scientific reports

OPEN

Temporal link prediction via adjusted sigmoid function and 2-simplex structure

Ruizhi Zhang[1], Qiaozi Wang[2], Qiming Yang[1] & Wei Wei[1,3]✉

Temporal network link prediction is an important task in the field of network science, and has a wide range of applications in practical scenarios. Revealing the evolutionary mechanism of the network is essential for link prediction, and how to effectively utilize the historical information for temporal links and efficiently extract the high-order patterns of network structure remains a vital challenge. To address these issues, in this paper, we propose a novel temporal link prediction model with adjusted sigmoid function and 2-simplex structure (TLPSS). The adjusted sigmoid decay mode takes the active, decay and stable states of edges into account, which properly fits the life cycle of information. Moreover, the latent matrix sequence is introduced, which is composed of simplex high-order structure, to enhance the performance of link prediction method since it is highly feasible in sparse network. Combining the life cycle of information and simplex high-order structure, the overall performance of TLPSS is achieved by satisfying the consistency of temporal and structural information in dynamic networks. Experimental results on six real-world datasets demonstrate the effectiveness of TLPSS, and our proposed model improves the performance of link prediction by an average of 15% compared to other baseline methods.

With the rapid development of internet and network science, the explosive growth of various information volumes in different fields is accompanied by an urgent need for scientific research and industry to improve data processing capabilities. The complex network, which takes big data and complex associations among data as the research object, is the basic algorithmic framework for modeling data[1,2]. Link prediction in complex networks has been widely regarded as one of the most interesting problems in the information field[3].

The mission of link prediction is to predict the connection possibility of nodes that have not yet connected in the network, including the recovery of missing links and the formation of future links. The major difference of the aforementioned tasks is that the latter one mainly focuses on dynamic networks, which means links in those networks emerge at different time. For example, for protein networks as static data[4], due to the insufficiency of our empirical knowledge, prediction of two proteins' interaction can be thought as restoration of missing links. Static link prediction focuses on the completeness of the graph, while dynamic link prediction mainly predicts the formation of future links in order to simulate network evolution. It is well established that networks are highly dynamic objects with inherent dynamic properties[5]. Temporal link prediction aims to capture those properties in dynamic networks. It intends to extract the implicit driving force in the network and achieve the goal of network evolution analysis[6]. The most important application of it is in recommender systems[7], which have been widely used in many fields, such as e-commerce, social network and other scenarios[8,9].

Among all successful link prediction methods, similarity method is one of the most commonly used link prediction methodology. However, traditional similarity methods solely consider the current static state of networks, such as the topology structure, while ignoring the temporal dimensional evolution pattern of complex networks[10]. This type of method is not suitable for temporal networks, where edges are annotated with timestamps indicating the time they emerged. With the increasing demands of various applications in temporal networks, it is imperative to design a general temporal network link prediction method to effectively capture the temporal characteristics of network evolution. Several temporal link prediction methods have attempted to couple spatial and temporal information. LIST[11] characterized the network dynamics as a function of time, which used matrix factorization technique and integrated the spatial topology of network at each timestamp. By extracting target link features, SSF[12] used an exponential function to specify the influence of historical edges, and then combined the network structure to acquire the predictions. However, temporal link prediction methods based

[1]LMIB and School of Mathematical Sciences, Beihang University, Beijing 100191, China. [2]Longyuan (Beijing) Wind Power Engineering Technology Co., LTD., Beijing 100081, China. [3]Peng Cheng Laboratory, Shenzhen 518000, Guangdong, China. ✉email: weiw@buaa.edu.cn

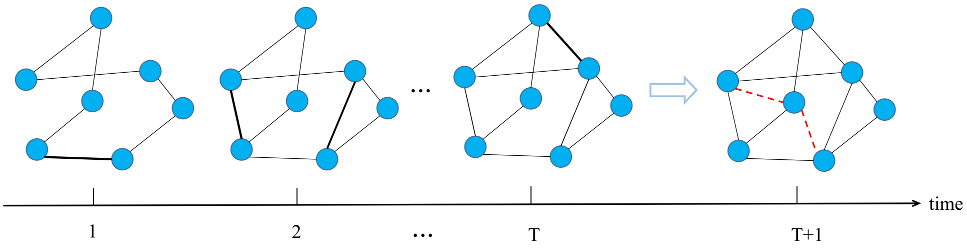

Figure 1. Schematic diagram of temporal link prediction.

Notation	Dscription
G^t	Graph snapshot at timestamp t
A^t	Adjacency matrix at timestamp t
B^t	Latent matrix at timestamp t
D^t	Degree vector at timestamp t
V	Node set
E^t	Edge set at timestamp t
$(x \sim h)$	Latent edge, x and h are nodes
p, q	Hyperparameters in adjusted sigmoid function

Table 1. Main notations.

on exponential decay ignore the life cycle of information that newly added edges in the network will remain active for a certain period of time, after which the link information decays to a stable state. Besides, many real-world networks are sparse and a majority number of existing structure-based similarity methods are common neighbor related[13–17], which might cause lower performance of these methods. In addition, due to the irregular connection characteristics of the network, each node has its unique local topology. Therefore, when considering the local structure of the target link, the high-order structure of the two endpoints should also be reflected.

To address these issues, we first utilize the characteristics of the sigmoid function to systematically modify the demerits of the exponential function[18]. We propose the adjusted sigmoid function (*ASF*) to quantify temporal information based on the simplified life cycle of information. Then, owing to the powerful mathematical representation of simplex in algebraic topology, we come up with hidden neighbor set and latent matrix sequence, which solve the dilemma that some node pairs do not have common neighbors due to network sparsity. Finally, considering the endpoints asymmetry with simplex structure, it fully represents the surrounding topology information around the target links. Combining them to achieve the consistency of temporal and structural information, thus, the link prediction model TLPSS is proposed for general dynamic networks. The main contributions of this paper are as follows:

- Based on the active, decay and stable states of information, we proposed a new time decay mode *ASF*, which adequately considers the decay time and rate for different network information.
- We define the latent matrix sequence composed of simplex high-order structure to reveal the spatial topology of the network. The richer high-order topological information in latent edges alleviates the problem that traditional similarity methods are affected by lack of common neighbors due to the sparsity of the network.
- Coupling temporal and structural information, we introduce a temporal link prediction metric TLPSS induced by the hidden neighbor asymmetry, and it is consistently feasible for various dynamic networks.
- We evaluate the TLPSS model on six real-world datasets and show that it outperforms other baseline models, which demonstrates the effectiveness of our proposed method.

Problem description

A dynamic network is defined as a graph $G_t = (V^t, E^t)$, where V^t is the node set, E^t is the set of links. A temporal link is denoted by $e^t(u, v)$, which means that the node u and v are connected at time $t \in \{1, 2, ..., T\}$. Since this paper focuses on the link prediction, we only consider the change of edge connection with time, and fix the node set at different time as V. Note that node pairs are allowed to have multiple edges generated at different timestamps, and only undirected networks are concerned in this paper.

For temporal link prediction, a temporal network G_t can be divided into a series of snapshots $G_t = \{G^1, G^2 ..., G^T\}$ at discrete time frames $\{1, ..., T\}$. For $t, s \in T$, $t < s$, G^t can be regarded as the historical information of G^s, and they are strongly correlated and involved with the same evolution mechanism. When a set of network snapshots are given within the time period $[1, T]$, the temporal link prediction method aims to study a function $f(\cdot)$ to predict a group of edge set $E^{T+1} = \{e^t(u, v)|u, v \in V, t = T + 1\}$ created at time $T + 1$. The problem is illustrated in Fig. 1. Main notations in this paper are introduced in Table 1 for future reference.

