and β also have the same payoff values, such that $u_\alpha^t(i, j) = u_\beta^t(i, j) = u_{ij}^t$. In this way, the payoff-matrix Ψ^t can be simplified as

$$\Psi^t = \begin{pmatrix} u_{11}^t & \cdots & u_{1n}^t \\ \cdots & \cdots & \cdots \\ u_{n1}^t & \cdots & u_{1n}^t \end{pmatrix} \tag{16}$$

Thus, the set of PNEs (see Definition 6) associated with Ψ^t can be solved by

$$\mathfrak{N}^t = \{\langle i, j\rangle : \max_i \max_j \{u_{ij}^t\} = \max_j \max_i \{u_{ij}^t\}\}, \tag{17}$$

implying that if $\langle i, j\rangle$ is a PNE during period t, u_{ij}^t must be the largest value in both ith row and jth column of the corresponding payoff-matrix Ψ^t.

Concerned with the original link prediction problem, one can defined Ψ^t as $\Theta^t = [\theta_{ij}^t] \in \mathbb{R}^{n\times n}$ or $\Phi^t = [\phi_{ij}^t] \in \mathbb{R}^{n\times n}$. The corresponding sets of PNEs associated with Θ^t and Φ^t are then denoted by $\mathfrak{N}^{t[M]}$ and $\mathfrak{N}^{t[R]}$, respectively. Thus each element $\langle i, j\rangle^{[M]} \in \mathfrak{N}^{t[M]}$ represents a pair of connected nodes, i.e., i and j, that both are willing to break off their current relationship. On the contrary, each element $\langle i, j\rangle^{[R]} \in \mathfrak{N}^{t[R]}$ implies that both i and j want to establish a new relationship with each other. Therefore, the network-transition function w.r.t. Problem 1 can be defined as the following form:

$$\Upsilon(E^t, \Theta^t, \Phi^t) = E^t - \mathfrak{N}^{t[M]} + \mathfrak{N}^{t[R]}, \tag{18}$$

which can forecast the topology structure of the future network $\widetilde{E}^{t+1}$, based on the current network E^t.

Definition 12. Predicted Network Evolution Path (PNEP): A PNEP w.r.t. Problem 1 is a sequence of static networks, i.e., $\widetilde{\Gamma}(\mathcal{G}^0(\delta)) = \{\widetilde{E}^1, \widetilde{E}^2, \cdots\}$, such that for every $t \geq 1$, $\widetilde{E}^t = \Upsilon(\widetilde{E}^{t-1}, \widetilde{\Theta}^{t-1}, \widetilde{\Phi}^{t-1})$, where $\mathcal{G}^0(\delta)$ can be any arbitrary initial δ-network.

Assumption 1. For every $t \geq 1$, the degree distribution of $\mathcal{G}^t$ (or E^t) follows the power law distribution.

Corollary 1. *With Assumption 1, the degree distribution of $\mathcal{G}^t$ can be simplified as $p^t(k) = (a^t - 1)k^{-a^t}$, where $a^t > 1$ is the constant parameter to be estimated during period t.*

Proof. Without loss of generality, let $p^t(k) = C^t k^{-a^t}$ denote the general form degree distribution of the static network during period t. To estimate C^t, the following condition must be established:

$$\int_{k_{min}}^{\infty} p^t(k)dk = \int_{k_{min}}^{\infty} C^t k^{-a^t} dk = \frac{C^t}{a^t - 1} k_{min}^{-a^t+1} = 1,$$

from which, rearranging terms, one obtains

$$C^t = (a^t - 1)k_{min}^{a^t-1}$$

In many real induced δ-networks, as $k_{\text{min}} = 1$ (see Fig. 4), C^t can be simplified as $a^t - 1$, the proof completes. □

Lemma 1. *Based on the network-transition functions (see Eqs. (17) and (18)) and Corollary 1, the expected probability for any arbitrary link $(i, j) \in E^t$ to be disconnected in period $t+1$, is estimated to be $(\frac{a^t-1}{a^t})^2$.*

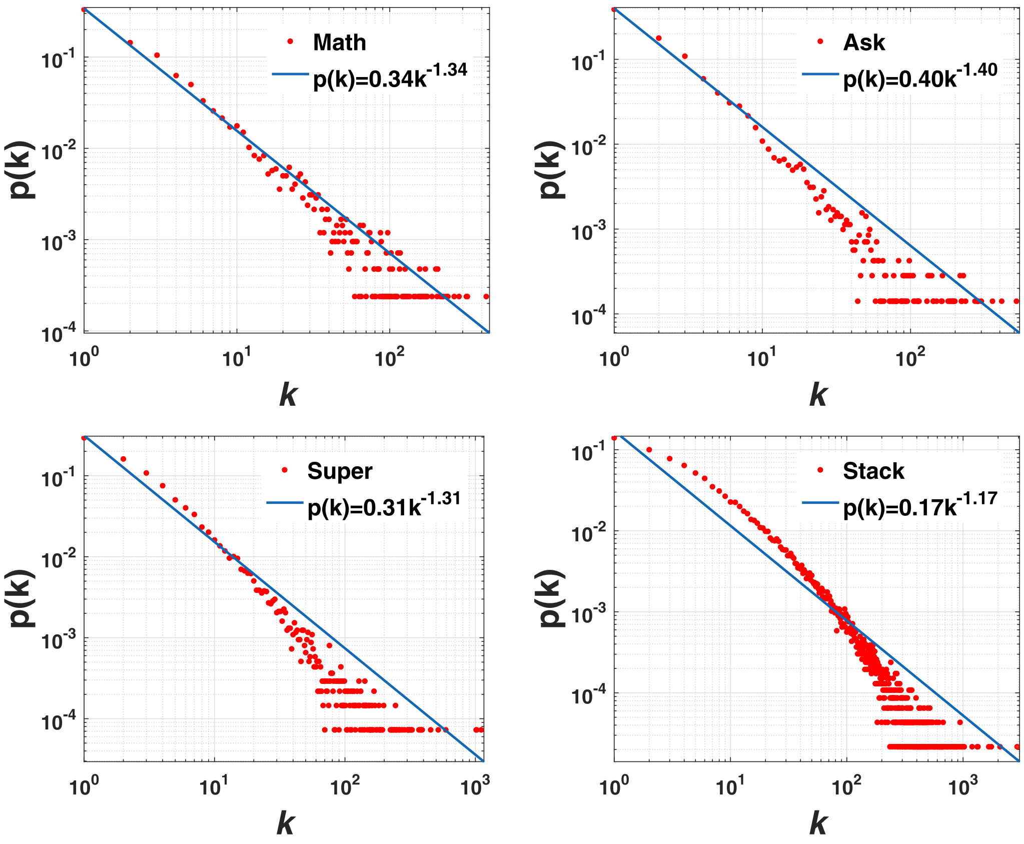

Fig. 4. Degree distributions of the four induced δ-networks.

Proof. During every period t, the expected size of $\mathfrak{N}^{t[M]}$ can be written as

$$Exp.(|\mathfrak{N}^{t[M]}|) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{A_{ij}^t}{k_i^t k_j^t},$$

where k_i^t and k_j^t are the degree of nodes i and j during period t. Let ϵ_{ij}^t denote the expected probability for any arbitrary link $(i, j) \in E^t$ to be disconnected in period $t+1$, which can be estimated by

$$\epsilon_{ij}^t = \frac{Exp.(|\mathfrak{N}^{t[M]}|)}{m^t} \approx \frac{\sum_{i=1}^n \sum_{j=1}^n \frac{1}{k_i^t k_j^t}}{n^2} = \left(\frac{\sum_{i=1}^n \frac{1}{k_i^t}}{n}\right)^2$$

Based on Corollary 1, one can write $\frac{1}{n}\sum_{i=1}^n \frac{1}{k_i^t}$ as

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}\frac{1}{k_i^t} &= \frac{1}{n}\int_1^n \frac{np^t(k)}{k}dk = \int_1^n (a^t-1)k^{-a^t-1}dk \\ &= \frac{a^t-1}{a^t}(1-n^{-a^t}) \overset{n\to\infty,\ a^t>1}{=} \left(\frac{a^t-1}{a^t}\right)\end{aligned}$$

Thus, ϵ_{ij}^t can be simplified as $(\frac{a^t-1}{a^t})^2$, the proof completes. □

Lemma 2. *Given that $\forall i, t,\ n \gg k_i^t$ and $a^t > 1$, the expected probability for any arbitrary node pair $(i, j) \notin E^t$ to be connected in period $t+1$, is estimated to be $(\frac{(a^t-1)\ln n}{n a^t})^2$.*

Proof. During every period t, the expected size of $\mathfrak{N}^{t[R]}$ is denoted as

$$Exp.(|\mathfrak{N}^{t[R]}|) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1-A_{ij}^t}{(n-1-k_i^t)(n-1-k_j^t)},$$

Table 1
Terms and notations.

