Joint multi-label learning and feature extraction for temporal link prediction

Shiyin Tan[a], Xiaoke Ma[a,d,*], Xianghua Xie[c], Xiaoxiong Zhong[b], Jingjing Deng[c]

[a]*School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, China*
[b]*Peng Cheng Laboratory, Shenzhen, Guandong, China*
[c]*Department of Computer Science, Swansea University, U.K.*
[d]*State Key Laboratory for Novel Software Technology,Nanjing University, Nanjing, Jiangsu, China*

Abstract

Networks derived from various disciplinary of sociality and nature are dynamic and incomplete, and temporal link prediction has wide applications in recommendation system and data mining system, etc. The current algorithms first obtain features by exploiting the topological or latent structure of networks, and then predict temporal links based on the obtained features. These algorithms are criticized by the separation of feature extraction and link prediction, which fails to fully characterize the dynamics of networks, resulting in undesirable performance. To overcome this problem, we propose a novel algorithm by joint multi-label learning and feature extraction (called *MLjFE*), where temporal link prediction and feature extraction are integrated into an overall objective function. The main advantage of *MLjFE* is that the features and parameter matrix for temporal link prediction are simultaneously learned during optimization procedure, which is more precise to capture dynamics of networks, improving the performance of algorithms. The experimental results on a number of artificial and real-world temporal networks demonstrate that the proposed algorithm significantly outperforms state-of-the-art methods, showing joint learning with feature extraction and temporal link prediction is promising.

*Corresponding author
Email address: xkma@xidian.edu.cn (Xiaoke Ma)

Keywords: Temporal Link Prediction; Non-negative matrix factorization; Multi-Label Learning; Dynamic Networks

1. Introduction

Network (graph) is an efficient and effective tool to model and characterize many complex systems from nature and society, where each vertex represents an entity and each edge denotes the relation between a pair of vertices. For instance, in social networks vertices correspond to individuals and edges to the relations among them [1]. In cancer networks, biological molecules, such as genes and proteins, are denoted by vertices and biological interactions among genes, such as protein-protein interactions and transcriptional factor binding interactions, are represented by edges [2]. There are various complex networks, including social networks [3], gene regulation networks [4], transportation networks [5], and scientific collaboration networks [6].

Network analysis aims at extracting interesting graph patterns by exploiting the topological structure of networks, which shed light on the structure and function of underlying systems. For instance, the hub vertices (with large degree) correspond to the scientists with prestigious reputation in scientific collaboration networks [6]. The module structure is defined as a group of vertices whose connectivity within group is strong inside and weak outside. And, modules in protein interaction networks may correspond to protein complexes, which execute critical biological processes, such as apoptosis and gene expression regulation [7]. Moreover, the time dependence of overlapping communities uncovers basic relationships characterizing community evolution, which are essential for a deeper understanding of the development and self-optimization of society as a whole [8]. The prerequisite for network analysis is that the involved networks are reliable and complete.

However, most of available networks are incomplete because our knowledge about the underlying complex systems is really limited, which hampers the successful application of network analysis to discover interesting graph patterns.

For example, it is reported that more than 90% protein interactions of human are missing [9]. Thus, there is a critical need to complete the involved networks. The most reliable and straightforward strategy is to validate the interactions by experiments, which is unpracticable in many cases because of limitation of finance and technique constraints. Therefore, predicting missing links in networks based on the observation of interactions using the computational techniques is popular, which provides an alternative for experimental based methods [10].

Great efforts have been devoted to missing link prediction, which can be roughly classified into two categories: topological analysis [11] and matrix decomposition based methods [12]. The former ones aim at exploiting the similarity between vertex pair to predict missing links based on the assumption that vertices with similar structure are more likely to be connected. The difference among these algorithms lay on how to define and calculate the similarity between vertices, such as local topological structure information [13] and global reliable paths [11]. However, these methods is insufficient to fully characterize the structure of missing links since it only explores the topological information. To solve this problem, the latter methods exploit the latent structure by factorizing matrices associated with networks based on the assumption that the implicit features can complement topological information. For example, SEMAC that jointly exploits fine-grained node features as well as the overall graph topology, which represents vertices using subgraph embedding via convex matrix completion [12]. There are also variants of link prediction, such as hyperlink and multi-relational link prediction [14].

Nevertheless these algorithms focus on predicting missing links in static networks, ignoring the temporality of networks. Actually, dynamic networks are ubiquitous, implying that the structure of networks changes at various conditions and time steps. For example, switch of individual occupations and locations results in dynamicity of structure and patterns in social networks [15]. Cancer cells disrupt the biological functions, leading to cancer progression from benign to malignant stages. Thus, there is a critical need for link prediction in temporal networks since temporal networks are more precisely to describe the

underlying complex systems. Compared to missing link prediction, temporal link prediction forecasts the network at $T+1$ based on the observations from 1 to T [16].

Unfortunately, it is difficult to predict temporal links mainly due to several reasons. First, compared to static networks, dynamics of networks is difficult to accurately depict, which is the foundation for temporal link prediction. Second, the multiple layers of temporal networks pose a great challenge on designing effective and efficient algorithms to predict links. The most intuitive strategy for temporal link prediction is to collapse dynamic networks into static one, where link prediction algorithms can be directly applied. However, this strategy is criticized for incredibly sacrificing the accuracy of prediction [17].