Literature review

A large number of static link prediction methods have been proposed, and these methods can be divided into three categories[19]. The first category is the method based on probability statistics[20]. The basic idea of these methods is to build a parametric probabilistic model and use optimization strategies such as maximum likelihood estimation to find the optimal parameters. This type of model usually acquires satisfying results by assuming that the network has known structures or obeys specific distributions. But usually the great computational cost makes it not suitable for large-scale network, and representative works are like[21–23]. The second category is machine learning-based methods. The link prediction problem in the network can be regarded as a classification problem in machine learning[24], and the related methods work on massive training data to achieve high prediction accuracy in large-scale networks though explainable features are difficult to be extracted. Furthermore, inspired by the superiority of deep learning and graph representation learning in capturing node feature representations[25], the link prediction task can be transformed into computing distances between nodes to reveal the underlying correlation. The advantage of this type of method is that with the iterative update of representation learning algorithms, such as deepwalk[26], node2vec[27] and their derivatives, the link prediction accuracy can be gradually improved, but the prediction mechanism is difficult to explain in an explicit way. The third category is the similarity-based method[10], which is based on the assumption that the connection probability of nodes is positively correlated with the similarity between them[28,29]. Such methods assign a score to each pair of nodes by defining a similarity function, and higher scored node pairs will have more potential to be linked together.

Recently, more complicated metrics based on temporal and structural information have been proposed for link prediction. Yu et al.[11] proposed a link prediction model LIST with spatio-temporal consistency rule, which described the dynamic characteristics of the network as a function of time, and integrated the spatial topology of the network at each time point and the temporal characteristics of network evolution. Chen et al.[30] proposed the temporal link prediction model STEP, which integrated structural and temporal information, and transformed the link prediction problem into a regularized optimization problem by constructing a sequence of high-order matrices to capture the implicit relationship of node pairs. Li et al.[12] proposed a structure subgraph feature model SSF that is based on link feature extraction. This method effectively represented the topological features around the target link and normalized the influence of multiple links and different timestamps in structural subgraphs.

Complex networks have become the dominant paradigm for dynamic modeling of interacting systems. However, networks are inherently limited to describe the interactions of node pairs, while real-world complex systems are often characterized by high-order structures. Furthermore, interaction behaviors based on structure take place among more than two nodes at once[31]. Empirical studies reveal that high-order interactions are ubiquitous in many complex systems, and such community behaviors play key roles in physiological, biological, and neurological field. In addition, high-order interactions can greatly affect the dynamics of networked systems, ranging from diffusion[32], synchronization[33], to social evolution processes[34], and may lead to the emergence of explosive transitions between node states. For a deeper understanding of the network pattern structure, we can model it via set systems from the perspective of algebraic topology. For example, high-order structures, such as hypergraphs and simplicial complex are better tools for characterizing many social, biological systems[35]. In addition to recognize the high-order structure in the network, it is important to measure the interaction information of the different structures. Applying simplex structure to complex networks due to its powerful mathematical representation and fully quantifying structure interaction information is the key to the performance of link prediction.

Methods

Time decay function.

The most crucial part of the temporal network link prediction task is to effectively process historical information. Based on the accumulation of historical time data, network evolution pays attention to the overall changes of the network, and performs the complex behavior of dynamic networks. Similarly, the purpose of link prediction is to understand how these characteristics will evolve over time. Link prediction makes use of temporal information to reveal the relationship between the current state of the network and its most recent states. The basic principle of dynamic link prediction is temporal smoothing[36], which assumes that the current state of a network should not change dramatically from its most recent history in general. Several researchers concern the exponential function as time decay function[11,12],

$$f(s,t) = e^{-\theta(t-s)}, \tag{1}$$

which means the remaining influence of a history link l with timestamp s at present time t, and $\theta \in (0,1)$ is a damping factor to control the speed of decay, and as a parameter, θ needs to be pre-learned.

Choosing the exponential function as the time decay function has improved some link prediction algorithms[11,12], and it can be regarded as one of the information decay modes. Besides, scholars have been discussing the society as an information- and knowledge-based society. By giving an insight into them, information resources can be clarified with life cycle model[37–39]. The life cycle phases consist of generation, institutionalization, maintenance, enhancement, and distribution. Inspired by this theory, we assume that the generation of new edges tends to remain active for a certain period of time, and then decay to a stable state. For example, the 2022 Grammys song of the year *Leave the door open*, its Billboard chart history follows the above hypothesis that it remained popular (top five) for 14 weeks, then it was gradually losing its position in the following 25 weeks, finally it fell off the chart in week number 40. Based on this assumption, we find that the sigmoid function in neural network[18] is accompanied with such properties. Sigmoid function has been widely used because of its good properties, for example, it is smooth and can be easily parameterized. Based on Occam's Razor principle, we obtain the adjusted sigmoid function (*ASF*) by parameterizing the sigmoid function, which satisfies the assumption, as the temporal information decay function. It can be divided into active state, decay state and stable state. The formal definition of *ASF* is as follows.

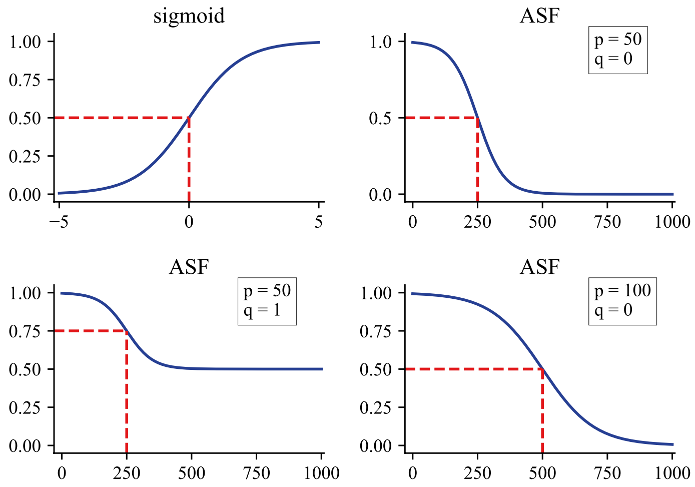

Figure 2. Sigmoid and ASF function. The upper left figure shows the original sigmoid function. The comparison of the upper right and lower left figures shows that the more information is remained with the larger parameter *q*. The comparison of the upper right and lower right figures shows that the active period of link information is determined by the parameter *p*.

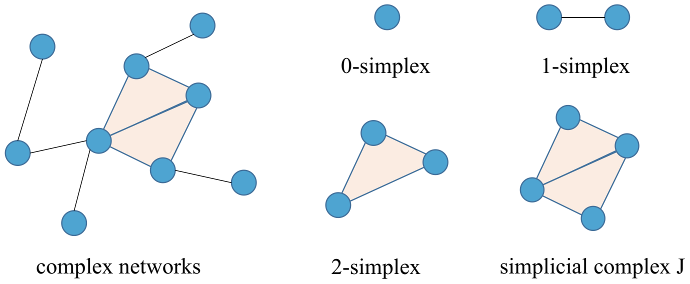

Figure 3. Examples of different types of simplex structures in networks.

$$ASF(x) = \frac{\frac{1}{1+exp\{x/p-a\}} + q}{q+1}, \tag{2}$$

in which the parameter p represents the active period of the information, and an increase in p means that the information is active for a long time. For parameter q, it controls the decay range of information, a larger q means that the lower bound of link information gets greater. As shown in Fig. 2, the influence of the parameter p is mainly reflected in the first stage of the *ASF* function. Compared to the upper right figure, the lower right one has a longer active time with the larger p. Besides, the role of the parameter q is reflected in the value range of the *ASF* function. Comparison with the upper right and lower left figures indicates that the more information is remained with the larger q in stable state. Unlike the former parameters, the role of parameter a is only to fix the position of ASF. In the experiment, we set $a = 5$. It is obvious that the lower bound of *ASF* is $q/(q+1)$, which means that the remaining temporal information of all links in the network is greater than this value. The sigmoid fuction and variation of *ASF* with parameters are illustrated in Fig. 2.

Simplex structure in link prediction.

The basic premise of network model is to represent the elements of system as nodes and to use links to capture the pairwise relationships. High-order interactions mean that there are more than two entities involved at once[31], which are ubiquitous in various scenes[40–42]. Capturing and characterizing high-order structures in networks is helpful for revealing network evolution mechanisms. Motivated by the significance of the triangular structures in network clustering and the theory of triadic closure in social networks[43], we employ this theory via the increasing of structural order. Similar to the definition in algebraic topology, a set of k nodes is called a $(k-1)$-simplex, and the set with all possible connections is called a k-clique in graph theory. Likewise, a simplicial complex is a finite set of simplices[31,35]. As shown in Fig. 3, 0-simplices are nodes, 1-simplices are edges, and 2-simplices are triangles. The simplicial complex J is composed of two 2-simplices.