Notations	Description
$\mathbb{AOC}^{LP}=\{\mathbb{V},t,L,\vec{\mathbf{W}}^{M*},\vec{\mathbf{W}}^{R*}\}$	An AOC system for link prediction
$\widetilde{\Gamma}(\mathcal{G}^0(\delta))=\{\widetilde{E}^1,\widetilde{E}^2,\cdots\}$	The predicted network sequence from an initial static network $\mathcal{G}^0(\delta)=(\mathbf{V},E^0)$
$\widetilde{N}_i^t=\{j:(i,j)\in\widetilde{E}^t\}$	The neighbor set of node i in the tth predicted network $\widetilde{E}^t$
$\widehat{N}_i^t=\{j:\exists x\geq 0,t-L+1\leq x\leq t,\ \ (i,j)\in\widetilde{E}^x\}$	The neighbor set of node i in the corresponding projection network $\widehat{P}^t(L)$
$\mathbb{N}_i^{t[\lambda]}=\{j:dis_{ij}^t\leq\lambda\}$	The set of reachable nodes associated with node i
$\widetilde{\omega}_i^t=\arg_{j\in\widehat{N}_i^t}\max\theta_{ij}^t$	The most likely disappearing neighbor of node i at time stamp $t+1$
$\widetilde{\varpi}_i^t=\arg_{j\in(\widetilde{N}_i^{t[\lambda]}-\widehat{N}_i^t)}\max\phi_{ij}^t$	The most likely emerging link ended by node i at time stamp $t+1$
$\widetilde{\mathfrak{N}}^{t[M]}=\{\langle i,j\rangle:\max_i\max_j\{\theta_{ij}^t\}=\max_j\max_i\{\theta_{ij}^t\}\}$	The predicted missing link set at time stamp t
$\widetilde{\mathfrak{N}}^{t[R]}=\{\langle i,j\rangle:\max_i\max_j\{\phi_{ij}^t\}=\max_j\max_i\{\phi_{ij}^t\}\}$	The predicted reconnected link set at time stamp t

similarly, let ε_{ij}^t denote the expected probability for any arbitrary node pair $(i,j)\notin E^t$ to be connected in period $t+1$, which can be estimated by

$$\varepsilon_{ij}^t\approx\frac{\sum_{i=1}^n\sum_{j=1}^n\frac{1}{(n-1-k_i^t)(n-1-k_j^t)}}{n^2}=\left(\frac{1}{n}\sum_{i=1}^n\frac{1}{n-1-k_i^t}\right)^2$$

One can also simplifies $\frac{1}{n}\sum_{i=1}^n\frac{1}{n-1-k_i^t}$ as

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^n\frac{1}{n-1-k_i^t}&=\frac{1}{n}\int_1^n\frac{np^t(k)}{n-1-k}dk=\int_1^n\frac{a^t-1}{(n-1-k)k^{a^t}}dk\\&=(a^t-1)\int_1^n\left(\sum_{l=1}^n\frac{1}{(n-1)^lk^{a^t-l+1}}\right)dk\\&=(a^t-1)\sum_{l-1}^n\left(\frac{1}{(n-1)^l}\frac{1}{t-a^t}(n^{t-a^t}-1)\right)\\&\approx(a^t-1)\frac{1}{n^{a^t}}\sum_{l=1}^n\frac{1}{l-a^t}\\&\approx\frac{(a^t-1)\ln n}{n^{a^t}}\end{aligned}$$

Thus, ε_{ij}^t can be simplified as $(\frac{(a^t-1)\ln n}{n^{a^t}})^2$, the proof completes. □

Corollary 2. *Based on Lemmas 1 and 2, the number of links in the evolutional networks would be more likely to reduce continuously.*

Proof. Without loss of generality, the ratio of reconnecting and missing links during period t can be measured by

$$\frac{\left(\frac{(a^t-1)\ln n}{n^{a^t}}\right)^2\left(\frac{1}{2}n(n-1)-m^t\right)}{\left(\frac{a^t-1}{a^t}\right)^2m^t}\leq\frac{\left(a^t\ln n\right)^2}{2m^tn^{2a^t}-1}=\frac{\left(a^t\ln n\right)^2}{\langle k\rangle^tn^{2a^t-1}},$$

where $\langle k\rangle^t$ denotes the average degree of network during period t; in most cases, the number of reconnected links will be less than that of missing links during the evolution, the proof completes. □

5. AOC system for link prediction

This section proposes an AOC system for link prediction (in short as AOCLP henceforth). We first detail the environment of our AOC system and then present algorithmic details including the complexity analysis. Important terms and notations associated with AOCLP are listed in Table 1 for easy reference.

5.1. Environment of AOC system

The proposed AOCLP, denoted by $\mathbb{AOC}^{LP}$, is formally defined as $\mathbb{AOC}^{LP}=\{\mathbb{V},t,L,\vec{\mathbf{W}}^{M*},\vec{\mathbf{W}}^{R*}\}$, where $\mathbb{V}=\{v_1,v_2,\cdots,v_n\}$ is the set of n “intelligent” agents, each characterizes a node i in the initial static network $\mathcal{G}^0(\delta)=(\mathbf{V},E^0)$, t is the system time,

L is the user-specified time window length, $\vec{\mathbf{W}}^{M*}$ and $\vec{\mathbf{W}}^{R*}$ are the maximum partial log-likelihood estimators for MLP and RLP, respectively. The goal of $\mathbb{AOC}^{LP}$ is to generate a predicted network sequence, i.e., $\widetilde{\Gamma}(\mathcal{G}^0(\delta)) = \{\widetilde{E}^1, \widetilde{E}^2, \cdots\}$, such that for every $t \geq 1$, the predicted $\widetilde{E}^t$ could match some snapshot in the real ϵ-adjacent network sequence (e.g., $E^{t'} \in \Gamma^{\epsilon}(\mathcal{G}^0(\delta))$) as well as possible. In such a scenario, an agent $v_i \in \mathbb{AOC}^{LP}$ can be further characterized by a three-tuple $\langle \widetilde{N}_i^t, \widetilde{\omega}_i^t, \widetilde{\varpi}_i^t \rangle$, where

- $\widetilde{N}_i^t = \{j : (i, j) \in \widetilde{E}^t\}$ is the neighbor set of node i in the tth predicted network $\widetilde{E}^t$. Thus the neighbor set of node i in the corresponding projection network $\widehat{P}^t(L)$, can be defined as
$$\widehat{N}_i^t = \{j : \exists x \geq 0, t - L + 1 \leq x \leq t,\ \ (i, j) \in \widetilde{E}^x\}. \tag{19}$$
Based on the definition of $\widehat{N}_i^t$, the set of reachable nodes associated with node i is defined as
$$\mathbb{N}_i^{t[\lambda]} = \{j : dis_{ij}^t \leq \lambda\}, \tag{20}$$
where dis_{ij}^t indicates the topological distance between i and j in projection network $\widehat{P}^t(L)$.
- $\widetilde{\omega}_i^t$ indicates some neighbor of node i during period t, who is more likely to be disconnected from node i at the next period, such that:
$$\widetilde{\omega}_i^t = \arg_{j \in \widehat{N}_i^t} \max \theta_{ij}^t, \tag{21}$$
where θ_{ij}^t is the relative risk associated with node pair (i, j) being unconnected in period $t + 1$ (see Eq. (13)).
- $\widetilde{\varpi}_i^t$ denotes some unconnected object from node i during period t, who is more likely to be connected by node i in period $t + 1$, i.e.,
$$\widetilde{\varpi}_i^t = \arg_{j \in (\widetilde{\mathbb{N}}_i^{t[\lambda]} - \widehat{N}_i^t)} \max \phi_{ij}^t, \tag{22}$$
where ϕ_{ij}^t is the relative risk associated with node pair (i, j) being connected in period $t + 1$ (see Eq. (14)). In this paper, we assume that each node during every period t can only catch sight of those nodes within $\lambda = 2$ distance in the corresponding projection network $\widetilde{P}^t(L)$.