To avoid collapsing temporal networks, many algorithms have been developed for temporal link prediction. For example, Dunlavy et al. [18] consider the CANDECOMP/PARAFAC tensor decomposition, which retains its natural three-dimensional structure, instead of collapsing the networks. Zhu et al. [19] propose a temporal latent space model (*TLSM*) for temporal link prediction based on the assumption that vertices can change smoothly in the latent space over time. However, the performance of *TLSM* largely depends on the quality of features at the previous time. To attack this issue, *SNMF-FC* obtains the features for vertices for each time using symmetric nonnegative matrix factorization (SNMF), and then predicts temporal links by collapsing the features at various time [17]. But, *SNMF-FC* has the limitation that the features of vertices at each various time are independent, failing to characterize dynamics of networks in features. Thus, Ma et al. [20] propose graph regularized nonnegative matrix factorization algorithm (*GrNMF*), where the historical topological structure information is incorporated into features of vertices at the current time via regularization strategy. *GrNMF* is superior to *SNMF-FC*, implying that fusing features and temporality is promising for temporal link prediction, which is also the motivation of this study.

Although considerable efforts have been devoted to the temporal link prediction, there are still many unsolved problems. For example, the theoretical foun-

dations for temporal link prediction is critical needed. In details, even though the equivalence among GrNMF and NMF is proven, the relation between temporal link prediction and typical algorithms such as multi-label learning is still unknown. Second, the available methods consists of two major components: feature extraction for each time and temporal link prediction based on the obtained features. However, these two major steps run independently, which may fail to fully characterize the structure of temporal networks. Actually, great evidences demonstrate that joint learning is more precise than independently learning [21]. Therefore, one of the motivation of this study is to investigate the possibility to joint link prediction and feature extraction.

To address these problems, we further demonstrate that multi-label learning is a generalization of link prediction, which is also the reason why it is selected. And, a novel algorithm by joint multi-label learning and feature extraction for temporal link prediction ($MLjFE$) is proposed, where parameter matrix for link prediction and features simultaneously are learned. The advantage of $MLjFE$ is that it extract both local and glocal information at same time since features can capture the high-order local structure information of the networks during optimization procedure. In this case, the temporal link prediction problem using joint learning is transformed into an constrained optimization problem. Then, constrained updating rules are deduced to optimize the objective function. Finally, jointing feature extraction and multi-label learning significantly improves the accuracy of algorithms since it provides a better way to characterize dynamics of networks.

In all, the main contributions of this study can be summarized as follows:

- We show link prediction is a special case of multi-label learning, which is a good reason to explain why multi-label learning can be used for link prediction. And, we extend multi-label learning for temporal link prediction. Even though some efforts have been devoted to link prediction using multi-label learning [22], as far as we know, this study is the first multi-label learning for temporal link prediction.

- *MLjFE* is developed for temporal link prediction by joint multi-label learning and feature extraction, where features can capture the information in parameter matrix for temporal link prediction. It can also consider as a integrative framework for multiple algorithms for temporal link prediction.

- Experimental results over a number of both artificial and real world temporal networks demonstrate that the proposed approach significantly outperforms state-of-the-art methods without increasing time complexity.

The remainder of the paper is organized as follows. Section 2 introduces notations. Related work is summarized in Section 3. The proposed algorithm is described in Section 4. The experimental results on various temporal networks are shown in Section 5. The conclusion is drawn in Section 6.

2. Notations

Notations that are widely used in the forthcoming sections are described in this section.

Let $G = (V, E)$ be a graph with vertex set $V = (v_1, \ldots, v_n)$ and edge set E, where n is the number of vertices. The adjacent matrix of G is denoted by matrix $W \in R^{n \times n}$ whose element w_{ij} is the weight on edge (v_i, v_j). If network G is un-weighted, w_{ij} is 1 if edge (v_i, v_j) exists, 0 otherwise. The degree of vertex v_i is the sum of weights on edges connected to it, i.e. $d_i = \sum_j w_{ij}$. The laplacian matrix is defined as $L = D - W$, where D is the degree diagonal matrix, i.e. $D = diag(d_1, d_2, \ldots, d_n)$. A low-case bold letter $\boldsymbol{x}$ presents a vector and a upper-case letter Z presents a matrix in this paper.

Let $\{1, 2, \ldots, T\}$ be the set of time steps (time for short). Any variable with attached subscript t denotes value of the corresponding variable at time t. *Temporal* (*dynamic*) networks $\mathcal{G}$ is combination of a sequence of graphs with order, i.e. $\mathcal{G} = (G_1, \ldots, G_T)$, where G_t is the snapshot at time t. Temporal networks $\mathcal{G}$ can be represented by a tensor $\mathcal{W} \in R^{n \times n \times T}$, where w_{ijt} corresponds to the weight on edge (v_i, v_j) in G_t. Let $W_{i:t}$ and $W_{:it}$ denote the i-th row and column of W_t, respectively.

The temporal link prediction is defined as: *given temporal networks* $\mathcal{G} = (G_1, \ldots, G_T)$*, how to predict network at* $T+1$*, i.e. how to construct a function* f *such that*

$$W_{T+1} = f(W_1, \ldots, W_T). \tag{1}$$

3. Related work

In this section, we briefly review the matrix based algorithm for temporal link prediction problem, which are classified into three categories: network collapse, topological analysis and matrix factorization based algorithms.

3.1. Network collapse based methods

The simplest strategy for temporal link prediction is to directly apply algorithms for link prediction by collapsing dynamic networks into a static network. Collapsed tensor (CT) [23] adopts the average of edge weights on all slides of dynamic networks as

$$W^* = \sum_{t=1}^{T} W_t/T. \tag{2}$$

However, it is criticized for setting the importance of network G_t as a constant $1/T$. Actually, snapshots close to $T+1$ are more important than those far away from $T+1$ because evolution of networks originates from 1 to T. Thus, weighted CT (WCT) [24] collapse networks $\mathcal{G}$ as

$$W^* = \sum_{t=1}^{T} (1-\theta)^{T-t} W_t, \tag{3}$$

where $\theta \in (0, 1)$ is a parameter controlling the relevant importance of W_t.