Researchers apply high-order structure to link prediction to capture the topology information around the target link. For example, similarity metrics CAR[14] and CCLP[15] use the triangle structure, which gave some insights into the mechanism of high-order interaction. But such methods mainly focus on the quantity information of

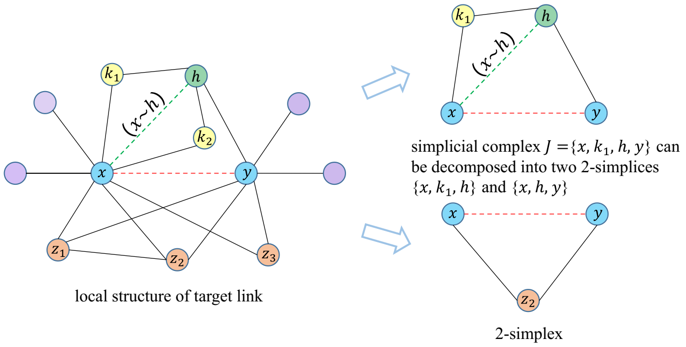
k_1
h
$(x\sim h)$
k_2
x
y
z_1
z_2
z_3
local structure of target link
simplicial complex $J=\{x,k_1,h,y\}$ can be decomposed into two 2-simplices $\{x,k_1,h\}$ and $\{x,h,y\}$
2-simplex

Figure 4. Schematic diagram of the topology around the target link. In this figure, the node pair x and y is to be predicted, $\{z_1, z_2, z_3\}$ are their common neighbors, and they form 2-simplices $\{x, z_i, y\}, i = 1, 2, 3$. h is the hidden neighbor of endpoint x, and $(x \sim h)$ is the latent edge. The simplicial complex J can be decomposed into two 2-simplices $\{x, k_1, h\}$ and $\{x, h, y\}$. Symmetrically, k_1 and k_2 are hidden neighbors of node y.

triangles, while ignoring the interaction information between different structures. Simplex has been used in the study of complex dynamical systems due to its powerful mathematical representation[35].

In this paper, we introduce the concept of latent edge, its detailed definition is given in the following subsection. Thus, the 2-simplices around the target link form a simplicial complex structure. We measure the interaction information of these two 2-simplices to capture the local topology structure around the target link.

Proposed algorithm. Since dynamic network G_t consists of a series of snapshots, adjacency matrix sequence can be expressed as $A_t = \{A^1, A^2..., A^T\}$, $A^t = \left[a_{i,j}^t\right]_{N\times N}$, $a_{i,j}^t \in [0,1]$, $N = |V|$ is the total number of nodes. Given time t, $a_{i,j}^t \neq 0$ means that node i and node j are connected, and the value is the quantification of corresponding time information by *ASF* fuction. The smaller the value is, the earlier the edge is generated, and $a_{i,j}^t = 0$ means that node i and node j are unconnected. Calculation process of $a_{i,j}^t$ is demonstrated as follows. We first normalize the timestamp information, so that the timestamp of the network starts from 1. Then, by dividing the time interval, we map timestamp information of each edge in the network into age (denoted by $age_{i,j}^t$) to show that how long it has existed, and a larger age corresponds to an earlier appearance. Finally, put the age ($age_{i,j}^t$) into the *ASF* function, which is a decreasing function, we get the weight $a_{i,j}^t = ASF(age_{i,j}^t)$ of every edge. A larger weight $a_{i,j}^t$ indicates that the edge appears later.

In general networks, the degree of a node is defined as the number of its neighbor nodes. In our study, the elements in the adjacency matrix are no longer just 0 or 1, so the degree of a node is no longer an integer but a continuous number. Adjacency matrix can be regarded as a weighted matrix, and the weighted adjacency matrix is different at each moment. Therefore, based on the adjacency matrix sequence, the node degree information should also vary over time. The formal definition of degree vector sequence (*DVS*) is as follows.

Definition 1 (Degree Vector Sequence) Given the adjacency matrix sequence, the degree information of nodes changes as the network evolves, and it can be calculated from the adjacency matrix at the corresponding snapshots. We can define *DVS* as $D_t = \{D^1, D^2, ..., D^T\}$, and each degree vector is obtained by the following calculation formula.

$$D^t = [w^t(v)]_{N\times 1}, w^t(v) = \sum_{z\in\Gamma(v)} A^t(v,z), \tag{3}$$

in which $\Gamma(v)$ is the set of neighbors of node $v \in V$.

The core of link prediction is correlation analysis, which reveals the intrinsic similarity between objects. A higher score of the link prediction metric indicates a higher probability of forming a link. Methods based on node centrality or common neighbors and their relevant variants indeed achieved good results[13,16,44]. For example, Resource Allocation index (RA)[17] considers that each node in the network has certain resources and distributes the resources equally to its neighbors. Besides, RA index shows good performance with low time complexity and high accuracy on some datasets. The formula of RA index is as follows.

$$RA(x,y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{|\Gamma(z)|}. \tag{4}$$

However, this method only considers the transmission of resources through common neighbor paths, while ignoring the potential resources transmitted through local paths between two endpoints. As shown in Fig. 4, for example, RA index only uses the 2-simplices $\{x, z_i, y\}, i = 1, 2, 3$, which ignores the importance of neighbor nodes of y that are not directly connected to x. However, theses nodes participate in forming the high-order structure $J = \{x, k_i, h, y\}, i = 1, 2$ around the target link. It is assumed that the resources of node are allocated to its neighbors according to the importance of the nodes. The role of common neighbors in information transmission is important. But we should also pay attention to those nodes that are only directly connected to one endpoint of the target link, such as node h. Combining the above analysis, we define hidden neighbor set (*HNS*) for endpoints that is crucial in information transmission.

Definition 2 (Hidden Neighbor Set) For each endpoint of target links, we define its hidden neighbor as the kind of node that is connected to one endpoint and the neighbor of the other endpoint. Given a node pair x and y, the *HNS* of endpoint x can be formulated as follows,

$$H_x = \{h | h \in \Gamma(y), h \notin \Gamma(x), \Gamma(x) \cap \Gamma(h) \neq \varnothing\}. \tag{5}$$

Vice versa, for endpoint y,

$$H_y = \{h | h \in \Gamma(x), h \notin \Gamma(y), \Gamma(y) \cap \Gamma(h) \neq \varnothing\}. \tag{6}$$

Based on the definition of *HNS*, we can divide the neighbors of an endpoint into three categories according to their topological significance. The first is the common neighbors, the second is the hidden neighbors, and the third is the rest nodes. The consideration of hidden neighbors makes the link prediction method take higher-order structure into account than traditional common neighbor based similarity methods.

After the definition of hidden neighbor, we assume there is a high probability that nodes will be connected to its hidden neighbors. For example, as shown in Fig. 4, the hidden neighbor h of the node x might be connected to x. Therefore, we call this edge in the network that is temporarily unconnected but carries target link information as a latent edge, obviously, it is composed of endpoint and hidden neighbor. Hidden neighbor and latent edge play an important role in improving link prediction performance because they participate in forming a simplicial complex structure around the target link. As shown in Fig. 4, simplicial complex structure J is composed of two 2-simplices $\{x, k_1, h\}$ and $\{x, h, y\}$. Besides, their intersection is the latent edge, which contains certain information of the endpoints. We give the formal definition of latent edge (*LE*) as follows.

Definition 3 (Latent Edge) Latent edge is a hypothetical edge that connects node and its hidden neighbor. It is denoted as $(x \sim h)$, in which x and h are node and its hidden neighbor respectively. The weight of latent edge will be explicitly given in the following definition of Latent Matrix Sequence.