5.2. The evolutionary algorithm

Given an initial static network $\mathcal{G}^0(\delta) = (\mathbf{V}, E^0)$, the user-specified time window length L and maximum iteration number $\mathbb{P}$, and the corresponding maximum partial log-likelihood estimators $\vec{\mathbf{W}}^{M*}$ and $\vec{\mathbf{W}}^{R*}$, the network evolutionary algorithm (in short as NEA), is computed by a fast iterative process as shown in Algorithm 1.

Since every node is identical, the description of the algorithmic details will be presented from the viewpoint of a single agent. Lines 1–2 initialize $\mathbb{AOC}^{LP}$ by characterizing each v_i using $\langle N_i^0, \varnothing, \varnothing \rangle$, where N_i^0 indicates the neighbor set associated with node i according to the initial network $\mathcal{G}^0$. Lines 3–25 treat the current network, i.e., $\widetilde{\mathcal{G}}^t$, as a discrete-time dynamical system, and iteratively employ the network-transition function based on Eq. (18) to predict the network topology in the next period. During each iteration, Lines 5–11 calculate each node's most likely lost/new friends (e.g., $\widetilde{\omega}_i^t$ and $\widetilde{\varpi}_i^t$) in parallel; then in Lines 12–21, the system integrates all potential Nash equilibrium points in the corresponding payoff-matrix, i.e., $\forall \langle i, \widetilde{\omega}_i^t \rangle$ and/or $\langle i, \widetilde{\varpi}_i^t \rangle$, to further uncover the predicted PNE sets w.r.t. MLP and RLP, respectively; finally in Line 22, the future link set is updated by $\widetilde{E}^t - \widetilde{\mathfrak{N}}^{t[M]} + \widetilde{\mathfrak{N}}^{t[R]}$. The above iterative process will stop when the system time reaches the given maximum iteration number $\mathbb{P}$, thus we get an evolutionary network sequence $(\widetilde{\Gamma}(\mathcal{G}^0))$, which starts from the given initial network $\mathcal{G}^0$.

5.3. Complexity analysis

Let n, d and $\widehat{m}^t$ denote the number of nodes in the initial network $\mathcal{G}^0(\delta)$, the number of dimensions of the feature vector associated with each node pair, and the number of links in the tth snapshot of the projection network $\widehat{P}^t(L)$, respectively. We assume that the neighbors of each node are stored in the sorted lists, where each element denotes one neighbor ID of a given node. The average degree of the projection network $\widehat{P}^t(L)$ is then denoted by $\langle \widehat{k} \rangle^t = \frac{2\widehat{m}^t}{n}$. We remark that during each iteration, the main time cost is dominated by Line 6, i.e., calculating the $\vec{\mathbf{F}}_{ij}^t$ for every node pair $\{(i, j) : j \in \mathbb{N}_i^{t[2]}\}$ associated with every node i, which can be simplified as $O(nd(\langle \widehat{k} \rangle^t)^3)$. Therefore, the total time complexity of NEA is the sum of the cost of the $\mathbb{P}$ iterations: $O(\sum_{t=0}^{\mathbb{P}} nd(\langle \widehat{k} \rangle^t)^3)$. In our simulations, $\widehat{m}^t$ is comparable to, or even less than m (the number of edges in the initial network) during each iteration; the total computational complexity will shrink to $O(\mathbb{P}nd\langle k \rangle^3)$, where $\langle k \rangle = \frac{2m}{n}$ is the average degree of the initial network.

6. Experiment

In this section, we first present some experimental settings, including datasets, baselines and evaluation measures. Then we explore how different covariates affect our approach, and compare our approach with some baselines on different temporal networks.

Algorithm 1 : Network Evolutionary Algorithm (NEA).

```
Require: $\mathcal{G}^0(\delta) = (\mathbf{V}, E^0)$, $L$, $\mathbb{P}$, $\vec{\mathbf{W}}^{M*}$, $\vec{\mathbf{W}}^{R*}$;
Ensure: The evolutionary network sequence $\widetilde{\Gamma}(\mathcal{G}^0)$;
$t \leftarrow 0$;
$\forall i \in \mathbf{V}$, $v_i \leftarrow \langle N_i^0, \varnothing, \varnothing \rangle$;
repeat
    paralleled execution $\forall i \in \mathbf{V}$
        Update $\widehat{\mathrm{N}}_i^t$ and $\mathbb{N}_i^{t[2]}$ using Eqs. (19) (20);
        $\forall j \in \mathbb{N}_i^{t[2]}$, update $\vec{\mathbf{F}}_{ij}^t$;
        $\forall j \in \widetilde{\mathrm{N}}_i^t$, $\theta_{ij}^t \leftarrow (\vec{\mathbf{W}}^{M*})^{\mathrm{T}} \vec{\mathbf{F}}_{ij}^t$ (Eq. (13));
        $\forall j \in \mathbb{N}_i^{t[2]} - \widetilde{\mathrm{N}}_i^t$, $\phi_{ij}^t \leftarrow (\vec{\mathbf{W}}^{R*})^{\mathrm{T}} \vec{\mathbf{F}}_{ij}^t$ (Eq. (14));
        $\widetilde{\omega}_i^t \leftarrow \arg_{j \in \widetilde{N}_i^t} \max \theta_{ij}^t$ (Eq. (21));
        $\widetilde{\varpi}_i^t \leftarrow \arg_{j \in (\mathbb{N}_i^{t[2]} - \widetilde{N}_i^t)} \max \phi_{ij}^t$ (Eq. (22));
        $\widetilde{\mathfrak{N}}^{t[M]}.insert(\langle i, \omega_i^t \rangle)$, $\widetilde{\mathfrak{N}}^{t[R]}.insert(\langle i, \varpi_i^t \rangle)$;
    for each $\langle i, \widetilde{\omega}_i^t \rangle \in \widetilde{\mathfrak{N}}^{t[M]}$ do
        if $\nexists \langle j, \widetilde{\omega}_j^t \rangle \in \widetilde{\mathfrak{N}}^{t[M]}$, $\langle i, \widetilde{\omega}_i^t \rangle = \langle \widetilde{\omega}_j^t, j \rangle$ then
            $\widetilde{\mathfrak{N}}^{t[M]}.delete(\langle i, \widetilde{\omega}_i^t \rangle)$;
        end if
    end for
    for each $\langle i, \widetilde{\varpi}_i^t \rangle \in \widetilde{\mathfrak{N}}^{t[R]}$ do
        if $\nexists \langle j, \widetilde{\varpi}_j^t \rangle \in \widetilde{\mathfrak{N}}^{t[R]}$, $\langle i, \widetilde{\varpi}_i^t \rangle = \langle \widetilde{\varpi}_j^t, j \rangle$ then
            $\widetilde{\mathfrak{N}}^{t[R]}.delete(\langle i, \widetilde{\varpi}_i^t \rangle)$;
        end if
    end for
    $\widetilde{E}^{t+1} \leftarrow \widetilde{E}^t - \widetilde{\mathfrak{N}}^{t[M]} + \widetilde{\mathfrak{N}}^{t[R]}$;
    $\widetilde{\Gamma}.insert(\widetilde{E}^{t+1})$;
    $t \leftarrow t + 1$;
until $t \leq \mathbb{P}$.
return $\widetilde{\widetilde{\Gamma}}(\mathcal{G}^0)$;
```

6.1. Experimental settings

6.1.1. Datasets

In this paper, we manually generate a set of ϵ-adjacent network sequences, which are derived from four temporal datasets on Stanford Network Analysis Platform (SNAP).[1] Specifically, `Math`, `Ask`, `Super` and `Stack` are four temporal networks of interactions on the stack exchange web sites: Math Overflow, Ask Ubuntu, Super User and Stack Overflow, respectively. For each dataset, there are three different types of interactions represented by a temporal link $\langle i, j, \tau \rangle$: 1) user i answered user j's question at time τ; 2) user i commented on user j's question at time τ; and 3) user i commented on user j's answer at time τ. The summary statistics of the raw temporal datasets described above are shown in Table 2, where the time resolution of the temporal links in all datasets is one second. Here we do not distinguish the direction on each temporal link, thus one specific $\langle i, j, \tau \rangle$ tuple can induce an undirected and unweighed link.