CT and WCT predict temporal links as $W_{T+1} = W^*$. The advantage of collapse based methods is simple. However, they are criticized by the low accuracy of prediction because they assume that edges in temporal networks are independent.

3.2. Topological analysis based methods

To overcome the limitation of network collapse based methods, the topological analysis based algorithms aim at obtaining the relations among vertices by exploiting the structure of networks. The critical difference among these algorithms is how to define the similarity among vertices.

The typical similarity for a pair of vertices is to count the percentage of overlapping neighbors. Specifically, given a pair of vertices v_i and v_j, the Jaccard coefficient is defined as

$$s_{ijt} = \frac{|N_i(W_t) \cap N_j(W_t)|}{|N_i(W_t) \cup N_j(W_t)|}, \tag{4}$$

where $N_i(W_t)$ is the set of neighbors of node i in network G_t. Nevertheless, Jaccard coefficient only explores the paths with length of 2, which is insufficient to fully characterize topological structure of networks. Katz [25] quantifies the closeness between a pair of vertices by counting the number of paths with various lengths connecting them, which is defined as

$$S_t = \sum_{i=1}^{\infty} \beta^i W_t^l = (I - \beta W_t)^{-1} - I, \tag{5}$$

where I an identity matrix.

There are also various similarity indexes for temporal link prediction, including graph communicability [17], eigenspace of spectrum of networks [26].

3.3. Matrix decomposition based methods

Although topological structure analysis overcomes the drawback of network collapse based methods, it is not a panacea because the latent information of networks cannot be fully depicted by topology. Matrix decomposition is popular to obtain latent feature for networks.

The most intuitive strategy is to factorize W^* using singular value decomposition (SVD), which approximates W^* by the product of three matrices such

that

$$W^* = P\Sigma Q^{'}, \tag{6}$$

where P, Q are left and right singular matrices respectively, and $\Sigma = diag(\delta_1, \ldots, \delta_n)$. The temporal links are predicted as

$$W_{T+1} = \sum_{i=1}^{k} \delta_i \mathbf{p}_i \mathbf{q}_i, \tag{7}$$

where $\mathbf{p}_i$ and $\mathbf{q}_i$ are the i-th column of P, Q respectively, and k is the number of singular vectors selected for prediction. Some constraints are imposed on matrix factorization in order to achieve better interpretation. For example, NMF approximates W^* using two low-rank nonnegative matrices X, Y such that

$$W^* \approx XY. \tag{8}$$

Factorizing collapsed matrix W^* is not popular since the collapsed network fails to preserve the topological structure of dynamic networks. The most intuitive way is to directly factorize $\mathcal{W}$ using tensor decomposition (TD) [18], which is defined as

$$\mathcal{W} = \sum_{i} \lambda_i \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i. \tag{9}$$

where $\circ$ denote the outer product between two vector, i.e. $\mathbf{a}_i \circ \mathbf{b}_i = \mathbf{a}_i \mathbf{b}_i^{\prime}$. And $\mathbf{a}_i$, $\mathbf{b}_i$, $\mathbf{c}_i$ are the decomposed vectors. Although TD identifies higher-order patterns in dynamic networks, the complexity and difficulty in incorporating priori information are two drawbacks.

To solve this problem, SNMF-FC [17] employes NMF to obtain feature matrix for each slide G_t as

$$W_t \approx X_t X_t^{'}. \tag{10}$$

where X_t is the extracted features in snapshot t. Then, it predicts temporal

links using feature matrices as

$$W_{T+1} = \sum_{i=1}^{T} (1-\theta)^{T-t} X_t X_t^{'}. \tag{11}$$

where θ controls the importance of different snapshots, and W_{T+1} is the prediction matrix for link prediction.

But, the feature matrices obtained by SNMF-FC are independent, ignoring the temporality of networks. To address this issue, GrNMF obtains feature matrix at t by simultaneously taking into account both W_t and W_{t-1} by minimizing the following cost

$$\|W_t - X_t X_t^{'}\|^2 + tr(X_t W_{t-1} X_t^{'}). \tag{12}$$

In addition, graph representation based methods, like DeepWalk[27], LINE[28], node2vec[29] etc., are also good at obtain latent feature for networks. We learn embedding matrix E from network by graph representation based methods, and then gain the similarity between nodes by analyzing embedding matrix, i.e. $S = \frac{EE^{'}}{2\|E\|^2}$ is the cosine similarity between nodes. However, these methods could not handle additional information, such as temporal smoothing information, during the representation learning process.

3.4. Multi-label learning and Link prediction

On an abstract level, the problems of multi-label learning (MLL) and link prediction (LP) are similar: LP aims to impute missing links and MLL aims to impute missing labels. Although LP and MLL are inherently interwoven, so far they have been mostly considered to be unrelated problems.

Since Chen et al. [30] introduce a novel method, called MLLP, which firstly combine two problems into a single joint objective function. The algorithm MLLP is designed for Bi-relational graphs, so the objective function is consist of three part: label prediction, link prediction, and label smoothness. MLLP do the network completion and label prediction simultaneously.

We treat the link prediction problem as multi-label learning problem, and we firstly introduce the temporal link prediction method which combine multi-label

learning, feature extraction, and temporal smoothness. In the subsequent section, we investigate the possibility of designing effective NMF-based algorithms by jointing the high-order topological structure and temporal information with multi-label learning.

4. Algorithm

The *MLjFE* is porposed in this section, which joint feature eaxtraction, multi-label learning, and temporal smoothing, as shown in Fig. 1. The objective function, optimization rules and algorithm analysis of the proposed algorithm are discussed in turn.