Combining the latent edge, as shown in Fig. 4, we can decompose the simplicial complex structure J of the target link into two 2-simplices $\{x, k_1, h\}$ and $\{x, h, y\}$. *LE* help us reveal the spatial topology information of the target link. Several traditional methods like CN, RA only take paths of length 2 into consideration, which is merely the fundamental of local topological structure. In order to further extend the usage of topological information, we take the paths of length 3 into account and we use the idea of latent edges to quantify the corresponding contribution of similarity value. Latent edge helps us transform the path with length of 3 into two asymmetric paths with length of 2. It can be seen that the latent edges contain non-negligible topological information of the target node pair, but the quantification of its significance remains unsolved. Here we define the quantification strategy of such edges by the definition of latent matrix sequence (*LMS*) as follows.

Definition 4 (Latent Matrix Sequence) The connection state of network at each snapshot can be represented by the adjacency matrix sequence $A_t = \{A^1, A^2 ..., A^T\}$. We define Latent Matrix Sequence as $B_t = \{B^1, B^2 ..., B^T\}$, the elements in the B^t are weights of latent edges. Latent edges use simplicial complex structure to fully consider the information transmission between endpoints. The value of latent edges is calculated as follows.

$$B^t(i,j) = inf\{ASF\} \cdot scale\ factor, \tag{7}$$

$$inf\{ASF\} = \frac{q}{q+1}, \tag{8}$$

$$scale\ factor = \frac{1}{min\{d(i), d(j)\}} \cdot \sum_{z \in \Gamma(i) \cap \Gamma(j)} \frac{A^t(i,z) + A^t(z,j)}{(m(i,z) + m(z,j))}, \tag{9}$$

where $m(i, z)$ is the number of multiple edges between node i and node z created at different timestamps, and $d(i)$ is the degree of a node in the traditional sense. $A^t(i, z)$ is the weight of link between node i and node z at time t.

These operations hold that the weight of the latent edge is less than the weight of the existing edge in the network. We simply prove it as follows.

Proof The first item $inf\{ASF\}$ in Eq. (7) is the lower bound of the *ASF* function, and it quantifies the time information of the connected edges in the network. It is clear that the numerator of the second term in Eq. (9) is less than the denominator. Since $|\Gamma(i) \cap \Gamma(j)| \leqslant min\{d(i), d(j)\}$, we obtain $scale\ factor \leqslant 1$. The product of the two terms in Eq. (7) ensures that the weight of the latent edge is less than the $inf\{ASF\}$. Thus, the weight of the latent edge is less than the weight of the existing edge in the network.

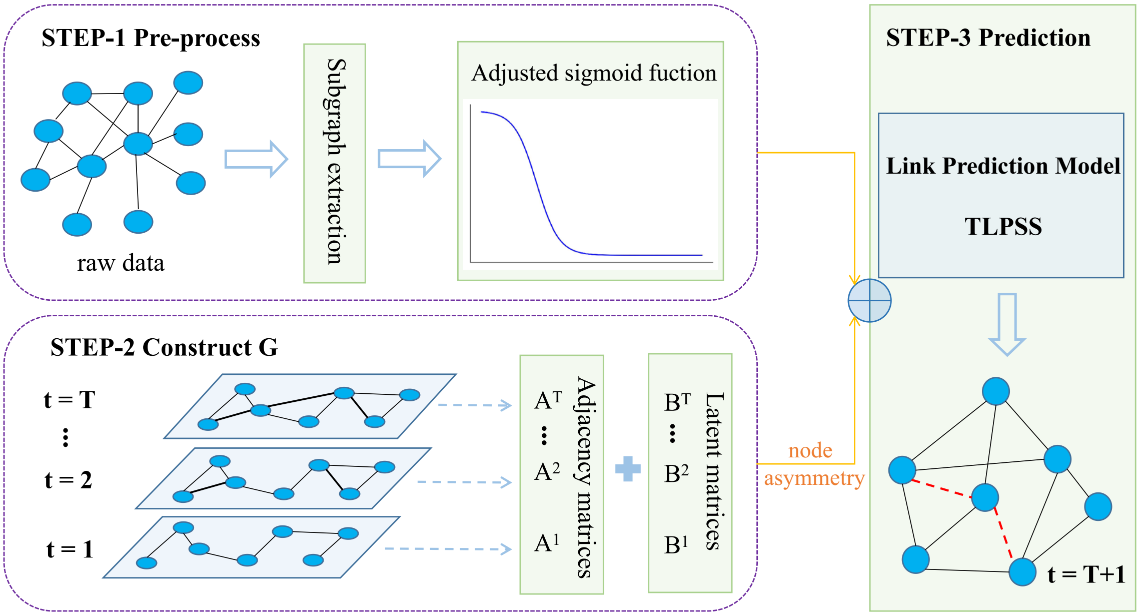

Figure 5. Diagram of the proposed model TLPSS. This model contains pre-process, construct graph and prediction steps. In first step, the data is processed and decayed by *ASF*. Then, according to the network snapshots, we obtain the adjacency matrix sequence and latent matrix sequence. Finally, coupling the temporal and structural information, the temporal link prediction method TLPSS is proposed.

The *LMS* further characterizes the topology information around the target link by using simplicial complex structures. Besides, considering the complete difference between two endpoints' hidden neighbors, it is important to introduce endpoints asymmetric topology information into the mechanism of link prediction. After the above analysis, based on the network history information from time 1 to T, we can predict the generation of new links at time $T+1$. By mixing the 2-simplices information in adjacency matrix and latent matrix of endpoints, we obtain endpoints similarity scores respectively. For endpoint x,

$$score(x \to y) = \sum_{z_i \in \Gamma(x) \cap \Gamma(y)} \frac{A^T(x, z_i)}{w^T(z_i)} + \sum_{z_i \in H_x} \frac{B^T(x, z_i)}{w^T(z_i)}. \tag{10}$$

Similarly, for endpoints y,

$$score(y \to x) = \sum_{z_i \in \Gamma(x) \cap \Gamma(y)} \frac{A^T(y, z_i)}{w^T(z_i)} + \sum_{z_i \in H_y} \frac{B^T(y, z_i)}{w^T(z_i)}. \tag{11}$$

Finally, we obtain the temporal link prediction method TLPSS that integrates the features of 2-simplex topological structures, endpoints asymmetry and the *ASF* time decay paradigm,

$$TLPSS(x, y) = \frac{1}{2}(score(x \to y) + score(y \to x)). \tag{12}$$

Based on the above analysis, the proposed algorithm can be divided into three steps, and the schematic diagram of TLPSS model is shown in Fig. 5. Firstly, it is necessary to pre-process the data, because the data we get is always noisy. Specifically, for every dataset, we remove its edges that don't possess timestamp attribute. After which we sort the remaining edges according to their timestamp information for the convenience in splitting train set and test set. The subgraph extraction strategy is used for large-scale networks to reduce computational cost. Then link information decays according to the historical time by *ASF*. Secondly, by constructing the processed data, we get the weighted adjacency matrices at different timestamps and the latent matrices on this basis. Thirdly, considering node asymmetry, the data is input into our link prediction model to evaluate the generation of new links at the next time period. Link prediction can be described as, after determining the link prediction algorithm, assigning similarity scores to node pairs in the network that are not connected. All the nonexistent links are sorted in descending order based on their similarity scores, and the top-L (L is a hyperparameter) links are selected as the prediction output.