Here we present the structural analysis of the initial δ-networks derived from the four temporal datasets in Table 2 with $\delta = 365$ days. Specifically, with each raw temporal dataset, we first constructed an undirected and unweighed network using the data tuples in the first one year; and then extracted the maximal connected subgraph as the corresponding initial δ-network. The detailed statistical characteristics associated with the four induced δ-networks are shown in Table 3 and Fig. 4, from which we can obtain a few empirical observations: 1) all these networks possess the "small world" property [37] (i.e., small diameter and average path length) and 2) the "scale-free" property [3] (i.e., power law degree distribution); moreover, 3) they all have a high clustering coefficient [31].

Given the four induced δ-networks in Table 3, we further generate the corresponding ϵ-adjacent network sequences with $\epsilon = 2\%$. We choose 2% for the duration sliding rate as $\delta \cdot \epsilon \approx 7$ days is close to the average maximum time window for a user to take part in the three types of interactions in most of our datasets. Fig. 5 shows the movements of the numbers of static links in the three link sets (i.e., E^t, M^t and R^t) along the corresponding ϵ-adjacent network sequences, where $\epsilon = 2\%$. Here, the x-axis represents the period index, and y-axis indicates the number of static links. As can be seen in Fig. 5, all the link number curves have the very similar downward trend, that is, they drop nearly continuously regardless of the dataset

[1] http://snap.stanford.edu/data/index.html.

Table 2
Summary statistics of raw temporal datasets.

Datasets	# nodes	# temporal links	time span (days)
Math	24,818	506,550	2350
Ask	159,316	964,437	2613
Super	194,085	1,443,339	2773
Stack	2,601,977	63,497,050	2774

Table 3
Statistical characteristics associated with the four induced δ-networks ($\delta = 365$ days): n - the number of nodes; m - the number of static links; $\langle k \rangle = \frac{2m}{n}$ - the average degree of the network; c - the clustering coefficient of the network; d - the diameter of the network; $\langle dis_{ij} \rangle$ - the average path length of the network.

δ-networks	n	m	$\langle k \rangle$	c	d	$\langle dis_{ij} \rangle$
Math	4199	21,273	9.981	0.173	9	3.454
Ask	7084	23,520	6.415	0.097	10	3.825
Super	13,733	60,454	8.574	0.054	11	3.789
Stack	46,232	563,699	24.386	0.044	7	2.916

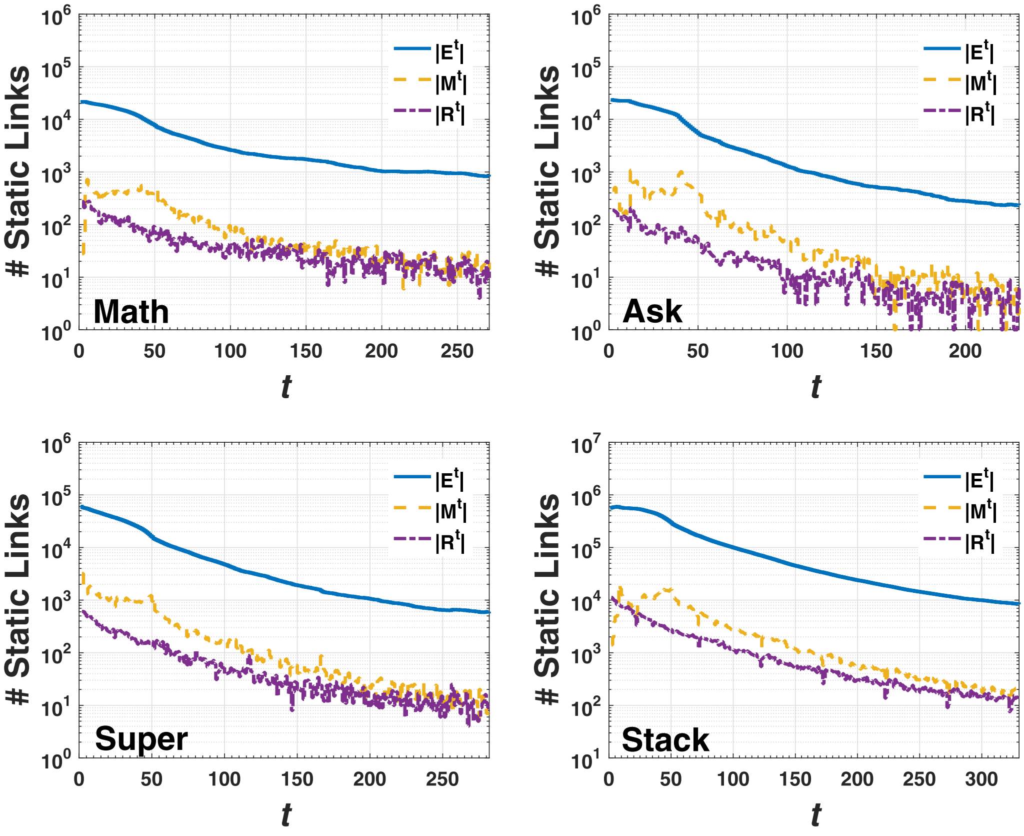

Fig. 5. The movements of the numbers of static links in the three link sets (i.e., E^t, M^t and R^t) along the corresponding ϵ-adjacent network sequences, where $\epsilon = 2\%$.

and the link type. For every dataset and every period $t \geq 1$, $|E^t|$ is about two orders of magnitude larger than $|M^{t+1}|$ and three orders of magnitude larger than $|R^{t+1}|$, indicating that there are very small proportions of the links disappearing from (nearly 1%), or appearing in (nearly 0.1%) the current/next snapshot of dynamic network. More interestingly, when $t \leq 100$, $|M^t| > |R^t|$, implying that more links are disappearing from the previous network; and this asymmetrical trend will be significantly improved when t exceeds 150 in most datasets except Stack.

In follow-up experiments, we will apply the network evolutionary algorithm as introduced in Algorithm 1 on the four induced δ-networks in Table 3. The corresponding ϵ-adjacent network sequences in Fig. 5 will be viewed as the baseline information, which can be used to evaluate the performance of our proposed approach.

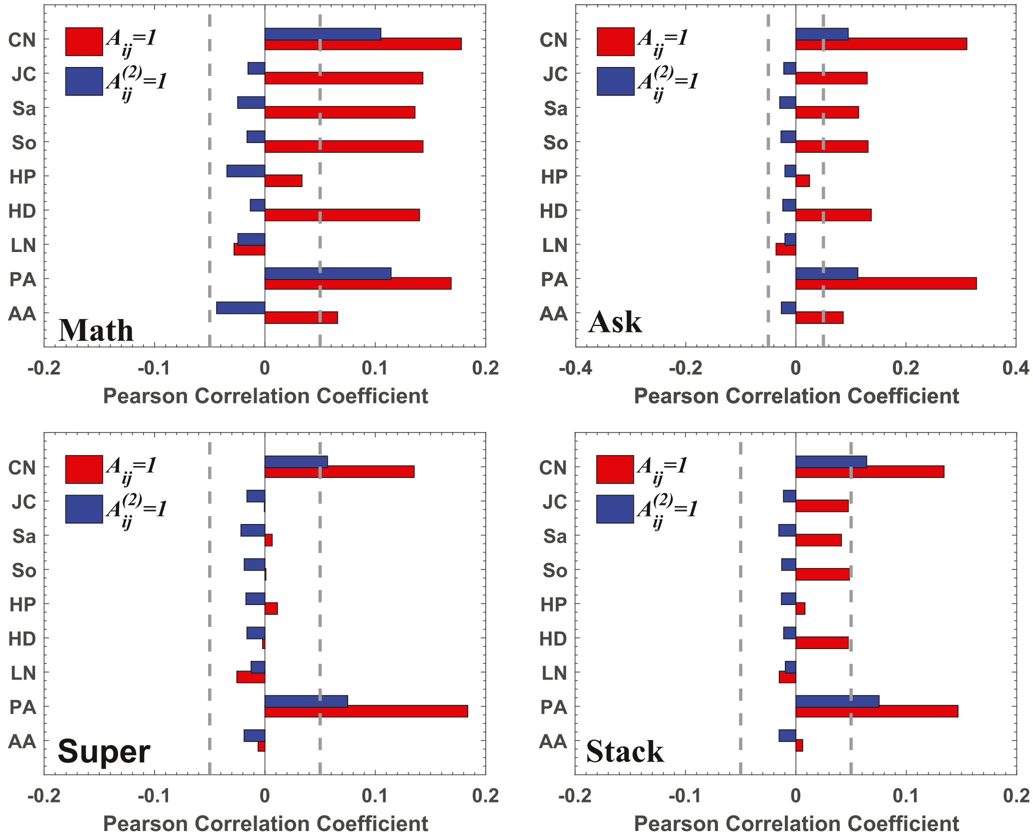

Fig. 6. Pearson correlation coefficients for proximity index-ζ_{ij} in the four δ-networks in Table 2, which exhibit the link appearance w.r.t. $(i,j)|_{A_{ij}=1}$ (red bars) and $(i,j)|_{A^{(2)}_{ij}=1}$ (blue bars) in the corresponding ϵ-adjacent networks ($\epsilon=2\%$). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