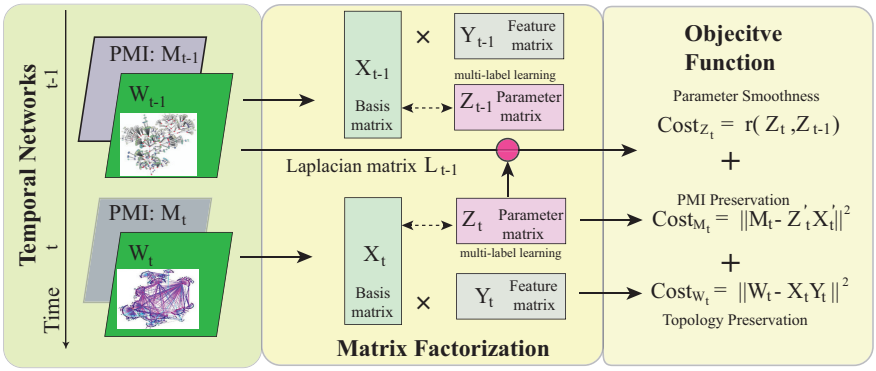

Figure 1: The overview of the proposed algorithm, which consists of two components: objective function and matrix factorization. The objective function compromises subcosts from multi-label learning, topological structure factorization and temporal smoothness. Matrix factorization procedure jointly factorizes adjacent and PMI matrices associated with temporal networks at the current time.

4.1. Objective function

Traditional classification algorithms, such as support vector machine (SVM), are criticized by the inefficiency for multiple labels. To overcome this issue, multi-label learning simultaneously predict multiple labels by constructing a function $f(\boldsymbol{x}, Z)$ where $\boldsymbol{x}$ and Z are feature vector of an object and parameter matrix. The output of multi-label learning is a vector with length l representing the various labels, i.e. $Z^{'}\boldsymbol{x} \in \{0,1\}^{l}$ [31]. As shown Fig.1, the objective function

of the proposed algorithm consists of three components: cost for multi-label learning, feature extraction and temporal smoothness, which are addressed in turns.

On the first concern, the multi-label learning can be easily applied to link prediction. Given the feature matrix for vertices $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^{'} \in R^{n \times k}$ of a network, multi-label learning constructs parameter matrix Z to predict links, such as $f(X, Z) = Z^{'}X^{'} \in \{0, 1\}^{n \times n}$. Thus, the link prediction is a special case of multi-label learning. To learn parameter matrix Z, multi-label learning minimizes the regularized empirical risk [31], which is defined as

$$J(Z) = \ell(W, X, Z), \tag{13}$$

where $\ell(W, X, Z)$ is the loss function to quantifying the goodness of prediction and usually the l_2 norm is adopted, i.e. $\ell(W, X, Z) = \|W - Z^{'}X^{'}\|^2$. Obviously, the empirical risk in Eq.(13) for G_t is reformulated as

$$J(Z_t) = \|W_t - Z_t^{'}X_t^{'}\|^2. \tag{14}$$

However, Eq.(14) cannot directly be applied to the temporal link prediction problem since it defaults the features is given by the data and ignores the dynamics of networks.

On the second concern, we expect the features of vertices can effectively capture the topological structure. Recently, representation models are originated from natural language process with an immediate purpose to learn contents c using core words w [32] by maximizing the function as

$$\max \sum_{w} \sum_{c} \log Pr(c|w) \tag{15}$$

where $Pr(c|w)$ is the condition probability for content c under word w. To extend representation model for networks, graph representation learning aim at learning continuous feature vector for each vertex, where critical topological

structure is preserved. It is proven that graph representation is equivalent with matrix factorization of pointwise mutual information (PMI) matrix[33], and the DeepWalk equivalent matrix is defined as [34]

$$m_{ij} = \log \frac{\left[\boldsymbol{e}_i \left(W + W^2 + \cdots + W^l\right)\right]_j}{l} \tag{16}$$

where $\boldsymbol{e}_i$ is the i-th standard basis and l is the window size. Thus, we expect feature matrix X_t hidden in PMI matrix M_t by matrix factorization, i.e.

$$J(X_t, Y_t) = \|M_t - X_t Y_t\|^2. \tag{17}$$

On the third one, Eqs.(14) and (17) quantify the costs for multi-label and feature extraction, failing to capture dynamics of networks. In our previous study [20], it is proven that incorporating dynamics of networks during feature extraction significantly improves the accuracy of temporal link prediction. Therefore, we expect parameter matrix Z_t also captures the topological structure of G_{t-1}. The most intuitive way is to measure the difference between parameter matrices at two consecutive times, i.e.

$$\begin{aligned} J(Z_t, Z_{t-1}) &= \|Z_t - Z_{t-1}\|^2. \\ &= tr((Z_t - Z_{t-1})(Z_t - Z_{t-1})') \end{aligned} \tag{18}$$

which denote that the parameter matrices changing gradually over time.

Therefore, we obtain the overall objective function at time t by combining Eqs.(14,17,18), i.e.

$$\begin{aligned} J_t &= J(Z_t) + \alpha J(X_t, Y_t) + \beta J(Z_t, Z_{t-1}) \\ &= \|W_t - Z_t^{'} X_t^{'}\|^2 + \alpha \|M_t - X_t Y_t\|^2 + \beta tr\left((Z_t - Z_{t-1})(Z_t - Z_{t-1})^{'}\right), \\ &\quad \text{s.t. } tr(X_t^{'} X_t - I) = 0, X_t \geq 0, Z_t \geq 0 \end{aligned} \tag{19}$$

where α and β are parameters determining the relevant importance of feature extraction and smoothness items, respectively. And non-negative constraints on

X_t can lead more easily discriminable clustering features and more interpretable results [35]. Furthermore, $tr(X_t^{'}X_t - I) = 0$ means the features extracted should be normalized during the process of optimization.