Theoretical analysis of TLPSS. The expression of TLPSS similarity index is explicit and mathematically elegant. At first, we will give theoretical explanation of its validity from the perspective of its magnification over the RA method in a simplified but general situation. Furthermore, the magnification on ER[45] random network and other two special cases will also be given. From a general perspective, assuming that the number of common neighbors of the target link is c, the number of hidden neighbors is d. Since our main purpose here is to determine the average magnification, we simplify the topological structure and the weight of links around the target pair of nodes. For notations, the average degree of the network is r, and the weights of the actual edges and latent edges in the network are set as α and β (both are constant) respectively. According to the TLPSS function, the average value of target link similarity is approximated as follows.

$$\begin{aligned}\overline{TLPSS(x,y)} &= \frac{1}{2}(\overline{score(x\rightarrow y)} + \overline{score(y\rightarrow x)})\\ &= \overline{score(x\rightarrow y)}\\ &\approx \sum_{z_i\in\Gamma(x)\cap\Gamma(y)} \frac{\overline{A^T(x,z_i)}}{\overline{w^T(z_i)}} + \sum_{z_i\in H_x} \frac{\overline{B^T(x,z_i)}}{\overline{w^T(z_i)}}\\ &= c\cdot\frac{\alpha}{\alpha\cdot r} + d\cdot\frac{\beta}{\alpha\cdot r+\beta}\end{aligned}$$

We know that the average value of target link similarity based on RA index can be expressed as $\overline{RA(x,y)} = c/r$. Therefore, the magnification of our proposed method TLPSS relative to RA can be expressed as follows.

$$magnification = \frac{\frac{c}{r} + \frac{d\beta}{\alpha r+\beta}}{\frac{c}{r}} = 1 + \frac{dr\beta}{c(\alpha r+\beta)}$$

In special cases, given the ER random network $G = (n, m)$, where n is the number of node, m is the linking probability between any pair of nodes. The average common neighbors and average hidden neighbors are as follows:

$$E(c) = C^1_{n-2}m^2 = (n-2)m^2,\ E(d) = C^2_{n-2}m^3 = \frac{(n-2)(n-3)}{2}m^3.$$

Thus, the magnification on the ER random network is:

$$magnification = 1 + \frac{\frac{(n-2)(n-3)}{2}m^3\cdot nm\cdot\beta}{(n-2)m^2(\alpha\cdot nm+\beta)} \quad \approx 1 + \frac{\beta}{2\alpha}\cdot\frac{n(n-2)(n-3)m^4}{n(n-2)m^3} = 1 + \frac{\beta}{2\alpha}\cdot(n-3)m.$$

Furthermore, there are two special cases we cannot ignore. First one is when two target nodes do not possess any common neighbor nodes, in this situation we could see that the RA index will give a value of zero while our TLPSS still have a positive chance to rank this type of unconnected nodes, i.e., our method is expected to perform better than RA index. The second situation is when we set β equals to zero, in this case our method has no difference with the RA index, which means that if we do not use latent edges our method will have the same performance as the RA index. Both conclusions can be verified in the experiment section.

For other properties, it is possible that nodes which are completely separated from the target links could have effects in the prediction. We can try to mine the simplex structure of the nodes separated from the target link. For more complex graphs, the local structure may not be directly decomposed into 2-simplices, but can be decomposed into high order simplex structures.

Computational cost of the TLPSS similarity method.

For notations, we assume that n is the number of nodes, k is the largest degree of a node. As mentioned earlier, the TLPSS majorly concerned about 2-hop and 3-hop neighbors, so its time complexity is $O(nk^3 f(k))$ where $f(k)$ is the computational cost for the calculation of similarity score between a pair of nodes. According to the symmetry of Eq. (10) and Eq. (11), $f(k)$ is decided by one of them. For the first term in Eq. (10), the intersection between two sets of size l and j can be computed with complexity $O(l+j)$ if we use the hash table technique. In this situation, the computation cost of the intersection is $O(k+k)$, and for each term in the summation the computation complexity is $O(1)$ since it is solely a lookup operation. For the second item in Eq. (10), the size of hidden neighbor set is at most equals to $k*k$ and for each term in the summation, the computational cost remains $O(1)$ if the latent matrix B^t is determined. According to Eq. (7,8,9), especially Eq. (9), we could see that the computation cost of latent matrix B^t requires $O(n*k^2*(k+k)) = O(nk^3)$ since only 2-hop neighbors are concerned. In summary, the computation cost of $f(k)$ equals to $O(2k*1+k^2*1) = O(k^2)$, which makes the total computation cost of the TLPSS method equals to $O(nk^3*k^2+nk^3) = O(nk^5)$. Due to the sparsity of network, $k << n$ stands in most large scale networks, and generally is a constant. According to[46], the computation cost of our method is competitive to the CN, RA, CAR indexes, which is acceptable since our index is a quasi-local method.

Experimental setting

In this section, we conduct experiments to evaluate the effectiveness of the proposed approach by using six real-world datasets for link prediction tasks, and compare its performance with six baseline algorithms. At first, we briefly introduce the datasets from different domains.

Dataset	Node number	Edge number	Ave. Degree	Start date	End date	Total duration	Snapshot number
Contact	273	28227	206.78	1970/1/1	1970/1/4	70h	70/h
DBLP	1169	10667	18.24	1986/1/1	1996/1/1	10y	10/y
Digg	3159	17661	11.18	2008/11/3	2008/11/11	8d	192/h
Enron	883	31092	70.42	2000/2/15	2000/6/14	4m	17/w
Facebook	3877	30480	15.72	2007/11/30	2008/8/26	9m	270/d
Prosper	2561	46540	36.34	2006/10/10	2006/12/11	2m	60/d

Table 2. Network datasets statistics.

Datasets description.

- **Contact**[47]: This network represents contacts between people, which is measured by carried wireless devices. Each node represents a person, and an edge between two persons shows that there was a contact between them.
- **DBLP**[48]: This is the citation network of DBLP, a database of scientific publications such as papers and books. Each node in the network is a publication, and each edge represents a citation of a publication by another publication.
- **Digg**[49]: This is the reply network of the social news website Digg. Each node in the network is a user of the website, and each edge denotes that one user replied to another user.
- **Enron**[50]: The Enron email network consists of emails sent between employees of Enron. Nodes in the network are individual employees and edges are individual emails. It is possible to send an email to oneself, and thus this network contains loops.
- **Facebook**[51]: This network contains friendship data of Facebook users. A node represents a user and an edge represents a friendship between two users.
- **Prosper**[52]: This network represents loans between members of the peer-to-peer lending network at Prosper.com. The network is directed from lender to borrower. Each edge is tagged the timestamps when the loan was occured.

All of these datasets are dynamic networks, i.e., each edge is annotated with timestamps showing the formation time. Since our main concern is whether there will be an edge between two nodes, the direction of the edge in the network is eliminated in the experiment. Table 2 shows major information of those datasets. *Total duration* is the length of the time span of dynamic networks, specifically, *h, d, w, m* and *y* stand for *hour, day, week, month* and *year* respectively. *Snapshot number* denotes the decay times of the network divided by the time information decay period, which is determined by the edge distribution of each dataset. Besides, we normalize the time attribute so that the timestamp of network starts from 1. In the link prediction evaluation stage, the existing link set E_t in the network is divided into two sets: train set $E(T)$ and test set $E(P)$ according to time evolution. The ratio between them is around 9:1. Link prediction focuses on the ranking results of node pairs. We found that different values of parameter a have no significant effect on the sorting results through the experiment, as long as it can help us anchor the ASF function so that its initial state is at a high level. So we set $a = 5$ to facilitate the selection of the parameter p and q.

Baseline methods and evaluation metrics.

Baseline methods. We compare our proposed model TLPSS with the following link prediction methods. These methods are usually used for static networks, it can also be applied to time-varying networks by aggregating all edges from different timestamps to one network. We make improvements over traditional baseline methods. The number of common neighbors of target link and the number of triangular closures around them are determined by the weights of edges. The specific definitions are shown in the Table 3.

Evaluation metrics. We use two commonly adopted evaluation metrics, AUC[54] and precision[55] to systematically evaluate the performance of the aforementioned methods. AUC can be interpreted as the probability that the similarity value of a randomly chosen new link is greater than a randomly chosen nonexistent link. A larger AUC value means better performance of the model. AUC measures the accuracy of the algorithm from a general perspective, while sometimes we pursue how many positive items the top part of link prediction methods output contains. Precision considers whether the edge in the top-L position is accurately predicted.