6.1.2. Baseline tools

The nine proximity functions listed in Section 3.2 capture the shared characteristics of two nodes. For each initial δ-network (denoted by E^0) as shown in Table 2, let's denote the corresponding ϵ-adjacent network as E^1, where ϵ is specified to be 2%. Let $\mathcal{A}=[A_{ij}]\in\mathbb{R}^{n\times n}$ and $\mathcal{A}^{(2)}=\mathcal{A}\times\mathcal{A}=[A^{(2)}_{ij}]\in\mathbb{R}^{n\times n}$ denote the one-step and two-step adjacency matrices respectively. We associate every node pair $(i,j)|_{A_{ij}=1}$ with a two-tuple $\langle\vec{\mathbf{F}}^M_{ij},\zeta_{ij}\rangle$, where $\vec{\mathbf{F}}^M_{ij}\in\mathbb{R}^d$ represents the set of $d=9$ proximity indices of the node pair (i,j); and $\zeta_{ij}=+1$ if $(i,j)\in E^1$, and -1 otherwise. Similarly, for every node pair with path length two $(i,j)|_{A^{(2)}_{ij}=1}$, we can also attach it with such a two-tuple $\langle\vec{\mathbf{F}}^R_{ij},\zeta_{ij}\rangle$, where $\zeta_{ij}=\{+1,-1\}$ reflects the appearance of the corresponding link (i,j) in E^1. Fig. 6 presents the Pearson correlation coefficients (PCC) for proximity index-ζ_{ij} in the four δ-networks in Table 2, which exhibit the link appearance w.r.t. $(i,j)|_{A_{ij}=1}$ (red bars) and $(i,j)|_{A^{(2)}_{ij}=1}$ (blue bars) in the corresponding ϵ-adjacent networks ($\epsilon=1.0$). A higher positive/negative score means that a specific proximity index is more positively related to the probability of the link appearance/disappearance; while a near zero score, i.e., $[-0.05,0.05]$, signifies that a given proximity index is independent from the link appearance/disappearance.

As shown in Fig. 6, most PCC scores are located between -0.05 and 0.05, indicating that a majority of the proximity indices cannot separate the newly formed/discarded links. This lack of separation is one indication that a predictor which combines information from several proximity indices might improve the performance of the link prediction problem. Fig. 6 also reveals that the manner in which the predictors should be combined is not as straightforward as one might envision. This is a result of the large class imbalance between the number of potential node pairs for new links and the actual numbers of new links formed, which is a common occurrence in large-scale networks. To sum up, the choice of which proximity indices to include may largely depend on the raw temporal dataset, as well as the topological structure of the initial δ-network.

For the sake of comparison, we will compare our network evolutionary algorithm with a plethora of well-known supervised learning frameworks, which can effectively combine different proximity indices for the link prediction problem. Specifically, five different classification algorithms, such as Logistic Regression (LR), SVM-Light (SVM), C4.5, Multilayer Perceptron Neural Network (MPNN) and Naive Bayes (Bayes), are selected as baseline tools. With each coupled network pair, i.e., (E^{t-1},E^t), on the corresponding ϵ-adjacent network sequence (see Fig. 5), we incrementally form two kinds of labeled training datasets as follows: we label all the links in M^t as positive training instances. For other links in $E^{t-1} - M^t$, we randomly sample a set of instances, as many as positive ones, and label them as negative training instances. The features for each of these instances are constructed using the proximity functions in Section 3.2 on the corresponding projection network, i.e., $P^{t-1}(L)$. A classifier is then trained on this labeled training set for missing link prediction. Along the same line, one can form another kind of labeled training dataset for newly-formed link prediction. Finally, we iteratively employ the trained classification models and the network-transition function in Eq. (18), to predict a network evolution sequence starting from each of the initial δ-network as shown in Table 3.

6.1.3. Evaluation metrics

Given the extracted ϵ-adjacent network sequence $\Gamma^* = \{E^{*1}, E^{*2}, \cdots, E^{*\mathbb{T}}\}$, let $\Gamma = \{E^1, E^2, \cdots, E^{\mathbb{P}}\}$ denote the predicted network evolution sequence which contains $\mathbb{P}$ coupled sequential networks. The forecast accuracy can be measured by comparing the similarity of these two network sequences.

Inspired by the classic evaluation method in community detection field [5,7,10], the first metric is the **Average F1Score**, which is defined as the following form:

$$AvgF1 = \frac{\sum_t \max_\tau F1(E^t, E^{*\tau})}{2\mathbb{P}} + \frac{\sum_\tau \max_t F1(E^t, E^{*\tau})}{2\mathbb{T}}, \tag{23}$$

where $F1(E^t, E^{*\tau})$ denotes the F1 score between link sets E^t and $E^{*\tau}$. Thus, $AvgF1(\Gamma, \Gamma^*)$ is the sum of the average F1 score of the best-matching real network snapshot to each predicted one, and the average F1 score of the best-matching predicted network snapshot to each baseline one.

To measure the alignment between the predicted network evolution sequence and the baseline sequence, we further consider the modified version of the Kendall tau rank distance. Let $IndexSeq(\Gamma, \Gamma^*) = \{\tau_1, \tau_2, \cdots, \tau_{\mathbb{P}}\}$ denote the best-matching index sequence, where $\tau_t = \max_\tau F1(E^t, E^{*\tau})$ indicates the period index of the best-matching baseline network snapshot to the tth network snapshot on the predicted network evolution sequence Γ. The Kendall's Tau Coefficient (KTC) associated with $IndexSeq(\Gamma, \Gamma^*)$ is defined as

$$KTC = \frac{C - D}{\frac{1}{2}\mathbb{P}(\mathbb{P}-1)}, \tag{24}$$

where C is the number of concordant pairs (e.g., $(\tau_t, \tau_{t'})|_{t<t', \tau_t \le \tau_{t'}}$) and D is the number of discordant pairs (e.g., $(\tau_t, \tau_{t'})|_{t<t', \tau_t > \tau_{t'}}$). The range of KTC is between -1 and 1, with $KTC = 1$ indicating the perfect agreement, and $KTC = -1$ denoting the reversed situation.

6.2. Significance analysis of the selected covariates

In this paper, a specialized statistical software, SPSS, has been employed to evaluate $\vec{\mathbf{W}}^{M*}$ and $\vec{\mathbf{W}}^{R*}$ through carrying out the Cox PHM analysis on the corresponding survival datasets, i.e., Ω^M and Ω^R, derived from each temporal network respectively. In using the Cox PHM in SPSS, it does not have to take a specific form of distribution of the baseline hazard function, thus $h_0(t)$ in Eq. (2) can be ignored in SPSS. For each temporal dataset in Table 2, we generated two survival datasets, using SDGM-MLP (see Definition 10) and SDGM-RLP (see Definition 11), by setting $\delta = 365$ days, $\epsilon = 2\%$ and $T = 50$.

Here, we first focus on the survival analysis on Math. The first survival dataset (i.e., Ω^M) includes 32,742 subjects and the second one (i.e., Ω^R) covers 122,87 subjects. Both Ω^M and Ω^R are excluded the censored subjects, as we only care about the relative "risk" associated with the selected eighteen covariates. Covariates are assessed based on whether they have a significant effect on the defined "events", and this is done through the Statistical Hypothesis Test. The Hypothesis Test includes a null hypothesis and an alternative hypothesis. The *Sig* value, is a statistical tool for significance testing, and is adopted here in the Hypothesis Test. A predetermined significance level is set as 0.05, meaning when the *Sig* value is below 0.05, the null hypothesis is refused because the chance of the null hypothesis being true is too small. In contrast, when the *Sig* value is greater than 0.05, the null hypothesis should be accepted. The coefficient of the covariate is estimated by employing the Maximum Log-Likelihood Estimation method (see Eqs. (11), (12)).