The model in Eq.(19) joints multi-label learning and feature extraction during optimization procedure with an immediate purpose to avoid local minima. The overall objective function of *MLjFE* is defined as the sum of J_t, i.e.

$$\begin{aligned} J = \sum_{t=1}^{T} J_t = & \sum_{i=1}^{T} (\|W_t - Z_t^{'}X_t^{'}\|^2 + \alpha\|M_t - X_tY_t\|^2 \\ & + \beta tr\left((Z_t - Z_{t-1})(Z_t - Z_{t-1})^{'}\right) \\ & \text{s.t. } tr(X_t^{'}X_t - I) = 0, X_t \geq 0, Z_t \geq 0, \forall t \in \{1, 2, \ldots, T\}. \end{aligned} \tag{20}$$

In this case, we transform the temporal link prediction problem into a joint optimization problem in Eq.(20). In the next subsection, we present the optimization procedures.

4.2. Optimization

It is difficult to directly optimize Eq.(20) since three matrices X_t, Z_t, Y_t are involved. Thus, an iterative three-step strategy is adopted, where at each iteration we optimize one matrix by fixing the others. The iteration is repeated until the algorithm converges or the maximum number of iterations is reached.

The Lagrange function for the optimization problem in Eq.(20) is constructed as

$$\begin{aligned} \mathcal{L}_t = & \frac{1}{2}(\|W_t - Z_t^{'}X_t^{'}\|_F^2 + \alpha\|M_t - X_tY_t\|_F^2 \\ & + \beta tr\left((Z_t - Z_{t-1})(Z_t - Z_{t-1})^{'}\right) + \gamma tr(X_t^{'}X_t - I)), \end{aligned} \tag{21}$$

where parameter γ controls the importance to normalize the features X_t.

By fixing Y_t and Z_t, the partial derivatives of $\mathcal{L}_t$ with respect to feature matrix X_t, Y_t is deduced as

$$\frac{\partial \mathcal{L}_t}{\partial X_t} = -(W_t^{'}Z_t^{'} + \alpha M_t Y_t^{'}) + X_t(Z_tZ_t^{'} + \alpha Y_tY_t^{'} + \gamma I). \tag{22}$$

Analogously, the partial derivatives of $\mathcal{L}_t$ with respect to feature matrix Y_t and Z_t is deduced as

$$\frac{\partial \mathcal{L}_t}{\partial Y_t} = \alpha(-X_t^{'} M_t + X_t^{'} X_t Y_t), \tag{23}$$

and

$$\frac{\partial \mathcal{L}_t}{\partial Z_t} = -X_t^{'} W_t^{'} + X_t^{'} X_t Z_t + \beta(Z_t - Z_{t-1}), \tag{24}$$

The nonnegativity of X_t is solved using Projected Gradient Methods for NMF (PGD)[36].

After obtaining feature matrix X_t, *MLjFE* predicts the temporal link W_{T+1} as

$$W_{T+1} = \sum_{t=1}^{T} \theta^{T-t} Z_t^{'} X_t^{'}, \tag{25}$$

where parameter $\theta \in$ [0,1] determines the importance of feature matrix X_t. In our previous study [20], θ=0.8 is a good choice. The procedure of *MLjFE* is illustrated in Algorithm 1.

4.3. Algorithm Analysis

The space complexity of *MLjFE* is $O(n^2T)$. Given a dynamic network $\mathcal{G}$, the 3-dimensional adjacency matrix $\mathcal{W}_{n\times n\times T}$ requires space $O(n^2T)$. The spcae complexity for matrix factorization is $O(nkT)$. Therefore, the overall complexity is $O(Tn^2)$.

Then, the time complexity is $O(k_1kn^2T)$, which is analyzed following. For updating feature matrix X_t, parameter matrix Z_t, and matrix Y_t, the time complex is $O(kn^2)$. So the overall time complexity of *MLjFE* is $O(k_1kn^2T)$, where k_1 is the number of iteration in Algorithm 1.

5. Experiment

To fully validate the performance of the proposed algorithm, a comparative comparison among various algorithms is conducted on a number of artificial and real dynamic networks.

Algorithm 1 The *MLjFE* algorithm

Input:

$\mathcal{G}$: Dynamic networks;
k: number of features;
α: weight for feature extraction;
β: weight for temporal smoothing

Output:

W_{T+1}: network at time T+1

Part 1: feature extraction and parameter learning

for t=1,2,...,T **do**
  Compute the PMI matrix M
  Make initial matrices X_t,Y_t,Z_t;
  for iteration until convergence or reach max iteration **do**
    Fixed matrices Y_t and Z_t, update X_t using Eq.(22);
    Handle nonnegativity using PGD;
    Fixed matrices X_t and Y_t, updating Z_t using Eq.(24);
    Handle nonnegativity of Z_t using PGD;
    Fixed matrices X_t and Z_t, updating Y_t using Eq.(23);
  end for
end for

Part 2: Temporal link prediction

Predicting temporal links using Eq.(25);
return W_{T+1}

5.1. Data and Setting

Seven typical algorithms are selected, including Katz index [25], GrNMF [20], SNMF-FC [17], GCNs [37], GCN-GAN [38], PMI, and CPLST_PMI [31]. The reasons for the selection of these algorithms will be shown below. Katz index is an excellent link prediction similarity based on topological structure analysis. GrNMF is the state-of-the-art matrix factorization-based methods without collapsing the dynamic network. SNMF-FC are the matrix factorization-based methods with feature collapsing. GCN-GAN is state-of-the-art deep learning algorithm, benefits of the graph convolutional network (GCN), long short-tern memory (LSTM), and the generative adversarial network (GAN). However, the space complexity of GCN-GAN is $O(n^3)$, which limits the dataset we can run. Therefore, GCNs as an alternative strategy of GCN-GAN, GCNs is for feature extraction, and then uses the Pearson similarity to calculate the prediction matrix. PMI and CPLST_PMI are closely related to the algorithm we proposed.