Results and discussions

In this section, we verify the effectiveness of TLPSS in real-world datasets with different evaluation metrics. Table 4 shows the AUC performance of different approaches on six dynamic networks, and the best performance on every datasets is highlighted. The proposed TLPSS model outperforms all baselines consistently across all six dynamic networks. The average performance of the TLPSS model outperforms other baseline methods about 15%, especially in Digg and Prosper datasets, our model leads with 20% and 30% respectively. Our proposed model TLPSS can be regarded as asymmetric modification of RA if we remove the latent edge terms in Eq. (10) and Eq. (11). Experimental results illustrate that capturing the local structure around the endpoints separately

Baseline Methods	Description	Definition
Common neighbors (CN)	The algorithm uses the number of common neighbors as an indicator to measure the possibility of establishing a link between two nodes[13]	$CN\_ASF(x,y) = \frac{1}{2} \sum_{z \in \Gamma(x) \cap \Gamma(y)} A^T_{x,z} + A^T_{y,z}$
Jaccard Index (JA)	This algorithm evaluates the probability of connecting edges also by measuring the number of common neighbors, it is the normalized version of *CN_ASF*[16]	$JA\_ASF(x,y) = CN\_ASF(x,y)/(w^T(x) + w^T(y))$
Preferential Attachment (PA)	In this algorithm, the probability that the target link is connected is proportional to the product of the degrees of the two endpoints, it is a hub-promoted method[53]	$PA\_ASF(x,y) = w^T(x) \cdot w^T(y)$
Resource Allocation (RA)	Common neighbors serve as a medium for resource transfer, and the weight of common neighbors is inversely proportional to its degree[17]	$RA\_ASF(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{w^T(z)}$
Cannistrai Alanis Ravai (CAR)	The algorithm utilizes the links between commmon neighbors, along with common neighbors information, where LCL'(x,y) is total weights of links between common-neighbors[14]	$CAR\_ASF(x,y) = CN\_ASF(x,y) \cdot LCL'(x,y)$
Clustering Coefficient-based Index (CCLP)	This metric employs clustering coefficient of common neighbors to reflect the density of triangles within a local network environment, where Δ' is the total weight of weighted triangles among common-neighbors[15]	$CCLP\_ASF(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{\Delta'}{d(z) \cdot (d(z)-1)/2}$

Table 3. Baseline link prediction methods for temporal networks.

AUC	CN	JA	PA	RA	CAR	CCLP	TLPSS
Contact	0.9525	0.8611	0.9020	0.9324	0.9334	0.8495	**0.9751**
DBLP	0.8627	0.8591	0.5201	0.8718	0.7748	0.8105	**0.9183**
Digg	0.6426	0.6445	0.5089	0.6472	0.6525	0.6119	**0.8905**
Enron	0.8872	0.8730	0.4681	0.8852	0.8745	0.8277	**0.9014**
Facebook	0.7505	0.7518	0.4731	0.7520	0.5798	0.7453	**0.9093**
Prosper	0.4018	0.4102	0.4639	0.3907	0.4993	0.4036	**0.7493**

Table 4. Comparison of the AUC value between TLPSS and baseline methods. Significant values are in bold.

Precision	CN	JA	PA	RA	CAR	CCLP	TLPSS
Contact	**0.9900**	0.3700	0.6600	0.9700	**0.9900**	0.9600	0.9600
DBLP	0.2100	0.1700	0.0000	0.0210	0.1400	0.0800	0.3600
Digg	0.0670	0.0530	0.0000	0.0240	0.0830	0.0070	**0.0910**
Enron	0.6500	0.6500	0.0000	0.2400	0.6100	0.1900	**0.6800**
Facebook	0.0080	0.0090	0.0000	0.0030	0.0050	0.0080	**0.0100**
Prosper	0.0004	0.0008	0.0024	0.0004	0.0024	0.0028	**0.0032**

Table 5. Comparison of the Precision value between TLPSS and baseline methods. Significant values are in bold.

could improve the link prediction performance. Besides, the reason for the superiority of TLPSS is that considering the latent edges in the network can address the cold-start problem for traditional common neighbor based link prediction methods, since the sparsity of the network might lead to a lack of common neighbors. In conclusion, the clear domination of the TLPSS index indicates that a deep understanding of life cycle of information and topology information could be converted to an outstanding link prediction algorithm.

Table 5 reports the precision values of TLPSS and other similarity algorithms. Due to the different scales of datasets, for *Contact, DBLP, Digg, Enron, Facebook* and *Prosper*, we set $L = \{100, 100, 1000, 100, 1000, 2500\}$ respectively. It can be seen from the table that the proposed method TLPSS is superior to other methods and can provide the highest accuracy on most datasets. In Contact dataset, five of the competing methods all achieve superior performance. It shows that in densely connected network, triangular closure structures based methods like CN are already sufficient. Besides, compared with CN on Contact dataset, JA has much lower performance, this demonstrates that the normalization operation does not always work. To sum up, TLPSS model has better accuracy on most sparse networks, which indicates that the consideration of hidden neighbor and latent edge can properly reveal the structural information around the target link.

Sensitivity test of parameter *p* in ASF. We first study the impact of different setting of the parameter p in Eq. (2), and set the parameter $q = 1$. Figure 6 shows the performance of different methods with varied parameter p on six real-world datasets. For *Contact, DBLP, Digg, Enron, Facebook* and *Prosper* datasets, we obtain the optimal value of parameter $p = \{3, 1, 10, 2.5, 5, 7\}$ respectively. There are several interesting phenomenons. First, TLPSS outperforms all other methods in most cases, and it can be interpreted that the consideration of hidden

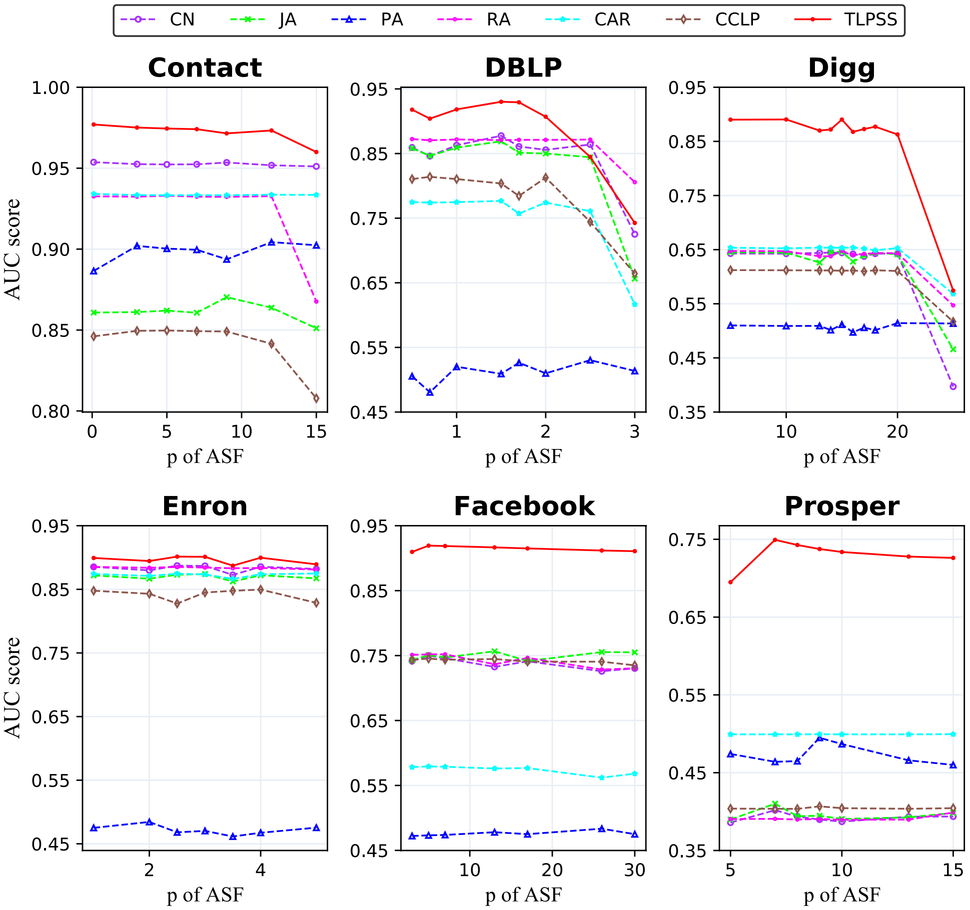

Figure 6. Performance comparison of varying parameter p in different dynamic networks. All methods are based on the same temporal information decayed by *ASF*. The performance of TLPSS is superior to other baseline methods.

neighbor and latent edge could unveil the spatial structure features around the target link. Second, the performance of most methods drop quickly when the parameter p takes a large value on *DBLP* and *Digg* datasets. We hold that the large number of parameter p will result in a longer decay time for the information, and inadequate utilization of temporal information due to the weight of new edge and historical edge is almost equal.Third, the optimal parameter p is different for each dataset, which indicates that the decay rate of datasets in different domains can be revealed by *ASF*. Forth, in *Digg* and *Facebook* datasets, the average degrees of these two social networks are low, common neighbor-based methods have similar performance. Unlike other approaches, TLPSS takes historical temporal information and simplex structure into account, thus, it has further improved the overall performance on temporal networks.