Table 4 shows the results produced from the Cox PHM analysis on Ω^M derived from Math using SPSS, where SE is the standard error of $\mathbf{w}^M$; *Wald* is the value of the Chi-Square test statistics ($Wald = |\frac{\mathbf{w}^M}{SE(\mathbf{w}^M)}|$), which statistic evaluates whether the coefficient of a given variable $\mathbf{w}^M$ is statistically significantly different from 0; $\exp(\mathbf{w}^M)$ indicates the relative risk w.r.t. the unit increase of $\mathbf{w}^M$. The "backward Likelihood Procedure" option has been chosen in SPSS in order to delete the covariates which are not significantly correlated with the "event" of link missing. Five steps have been taken in SPSS for the coefficient estimation of $\vec{\mathbf{W}}^{M*}$. In **Step 1**, the significance of all covariates was analyzed simultaneously. As the *Sig* values of some covariates were found to be larger than the predetermined significance level 0.05, they were ignored in the process of "backward Likelihood Procedure" regression. We continued this process to **Step 5**, as the *Sig* values of all of the rest of the covariates were less than 0.05, their effects were assumed to be significant for MLP; moreover, the *Wald* values of all these covariates all exceeded 10. To further identify the most significant "dangerous" covariates, their $\exp(\mathbf{w}^M)$ values were compared. From the sixth column of Table 4, we found that **So, HP** and **CN** are the three positively relevant covariates w.r.t. the event of link missing. On the contrary, **Sa** and **AA** are negatively correlated with the link missing event. The maximum partial log-likelihood estimator for RLP is shown in Table 5, which is obtained after **Step 6**. As shown in

Table 4
The maximum partial log-likelihood estimator for MLP ($\vec{\mathbf{W}}^{M*}$) on `Math`, where $\delta = 365$ days, $\epsilon = 2\%$ and $T = 50$. The bold covariates show to be positively correlated with the link missing event, while the bold italic ones are negatively correlated.

Covariates	$\mathbf{w}^M$	*SE*	*Wald*	*Sig*	$\exp(\mathbf{w}^M)$
Step 5			718.147	0.000	
CN	**0.04**	**0.01**	**204.00**	**0.00**	**1.04**
Sa	***−7.56***	***0.74***	***104.22***	***0.00***	***0.01***
So	**5.58**	**0.72**	**60.85**	**0.00**	**265.04**
HP	**1.07**	**0.09**	**153.45**	**0.00**	**2.93**
AA	***−0.20***	***0.04***	***30.15***	***0.00***	***0.82***

Table 5
The maximum partial log-likelihood estimator for RLP ($\vec{\mathbf{W}}^{R*}$) on `Math`, where $\delta = 365$ days, $\epsilon = 2\%$ and $T = 50$. The bold covariates show to be positively correlated with the link reconnection event, while the bold italic ones are negatively correlated.

Covariates	$\mathbf{w}^R$	*SE*	*Wald*	*Sig*	$\exp(\mathbf{w}^M)$
Step 6			67.440	0.000	
JC	***−11.91***	***4.12***	***8.35***	***0.01***	***0.00***
Sa	***−2.64***	***0.58***	***20.61***	***0.00***	***0.07***
So	**9.56**	**2.56**	**14.00**	**0.00**	**14194.70**

Table 6
The maximum partial log-likelihood estimators for MLP and RLP on the four temporal datasets in Table 2, where $\delta = 365$ days, $\epsilon = 2\%$ and $T = 50$. The bold covariates show to be positively correlated with the link missing/reconnection event, the bold italic ones are negatively correlated, while the covariates with missing values are independent from the related events.

Covariates	`Math`		`Ask`		`Super`		`Stack`	
	$\mathbf{w}^M$	$\mathbf{w}^R$	$\mathbf{w}^M$	$\mathbf{w}^R$	$\mathbf{w}^M$	$\mathbf{w}^R$	$\mathbf{w}^M$	$\mathbf{w}^R$
CN	**0.04**	–	**0.03**	**0.01**	–	**0.01**	***−0.01***	***−0.01***
JC	–	***−11.91***	***−14.29***	***−73.30***	***−5.55***	**20.62**	***−24.26***	**13.88**
Sa	***−7.56***	***−2.64***	***−5.00***	***−2.01***	***−4.63***	–	***−5.38***	**1.62**
So	**5.58**	**9.56**	**9.88**	**42.20**	**6.38**	–	**16.60**	***−5.16***
HP	**1.07**	–	**0.37**	–	**0.19**	**0.30**	***−0.07***	–
HD	–	–	–	–	–	***−11.43***	***−0.83***	***−3.85***
LN	–	–	**6.23**	–	**2.15**	***−11.53***	**8.91**	***−2.94***
PA	–	–	–	–	–	–	–	–
AA	***−0.20***	–	**0.51**	***−0.19***	**0.57**	***−0.16***	**1.50**	***−0.15***

Table 5, all the reserved covariates were significantly correlated with the interesting event, i.e., the edge formation by the pair of two previously unconnected nodes. One can observe that **So** is the most positively relevant covariate for the event of link reconnection; while **JC** and **Sa** are the two negatively relevant ones.

One can take the similar analysis processing on the other three datasets, to learn the corresponding maximum partial log-likelihood estimators for MLP and RLP, respectively. Due to the page limit, we only present the final learned estimators in Table 6. Among all the benchmark datasets, **PA** shows to be independent from the link appearance event. This finding is incompatible with existing studies on the preferential attachment model, which assumes that the probability that a new link is connected to the node i is proportional to k_i. In this work, we only focus on the network dynamics around a set of known nodes (e.g., nodes in the initial δ-network), but ignore the link formation caused by those unknown external nodes. Under this new scenario, the traditional preferential attachment model [3], which is used to generate evolving scale-free networks, may not be applicable.

6.3. Evaluation of link prediction performance

First, we fix $\mathbb{T} = \mathbb{P} = 100$, and perform Algorithm 1 on the four δ-networks as shown in Table 3. Fig. 7 presents the 100×100 F1 matrices on these four networks, where the x-axis indicates the iterative round of our approach, and the y-axis represents the period number in the baseline network evolutionary sequence, thus each pixel, i.e., (x, y), represents the F1 score between link sets E^x and E^{*y}. From Fig. 7, one can observe that in the first half of the iterations, our approach can achieve relatively high accuracy, i.e., the predicted link sets can well match the baseline ones between periods 1 and 20; however, the prediction accuracy will decrease linearly as x increases to 50. Different from existing link prediction approaches, this paper focuses on predicting the network evolutionary sequence starting from a given δ-network. As in the process of prediction, the $x + 1$th link set is only based on the network topological information in period x; with this setting, the prediction error will be gradually enlarged. Another interesting observation from Fig. 7 is that the best-matching