PMI is the matrix equivalent to DeepWalk, which perform matrix factorization on PMI matrix, and Pearson similarity calculating on the embeddings. CPLST_PMI is the non-joint version of *MLjFE*. CPLST_PMI performs feature extraction based on DeepWalk, and then train the multi-label classifier CPLST.

Eight datasets are employed to testify the performance of these algorithms, including both two artificial and six real temporal networks, which are widely adopted for temporal link prediction. The statistics of the six datasets are summarized in Table 1, with the number of vertices ranging from 128 to 39,685.

Given the temporal network $\mathcal{G}$ with T time steps, we predict the temporal links based on the networks from 1 to $T-1$, where network G_T is used to validate the performance of algorithms. All these algorithms run on HP workstation with Intel Core i7 3.2GHz CPU and 32G RAM using the default values of parameters. To quantify the accuracy of algorithms, both area under curve (AUC) and average precision(AP) score are employed.

Table 1: Statistics of temporal networks, where $|V|$ denotes the number of vertices, $|E|$ represents the number of edges, $|T|$ corresponds to the number of time steps.

Category	Description	Data	$\lvert V\rvert$	$\lvert E\rvert$	$\lvert T\rvert$
Real-world networks	DBLP	Scientists	31,855	91,095	5
	Sx-mathoverflow	Web	3,148	62,012	7
	Sx-askubuntu	Web	28,371	61,249	8
	email-EU-core	Email	857	150,309	8
	Facebook	Social networks	1,819	56,098	6
	CollegeMsg	Message networks	1,828	56,912	6
Artificial networks	SYN-FIX	NA	128	20480	10
	SYN-VAR	NA	256	59764	10

5.2. Artificial temporal networks

The artificial temporal networks are based on the well-known GN benchmark network[39], which consists of two types of dynamic networks: SYN-FIX and SYN-VAR. The GN static network consists of 128 vertices grouped into 4 communities where each of them contains 32 vertices, where each vertex has an average degree 16 and shares z edges connecting vertices outside of the cor-

responding community. SYN-FIX starts from a GN network as G_1. And, for each time t, 3 vertices are randomly selected from each community in G_{t-1} and assigned to others in G_t. Thus, the number of communities in the SYN-FIX network is 4 for all of the time steps. SYN-VAR is a modified version of SYN-FIX by forming and dissolving communities. Specifically, the initial GN network has 256 vertices with average degree 16, which are classified into four communities with 64 vertices in each. To introduce dynamics of networks, 8 vertices from each community are randomly selected in G_{t-1} and forms them into a novel community in G_t. This procedure is repeated for 5 time steps, then the vertices are returned to the original communities.

Priori to validating the performances of algorithms, how to select values for parameters are investigated. Three parameters are involved in *MLjFE*, where k is the number of features, α and β control the importance of multi-label learning term and temporal smoothness term. We investigate the impact of parameter k, α, and β by fixing the others.

First, we investigate how parameter α affect the performance of the proposed algorithm. By fixing β=1 and k=4, we testify the performance of *MLjFE* on the SYN-FIX networks by ranging α from 0 to 10 with gap 1. How AUC of *MLjFE* changes with various values of parameter α on the SYN-FIX networks is shown in Fig. 2 A. It is easy to conclude that, as parameter α increases from 0 to 2, AUC of *MLjFE* soars to 0.675. Furthermore, as parameter α from 2 to 9, the performance of the proposed algorithm is stable. The possible reason is that the objective function is dominated by the cost of classification if parameter α is small. As parameter α increases from 0 to 2, feature extraction becomes more important, resulting in the high quality features that are critical for the improvement of temporal link prediction. When $\alpha > 2$, the performance of *MLjFE* is quite stable. There is a good reason to explain this phenomenon. When parameter α is small, the objective function of the proposed model gives the top priority to improve the accuracy of prediction, ignoring the quality of features. In this case, the features of vertices cannot fully the topological structure of networks, resulting in an undesirable performance. As parameter α increases,

MLjFE pays more attention to the feature extraction, which ameliorates the quality of features. α =2, feature extraction and link prediction reaches a good balance, achieving the best performance. To check whether parameter α is sensitive to measurements, how AP score of *MLjFE* changes by varying parameter α from 0 to 9 is shown in Fig. 2 B, where the similar tendency repeats, indicating that parameter α is not sensitive to measurements. Thus, we only employ AUC to investigate the effects of parameters. Then, we check whether parameter α is sensitive to networks by replacing SYN-FIX with SYN-VAR networks. The performance of *MLjFE* in terms of AUC and AP score is illustrated in Fig. 2 C and D, respectively. From these panels, we assert that *MLjFE* achieves the best performance when α=2. Therefore, we set α=2 in the following experiments.

One of the critical difference between temporal and static link prediction is the smoothness of temporality, and parameter β controls the importance of smoothness. Then, we validate how parameter β effects the performance of *MLjFE* by fixing α=2. The plot of AUC vs parameter β on various networks is shown in Fig. Fig. 2 E and F, where panel E is for the SYN-FIX networks and $\beta > 1$. *MLjFE* obtains the optimal performance when β=1. When β is small, the temporality of networks is neglected, where features only reflects the topological structure of each snapshot without dynamics of networks. In this case, the features of vertices fail to characterize temporal networks, leading to the low accuracy. When β=1, the temporality and topological structure reach a good balance, generating discriminative features that are suitable for temporal link predictions. Fig. 2 F also indicates that β=1 is a good choice. Therefore, we set β=1.