Performance of latent matrix sequence in real-world networks. Based on the definition of *ASF* and *LMS*, we can conclude that the weight of the latent edge in latent matrix is closely related to the parameter q, and its upper bound is the lower bound of *ASF*, which is $q/(q+1)$. In order to further understand the mechanism of proposed model TLPSS, the influence of parameter q on AUC value is demonstrated in Fig. 7. We set the parameter p for each dataset to be the optimal value according to the former experiment. Besides, we choose different values of parameter q, which varies from 0 to 10 with step size 1, to compute AUC values of different algorithms. From Fig. 7, AUC value of TLPSS model fluctuates greatly when parameter q increases from 0 to 1. The special case of $q = 0$ indicates that there are no latent edges considered according to Eq. (7). If we remove the latent edge terms in Eq. (10) and Eq. (11), we could see that TLPSS is the asymmetric modification of RA. This will make the 2-simplex structure composed of endpoint and hidden neighbor to lose effect and bring damage to the performance of TLPSS method. Evidence can be found at the initial point of curves in Fig. 7, which shows that the AUC values of TLPSS and RA indexes are almost equal at a lower value. The AUC value of the TLPSS model increases significantly when parameter q varies from 0 to 1, which proves that considering latent edges in the network can address the cold-start problem. Traditional common neighbor-based link prediction methods face this problem since the sparsity of the network might lead to a lack of common neighbor.

It is clear that as q increases from 0 to 10, the prediction accuracy of TLPSS increases till an optimal value, after which it maintains stable. All methods are not sensitive to the change of parameter q in the stable state, this

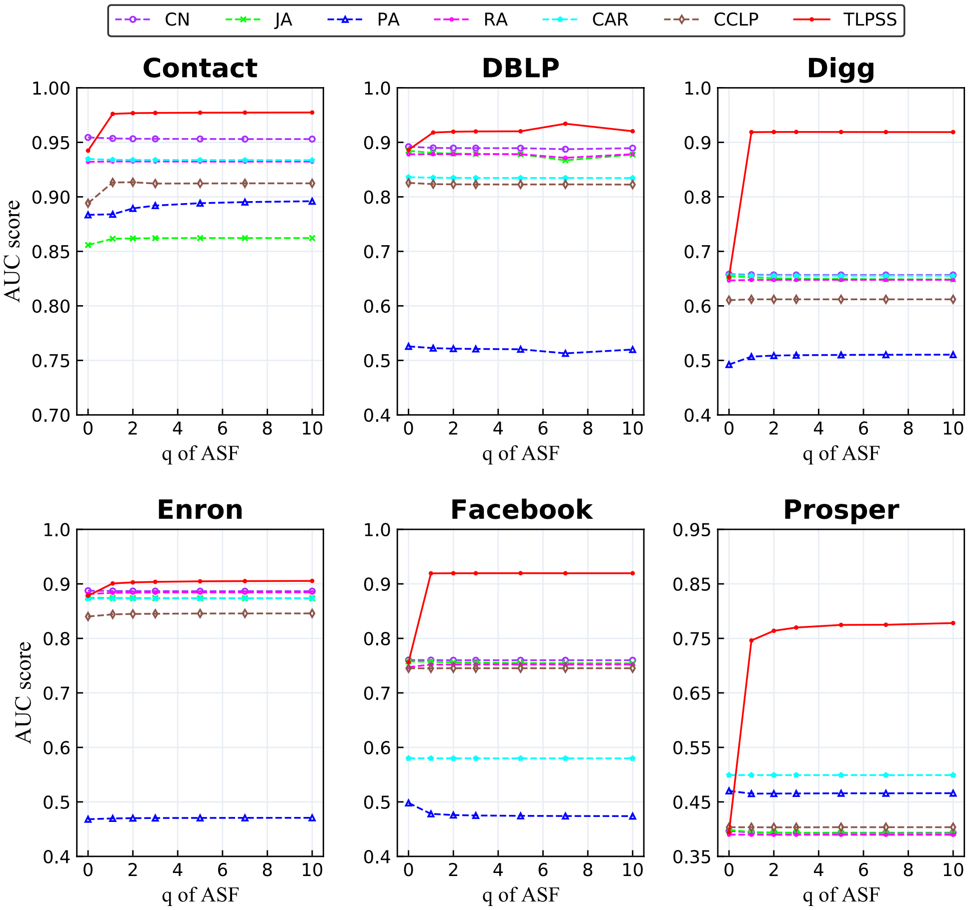

Figure 7. Performance comparison of varying parameter q in different dynamic networks. The AUC value of the TLPSS model includes rising stage and stable stage. Explosive rising stage illustrates the effectiveness of latent edge composed of 2-simplices structure.

is because that the similarity scores of positive and negative samples increase proportionally, and their ordinal relationship remains unchanged. Moreover, compared with TLPSS, other similarity methods have lower average performance. It is evident that TLPSS considers the role of latent edges composed of simplex high-order structures, which makes the surrounding topological information richer. To sum up, the large number of q leads to little fluctuation on TLPSS. According to the experimental results, it is recommended to set the parameter $q = 1$.

Conclusion

In this paper, we concentrate on the link prediction problem and design a general framework for temporal networks. We first provide a new time decay function *ASF* to quantify the remaining information of different timestamps links. Next, *HNS* and *LE* are introduced for the target link to extract the surrounding 2-simplex high-order structures. Besides, *LMS* effectively quantifies the weights of latent edges in the network, which alleviates the problem that traditional similarity methods are affected by lack of common neighbors due to the sparsity of the network. Finally, from the perspective of node asymmetry in the network, we propose the temporal link prediction method TLPSS by combining 2-simplex structural information in adjacency matrix and latent matrix. We theoretically analyze the optimality and validity of the parameters in the model. Extensive experiments on multiple datasets from different fields demonstrate the superiority of our model TLPSS compared to other baseline approaches. Our future work will focus on link prediction in directed temporal network with the consistency of life cycle of information and high-order structures. Also, the combination of *ASF* with other types of structures extracted with deep learning methods is left for further research.

Data availability

All datasets in this paper are available at http://konect.cc/networks/.

Received: 2 June 2022; Accepted: 23 September 2022
Published online: 05 October 2022