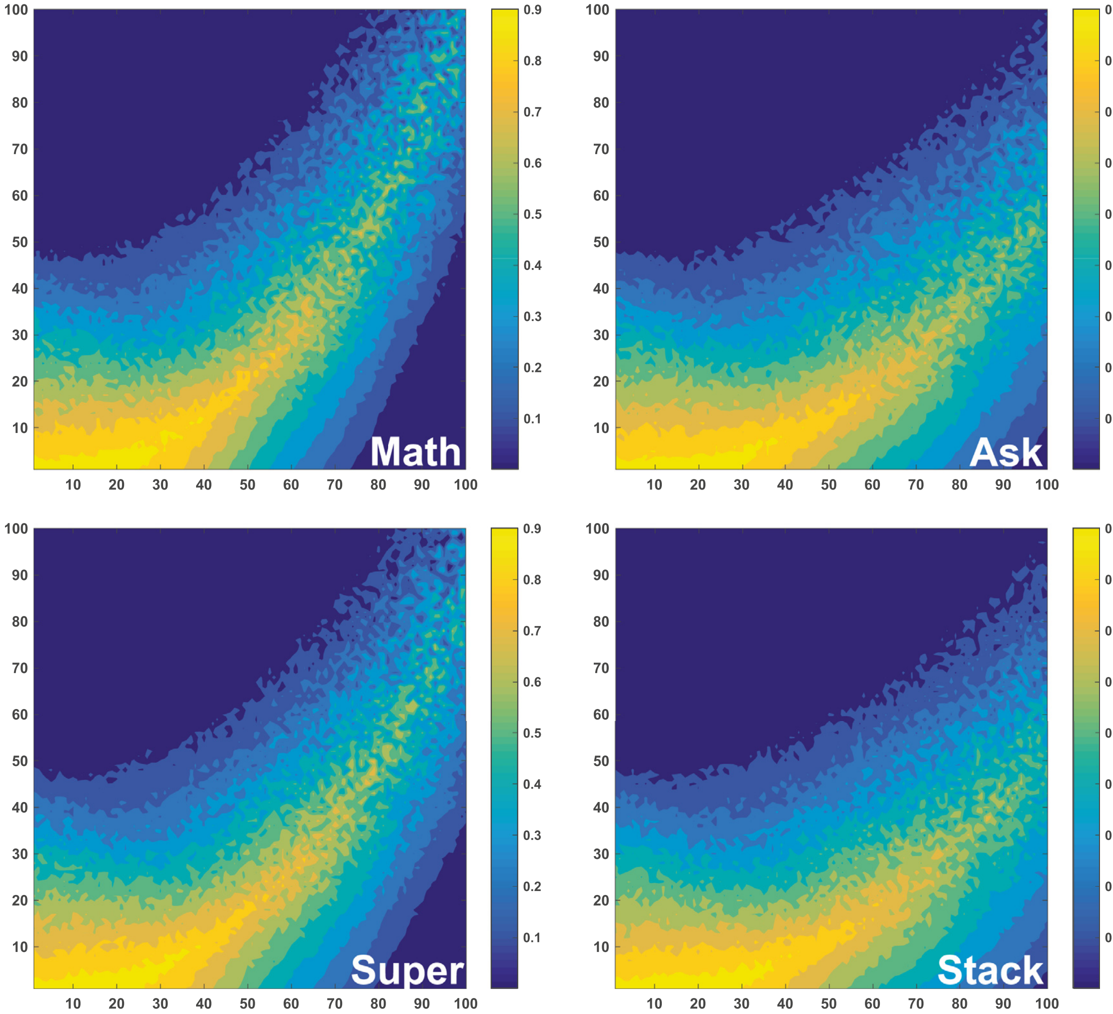

Fig. 7. The F1 matrices on the four networks, where the x-axis indicates the iterative round, and the y-axis represents the period number in the real network evolution sequence, thus each pixel, i.e., (x, y), represents the F1 score between link sets E^x and E^{*y}.

real network snapshot to each predicted one often shows to be overdue. This is because in our approach, the missing links and reconnected links are generated by the two-way choosing mechanism (i.e., finding the Nash equilibrium points in the corresponding payoff matrices), we remark that this setting might be too conservative; in other words, the predicted numbers of missing and reconnected links are much lower than the real ones.

Next, we quantitatively evaluate the quality of predicted network sequences, and give an overall comparison of all the approaches in terms of *AvgF*1 and *KTC*. As shown in Fig. 8, we compare our approach with some classic supervised learning methods, such as Logistic Regression (LR), SVM-Light (SVM), Decision Tree (C4.5), Multilayer Perceptron Neural Network (MPNN) and Naive Bayes (Bayes), on predicting the network evolution sequence. Clearly, in terms of *AvgF*1, our approach lightly outperforms all baseline methods. Most supervised learning methods are shown to be sensitive to the test networks, i.e., MPNN only performs well on `Stack`, but cannot accurately predict the network evolution in three other networks. As reported by Zhu et al. [43], supervised learning methods always suffer from the class imbalance problem, although existing attempts show that these approaches can achieve good performance on missing link prediction, they cannot thoroughly handle the problem of new link prediction. When we turn to *KTC*, most baseline approaches, except MPNN, perform poorly on all test networks, as they cannot control the trade-off between true and false positive rates. By introducing the game theory based two-way selection mechanism, our approach can accurately filter out most relevant links or node pairs. In spite of the sacrifice of recall score, our approach can achieve high precision score during each iteration. This relatively conservative prediction method is more fault-tolerant, thus performs better in terms of *KTC* as shown in Fig. 8.

We finally compare our approach with other baseline tools on the prediction time, the details are reported in Fig. 9, where the y-axis represents the total prediction time consumed by each approach in 100 iterations of continuous prediction. In this experiment, we consider the parallelized version of our approach by using four threads, which can keep the four

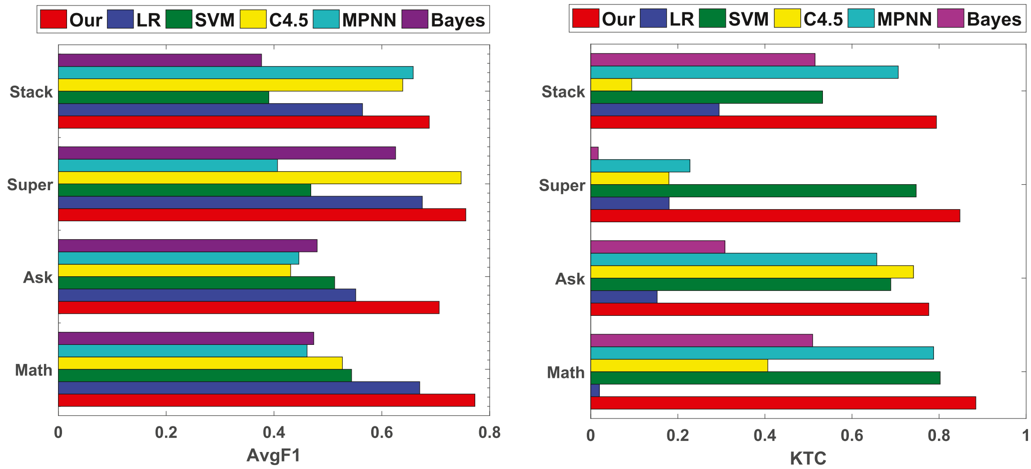

Fig. 8. Accuracy comparison on network evolution sequence prediction.

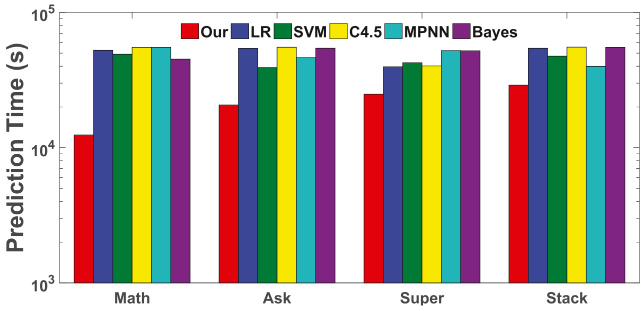

Fig. 9. The comparison of the total running time for 100 times of continuous prediction.

cores of the processor active. Since the differences between the fast and the slow algorithms are the orders of magnitude, multi-threading does not alter the conclusions. As shown in Fig. 9, our approach executes fast, and obtains at least a twice greater speed than other baseline tools. By employing the Cox PHM, our approach can select out most relevant features for link prediction; in other words, instead of considering all features, our method only needs to compute a small number of features associated with each node pair. On the other side, a clever game theory based two-way selection mechanism is employed in our link prediction framework; thus our approach can be effectively parallelized due to the fact that the computation of the link features associated with each node is independent.

7. Conclusion

Recently, temporal networks are ubiquitous and link prediction in temporal networks has attracted tremendous research interests, aiming to infer new interactions (links) among nodes that are likely to occur in the near future. Traditionally, in real-world temporal networks, link prediction may use structural properties of the network (i.e. nodes close proximity) or assumptions derived from observations (i.e. smooth evolution) to predict missing links in current networks and new or dissolution links in future networks, which is important in, for example, mining and analyzing the evolution of social networks. Existing traditional approaches need to unify these factors to strive for the spatial and temporal consistency in a dynamic network. To overcome this limitation, this paper proposes a novel framework to study the network evolution of the real-world temporal networks. To simplify the problem, we assume that nodes present across all periods are fixed; along this line, the network evolution prediction can be translated into the link prediction problem, i.e., given a network evolution sequence in a fixed period of time, how to predict the future missing and/or reconnected links. Unlike the classic link prediction problem, this paper focuses on predicting a network evolution sequence which can match the real one as closely as possible. Like existing supervised approaches, we choose a set of neighborhood based proximity functions to generate the feature vectors of node pairs in the observation networks; and further employ the Cox PHM to estimate the coefficients of the covariates. To reconcile the prediction effectiveness and the computational complexity, a novel game theory based two-way selection mechanism is proposed to compress the searching space from $O(n^2)$ to $O(n\langle k\rangle^2)$. Experiments on real-world temporal networks have demonstrated the effectiveness of our approach. To our best knowledge, this is the first work which combined the use of the survival analysis model and the game theory model to address the link prediction problem in the study of temporal networks. One direction of our future work is to integrate more external information; such as node attributes [6,8] and/or community structure [9], into the Cox PHM, so as to promote the prediction accuracy.