Finally, how parameter k affects the performance of *MLjFE* is analyzed. By fixing α=2 and β=1, we investigate how accuracy of *MLjFE* changes by ranging k from 1 to 10 for the SYN-FIX networks. As shown in Fig. 2 G, AUC of *MLjFE* improves as parameter k increases from 1 to 6, and, the performance of *MLjFE* is stable when $k \geq 7$. *MLjFE* obtains the best performance when k=9 ($\approx$ =7.0% of vertices in networks). To check whether parameter k is sensitive to the temporal networks. The procedure repeats by using the SYN-VAR networks

as parameter k increases from 10 to 180 as shown in Fig. 2 H, where k=20 ($\approx$ =7.8% of vertices in networks). Therefore, we suggest a reasonable interval for parameter k as $[5\%n, 10\%n]$, where n is the number of vertices in networks.

Before presenting the detailed performance of various on the prediction of temporal links, we give an illustrative example to demonstrate the superiority of the proposed algorithm by using the SYN-FIX networks as shown in Fig. 3, where panel A is the heatmap of truth-ground of links of SYN-FIX at T+1, B-D are the output of *MLjFE*, GCN-GAN, and GrNMF, respectively. There are four diagonal blocks in Fig. 3 A because there are four modules in the SYN-FIX networks. From Fig. 3, we assert that *MLjFE* is superior to the state-of-the-art methods since the output of *MLjFE* is close to the truth-ground. These panels imply that the joint learning strategy is promising for the temporal link prediction.

The accuracy of the compared algorithms for datasets SYN-FIX and SYN-VAR are summarized in Fig. 4. From these figures, we assert that the *MLjFE* algorithm outperforms the rest algorithms in terms of AUC and AP score on synthetic artificial networks. For experiments of algorithm comparing, we select the largest AUC and AP score of all experiments as our result. The left two figures show the comparison about AUC, and *MLjFE* works slightly better than NMFFC on SYN-FIX and better than CPLST_PMI on SYN-VAR. Due to the lack of source codes about AP score in several algorithms, we only compare GrNMF, PMI, CPLST_PMI, and *MLjFE* in terms of AP score. The right two figures shows that *MLjFE* works better than other algorithm on both SYN-FIX and SYN-VAR datasets.

The Katz method achieves the worst performance because it collapses networks without preserving the topological structure of networks. There are three possible reasons why the proposed algorithm outperforms others. First, the multi-label learning is promising for temporal link prediction. For example, CPLST_PMI achieves an excellent performance. Second, joint multi-label learning and matrix factorization have a good balance between feature extraction and temporal link prediction. This assertion is consist with the conclusion in Ref.

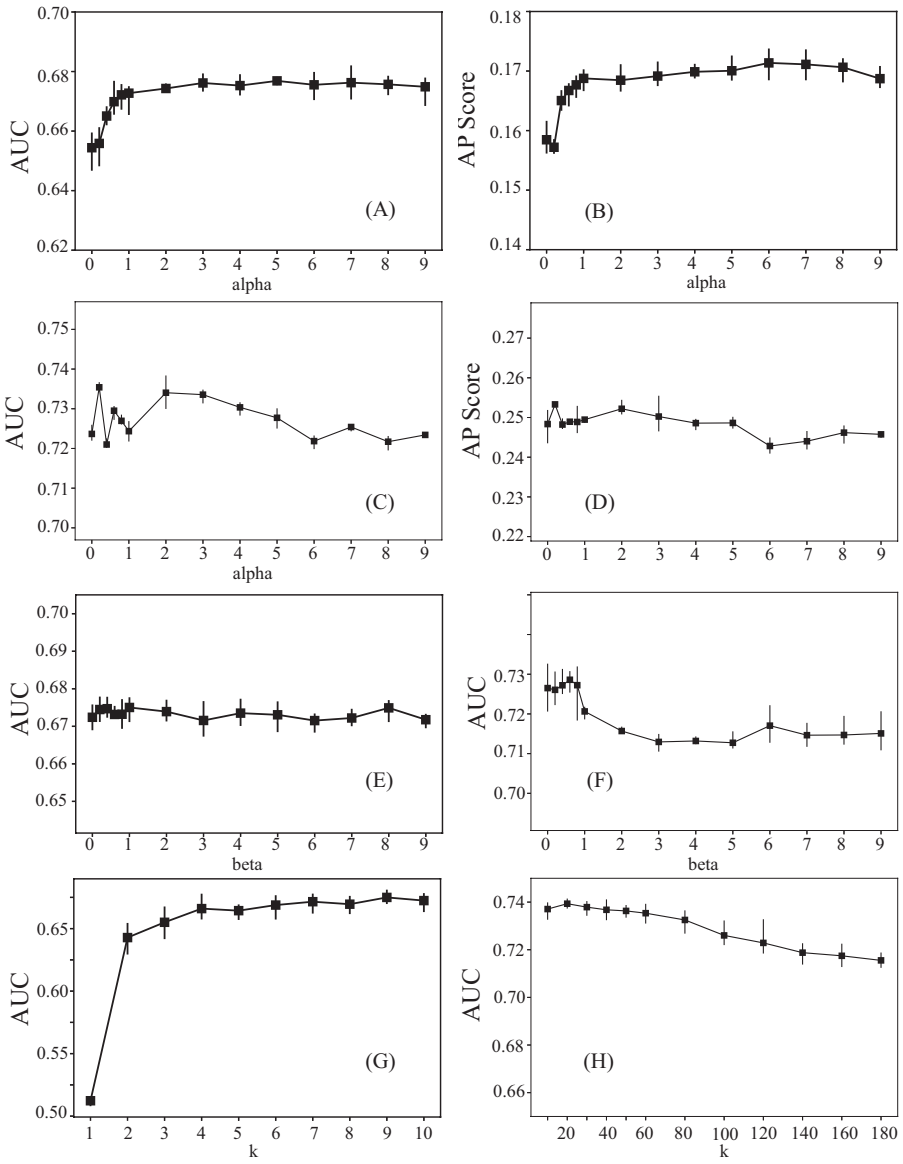

Figure 2: How parameters effect the performance of *MLjFE* on the artificial networks in terms of various measurements: (A) AUC vs α on the SYN-FIX networks, (B) AP score vs α on the SYN-FIX networks, (C) AUC vs α on the SYN-VAR networks, (D) AP score vs α on the SYN-VAR networks, (E) AUC vs β on the SYN-FIX networks, (F) AUC vs β on the SYN-VAR networks, (G) AUC vs k on the SYN-FIX networks, (H) AUC vs k on the SYN-VAR networks.