References

1. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. Rep.* **424**, 175–308 (2006).
2. Hand, D. J. Principles of data mining. *Drug Saf.* **30**, 621–622 (2007).
3. Liben-Nowell, D. & Kleinberg, J. The link-prediction problem for social networks. *J. Am. Soc. Inform. Sci. Technol.* **58**, 1019–1031 (2007).
4. Qi, Y., Bar-Joseph, Z. & Klein-Seetharaman, J. Evaluation of different biological data and computational classification methods for use in protein interaction prediction. *Prot. Struct. Funct. Bioinf.* **63**, 490–500 (2006).
5. Kuhn, F. & Oshman, R. Dynamic networks: models and algorithms. *ACM SIGACT News* **42**, 82–96 (2011).
6. Zhang, Z. *et al.* Efficient incremental dynamic link prediction algorithms in social network. *Knowl.-Based Syst.* **132**, 226–235 (2017).
7. Lu, J., Wu, D., Mao, M., Wang, W. & Zhang, G. Recommender system application developments: a survey. *Decis. Support Syst.* **74**, 12–32 (2015).
8. Wigand, R. T. Electronic commerce: Definition, theory, and context. *Inf. Soc.* **13**, 1–16 (1997).
9. Kossinets, G. & Watts, D. J. Empirical analysis of an evolving social network. *Science***311**, 88–90 (2006).
10. Hasan, M. A. & Zaki, M. J. A survey of link prediction in social networks. In *Social Network Data Analytics*, 243–275 (Springer, 2011).
11. Yu, W., Cheng, W., Aggarwal, C. C., Chen, H. & Wang, W. Link prediction with spatial and temporal consistency in dynamic networks. In *IJCAI*, 3343–3349 (2017).
12. Li, X., Liang, W., Zhang, X., Liu, X. & Wu, W. A universal method based on structure subgraph feature for link prediction over dynamic networks. In *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 1210–1220 (IEEE, 2019).
13. Lorrain, F. & White, H. C. Structural equivalence of individuals in social networks. *J. Math. Sociol.* **1**, 49–80 (1971).
14. Cannistraci, C. V., Alanis-Lobato, G. & Ravasi, T. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Sci. Rep.* **3**, 1–14 (2013).
15. Wu, Z., Lin, Y., Wang, J. & Gregory, S. Link prediction with node clustering coefficient. *Physica A* **452**, 1–8 (2016).
16. Sorensen, T. A. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.* **5**, 1–34 (1948).
17. Zhou, T., Lü, L. & Zhang, Y.-C. Predicting missing links via local information. *Eur. Phys. J. B* **71**, 623–630 (2009).
18. Tsai, C.-H., Chih, Y.-T., Wong, W. H. & Lee, C.-Y. A hardware-efficient sigmoid function with adjustable precision for a neural network system. *IEEE Trans. Circuits Syst. II Express Briefs* **62**, 1073–1077 (2015).
19. Martínez, V., Berzal, F. & Cubero, J.-C. A survey of link prediction in complex networks. *ACM Comput. Surv. (CSUR)* **49**, 1–33 (2016).
20. Wang, C., Satuluri, V. & Parthasarathy, S. Local probabilistic models for link prediction. In *7th IEEE International Conference on Data Mining (ICDM 2007)*, 322–331 (IEEE, 2007).
21. Zhao, H., Du, L. & Buntine, W. Leveraging node attributes for incomplete relational data. In *International Conference on Machine Learning*, 4072–4081 (PMLR, 2017).
22. Shuxin, L., Xing, L., Hongchang, C. & Kai, W. Link prediction method based on matching degree of resource transmission for complex network. *J. Commun.* **41**, 70 (2020).
23. Javari, A., Qiu, H., Barzegaran, E., Jalili, M. & Chang, K. C.-C. Statistical link label modeling for sign prediction: Smoothing sparsity by joining local and global information. In *2017 IEEE International Conference on Data Mining (ICDM)*, 1039–1044 (IEEE, 2017).
24. Al Hasan, M., Chaoji, V., Salem, S. & Zaki, M. Link prediction using supervised learning. In *SDM06: Workshop on Link Analysis, Counter-Terrorism and Security*, vol. 30, 798–805 (2006).
25. Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning* **14**, 1–159 (2020).
26. Perozzi, B., Al-Rfou, R. & Skiena, S. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 701–710 (2014).
27. Grover, A. & Leskovec, J. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864 (2016).
28. Srilatha, P. & Manjula, R. Similarity index based link prediction algorithms in social networks: A survey. *J. Telecommun. Inf. Technol.* (2016).
29. Cheng, W. *et al.* Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 805–814 (2016).
30. Chen, H. & Li, J. Exploiting structural and temporal evolution in dynamic link prediction. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 427–436 (2018).
31. Benson, A. R., Abebe, R., Schaub, M. T., Jadbabaie, A. & Kleinberg, J. Simplicial closure and higher-order link prediction. *Proc. Natl. Acad. Sci.* **115**, E11221–E11230 (2018).
32. Schaub, M. T., Benson, A. R., Horn, P., Lippner, G. & Jadbabaie, A. Random walks on simplicial complexes and the normalized hodge 1-laplacian. *SIAM Rev.* **62**, 353–391 (2020).
33. Millán, A. P., Torres, J. J. & Bianconi, G. Synchronization in network geometries with finite spectral dimension. *Phys. Rev. E* **99**, 022307 (2019).
34. Iacopini, I., Petri, G., Barrat, A. & Latora, V. Simplicial models of social contagion. *Nat. Commun.* **10**, 1–9 (2019).
35. Battiston, F. *et al.* The physics of higher-order interactions in complex systems. *Nat. Phys.* **17**, 1093–1098 (2021).
36. Chi, Y., Song, X., Zhou, D., Hino, K. & Tseng, B. L. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 153–162 (2007).
37. Levitan, K. B. Information resources as "goods" in the life cycle of information production. *J. Am. Soc. Inf. Sci.***33**, 44–54 (1982).
38. Anderson, C. R. & Zeithaml, C. P. Stage of the product life cycle, business strategy, and business performance. *Acad. Manag. J.* **27**, 5–24 (1984).
39. Lester, D. L., Parnell, J. A. & Carraher, S. Organizational life cycle: A five-stage empirical scale. *Int. J. Org. Anal.* (2003).
40. Milo, R. *et al.* Network motifs: Simple building blocks of complex networks. *Science* **298**, 824–827 (2002).
41. Ugander, J., Backstrom, L., Marlow, C. & Kleinberg, J. Structural diversity in social contagion. *Proc. Natl. Acad. Sci.* **109**, 5962–5966 (2012).
42. Grilli, J., Barabás, G., Michalska-Smith, M. J. & Allesina, S. Higher-order interactions stabilize dynamics in competitive network models. *Nature* **548**, 210–213 (2017).
43. Lou, T., Tang, J., Hopcroft, J., Fang, Z. & Ding, X. Learning to predict reciprocity and triadic closure in social networks. *ACM Trans. Knowl. Discov. Data (TKDD)* **7**, 1–25 (2013).
44. Latora, V. & Marchiori, M. A measure of centrality based on network efficiency. *New J. Phys.* **9**, 188 (2007).
45. Erdős, P. *et al.* On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci* **5**, 17–60 (1960).
46. Martínez, V., Berzal, F. & Cubero, J.-C. A survey of link prediction in complex networks. *ACM Comput. Surv. (CSUR)* **49**, 1–33 (2016).
47. Chaintreau, A. *et al.* Impact of human mobility on opportunistic forwarding algorithms. *IEEE Trans. Mob. Comput.* **6**, 606–620 (2007).

48. Ley, M. The dblp computer science bibliography: Evolution, research issues, perspectives. In *International Symposium on String Processing and Information Retrieval*, 1–10 (Springer, 2002).
49. De Choudhury, M., Sundaram, H., John, A. & Seligmann, D. D. Social synchrony: Predicting mimicry of user actions in online social media. In *2009 International Conference on Computational Science and Engineering*, vol. 4, 151–158 (IEEE, 2009).
50. Klimt, B. & Yang, Y. The enron corpus: A new dataset for email classification research. In *European Conference on Machine Learning*, 217–226 (Springer, 2004).
51. Viswanath, B., Mislove, A., Cha, M. & Gummadi, K. P. On the evolution of user interaction in facebook. In *Proceedings of the 2nd ACM workshop on Online social networks*, 37–42 (2009).
52. Redmond, U. & Cunningham, P. A temporal network analysis reveals the unprofitability of arbitrage in the prosper marketplace. *Expert Syst. Appl.* **40**, 3715–3721 (2013).
53. Barabási, A.-L. & Albert, R. Emergence of scaling in random networks. *Science***286**, 509–512 (1999).
54. Wang, H., Hu, W., Qiu, Z. & Du, B. Nodes' evolution diversity and link prediction in social networks. *IEEE Trans. Knowl. Data Eng.* **29**, 2263–2274 (2017).
55. Ren, Z.-M., Zeng, A. & Zhang, Y.-C. Structure-oriented prediction in complex networks. *Phys. Rep.* **750**, 1–51 (2018).

Acknowledgements

This work was supported by the Research and Development Program of China (Grant No. 2018AAA0101100), the National Natural Science Foundation of China (Grant Nos. 62141605, 62050132), the Beijing Natural Science Foundation (Grant Nos. 1192012, Z180005).

Author contributions

R.Z. designed the research, analysed the results and wrote the paper. Q.W. prepared figures, analysed data and evaluated the algorithm. Q.Y. conducted the experiments, analysed the results and wrote the paper. W.W. designed the research, analysed the results and wrote the paper.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to W.W.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.