Conflict of interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled, “Link Prediction in Temporal Networks: Integrating Survival Analysis and Game Theory”.

Acknowledgments

This research was partially supported by National Natural Science Foundation of China under Grants 71871109, 71801123, 91646204 and 71871233, Natural Science Foundation of Jiangsu Province of China under Grant BK20150988, and Beijing Natural Science Foundation under Grant 9182015.

References

[1] N.M. Ahmed, L. Chen, An efficient algorithm for link prediction in temporal uncertain social networks, Inf. Sci. 331 (2016) 120–136.
[2] N.M. Ahmed, L. Chen, Y. Wang, B. Li, Y. Li, W. Liu, Sampling-based algorithm for link prediction in temporal networks, Inf. Sci. 374 (2016) 1–14.
[3] A.-L. Barabási, Scale-free networks: a decade and beyond, Science 325 (5939) (2009) 412–413.
[4] N. Barbieri, F. Bonchi, G. Manco, Who to follow and why: link prediction with explanations, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 1266–1275.
[5] Z. Bu, J. Cao, H.-J. Li, G. Gao, H. Tao, Gleam: a graph clustering framework based on potential game optimization for large-scale social networks, Knowl. Inf. Syst. 55 (3) (2018) 741–770.
[6] Z. Bu, G. Gao, H.-J. Li, J. Cao, Camas: a cluster-aware multiagent system for attributed graph clustering, Inf. Fusion 37 (2017) 10–21.
[7] Z. Bu, H.-J. Li, J. Cao, Z. Wang, G. Gao, Dynamic cluster formation game for attributed graph clustering, IEEE Trans. Cybern. 49 (1) (2019) 328–341.
[8] Z. Bu, H.-J. Li, C. Zhang, J. Cao, A. Li, Y. Shi, Graph k-means based on leader identification, dynamic game and opinion dynamics, IEEE Trans. Knowl. Data Eng. (2019), doi:10.1109/TKDE.2019.2903712.
[9] Z. Bu, Z. Wu, J. Cao, Y. Jiang, Local community mining on distributed and dynamic networks from a multiagent perspective, IEEE Trans. Cybern. 46 (4) (2016) 986–999.
[10] J. Cao, Z. Bu, Y. Wang, H. Yang, J. Jiang, H.-J. Li, Detecting prosumer-community groups in smart grids from the multiagent perspective, IEEE Trans. Syst. Man Cybern. (2019), doi:10.1109/TSMC.2019.2899366.
[11] Y. Dhote, N. Mishra, S. Sharma, Survey and analysis of temporal link prediction in online social networks, in: 2013 International Conference on Advances in Computing, Communications and Informatics, 2013, pp. 1178–1183.
[12] M. Gao, L. Chen, B. Li, Y. Li, W. Liu, Y.C. Xu, Projection-based link prediction in a bipartite network, Inf. Sci. 376 (2017) 158–171.
[13] S. Gao, L. Denoyer, P. Gallinari, Temporal link prediction by integrating content and structure information, in: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, ACM, 2011, pp. 1169–1174.
[14] R. Guns, R. Rousseau, Recommending research collaborations using link prediction and random forest classifiers, Scientometrics 101 (2) (2014) 1461–1473.
[15] S. Hanneke, W. Fu, E.P. Xing, et al., Discrete temporal models of social networks, Electron. J. Stat. 4 (2010) 585–605.
[16] P. Holme, J. Saramäki, Temporal networks, Phys. Rep. 519 (3) (2012) 97–125.
[17] H. Hu, C. Zhu, H. Ai, L. Zhang, J. Zhao, Q. Zhao, H. Liu, Lpi-etslp: lncrna–protein interaction prediction using eigenvalue transformation-based semi-supervised link prediction, Mol. Biosyst. 13 (9) (2017) 1781–1787.
[18] C. Lee, M. Pham, M.K. Jeong, D. Kim, D.K. Lin, W.A. Chavalitwongse, A network structural approach to the link prediction problem, INFORMS J. Comput. 27 (2) (2015) 249–267.
[19] Z.L. Li, X. Fang, O.R.L. Sheng, A survey of link recommendation for social networks: methods, theoretical foundations, and future research directions, ACM Trans. Manag. Inf. Syst. 9 (1) (2017) 1.
[20] D. Liben-Nowell, J. Kleinberg, The link-prediction problem for social networks, J. Am. Soc. Inf. Sci. Technol. 58 (7) (2007) 1019–1031.
[21] D.Y. Lin, L.-J. Wei, The robust inference for the cox proportional hazards model, J. Am. Stat. Assoc. 84 (408) (1989) 1074–1078.
[22] L. Lü, T. Zhou, Link prediction in complex networks: a survey, Physica A 390 (6) (2011) 1150–1170.
[23] V. Martínez, F. Berzal, J.-C. Cubero, A survey of link prediction in complex networks, ACM Comput. Surv. (CSUR) 49 (4) (2016) 69:1–69:33.
[24] R.G. Miller Jr, Survival analysis, 66, John Wiley & Sons, 2011.
[25] D. Monderer, L.S. Shapley, Potential games, Games Econ. Behav. 14 (1) (1996) 124–143.
[26] Y. Murase, J. Török, H.-H. Jo, K. Kaski, J. Kertész, Multilayer weighted social network model, Phys. Rev. E 90 (5) (2014) 052810.
[27] J. Nash, Non-cooperative games, Ann. Math. 54 (2) (1951) 286–295.
[28] J. O'Madadhain, J. Hutchins, P. Smyth, Prediction and ranking algorithms for event-based network data, ACM SIGKDD Explor. Newslett. 7 (2) (2005) 23–30.
[29] A. Paranjape, A.R. Benson, J. Leskovec, Motifs in temporal networks, in: Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, ACM, 2017, pp. 601–610.
[30] A. Popescul, L.H. Ungar, Statistical relational learning for link prediction, in: Proceedings of the Workshop on Learning Statistical Models from Relational Data, 2003.
[31] E. Ravasz, A.-L. Barabási, Hierarchical organization in complex networks, Phys. Rev. E 67 (2) (2003) 026112.
[32] R. Rossi, J. Neville, Time-evolving relational classification and ensemble methods, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2012, pp. 1–13.
[33] P. Sarkar, A.W. Moore, Dynamic social network analysis using latent space models, in: Advances in Neural Information Processing Systems, 2006, pp. 1145–1152.
[34] U. Sharan, J. Neville, Temporal-relational classifiers for prediction in evolving domains, in: Proceedings of the 8th IEEE International Conference on Data Mining, IEEE, 2008, pp. 540–549.
[35] T. Tylenda, R. Angelova, S. Bedathur, Towards time-aware link prediction in evolving social networks, in: Proceedings of the 3rd ACM Workshop on Social Network Mining and Analysis, ACM, 2009, p. 9.
[36] C. Wang, V. Satuluri, S. Parthasarathy, Local probabilistic models for link prediction, in: Proceedings of the 7th IEEE International Conference on Data Mining, 2007, pp. 322–331.
[37] D.J. Watts, S.H. Strogatz, Collective dynamics of small-world networks, Nature 393 (6684) (1998) 440.

[38] J. Xie, M. Chen, B.K. Szymanski, Labelrankt: incremental community detection in dynamic networks via label propagation, in: Proceedings of the Workshop on Dynamic Networks Management and Mining, ACM, 2013, pp. 25–32.
[39] Z. Yang, T. Hao, O. Dikmen, X. Chen, E. Oja, Clustering by nonnegative matrix factorization using graph random walk, in: Advances in Neural Information Processing Systems, 2012, pp. 1079–1087.
[40] J. Yin, Q. Ho, E.P. Xing, A scalable approach to probabilistic latent space inference of large-scale networks, in: Advances in Neural Information Processing Systems, 2013, pp. 422–430.
[41] K. Yu, W. Chu, S. Yu, V. Tresp, Z. Xu, Stochastic relational models for discriminative link prediction, in: Advances in Neural Information Processing Systems, 2007, pp. 1553–1560.
[42] T. Zhou, L. Lü, Y.-C. Zhang, Predicting missing links via local information, Eur. Phys. J. B 71 (4) (2009) 623–630.
[43] L. Zhu, D. Guo, J. Yin, G. Ver Steeg, A. Galstyan, Scalable temporal latent space inference for link prediction in dynamic social networks, IEEE Trans. Knowl. Data Eng. 28 (10) (2016) 2765–2777.