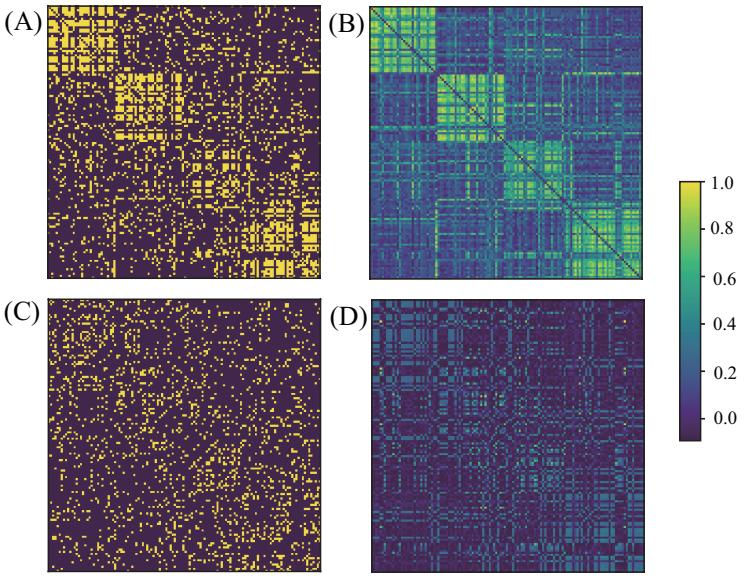

Figure 3: Heatmaps of various algorithms for temporal link prediction on the SYN-FIX networks: (A) Truth-ground, (B) *MLjFE*, (C) GCN-GAN, and (D)GrNMF.

[21] where joint learning models and parameters can significantly improve the accuracy of algorithms. The third reason is that the temporality of networks is regularized for feature extraction, which is more accurate to capture the dynamicity of networks. However, SNMF-FC independently extracts feature for each G_t, which is inferior to *MLjFE*. These results demonstrate that the proposed algorithm is promising for temporal link prediction.

5.3. Facebook: social temporal networks

The artificial networks in the previous subsection is insufficient to fully validate the performance of *MLjFE*. Thus, we check whether the proposed algorithm is also promising for predicting temporal links in social networks. The facebook social networks, named Swarthmore42 [1], is selected, where each vertex corresponds to an individual and each edge represents friendship relation [40]. Each snapshot contains the interactions of a month.

[1] http://networkrepository.com

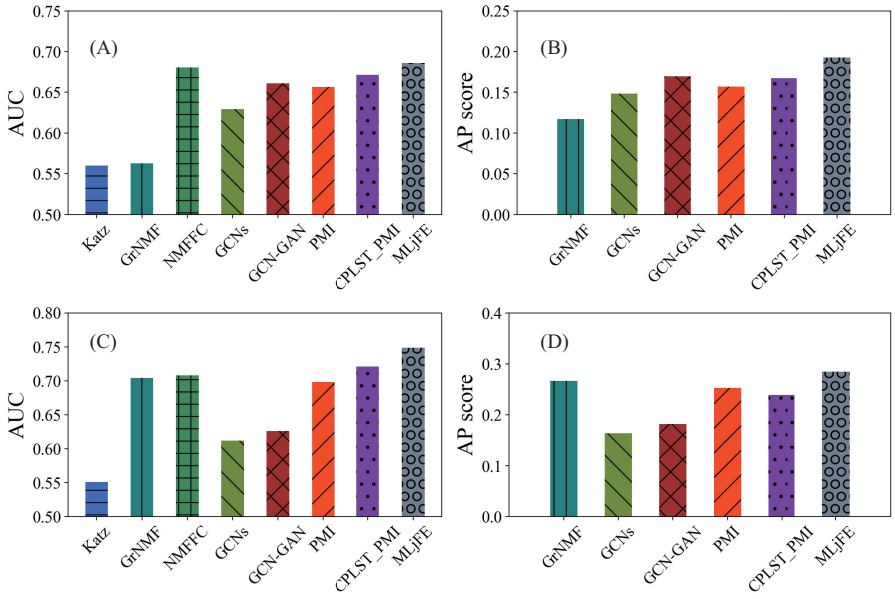

Figure 4: Performance of various algorithms on different networks in terms of various measurements: (A) AUC on the SYN-FIX networks, (B) AP score on the SYN-FIX networks, (C) AUC on the SYN-VAR networks, and (D) AP score on the SYN-VAR networks.

The performance of the compared algorithms on the facebook networks is shown in Fig. 5, where panel A is for AUC and B for AP score. From Fig. 5, we conclude that the proposed algorithm are superior to the other methods. In details, the AUC of *MLjFE* is 0.719, while AUCs are 0.665, 0.692, 0.679, 0.670, 0.655, and 0.675 for Katz, GrNMF, NMF-FC, SVD, PMI, CPLST_PMI, respectively. Moreover, the AP score of *MLjFE* is 0.454, which is significant higher than others, i.e. 0.141 (GrNMF), PMI (0.267), 0.218 (CPLST_PMI). *MLjFE* achieves the best performance, followed by GrNMF, implying that the latent features obtained by matrix factorization are precise to capture the dynamics of networks. These results demonstrate that the proposed algorithm is also promising for temporal link prediction in social networks.

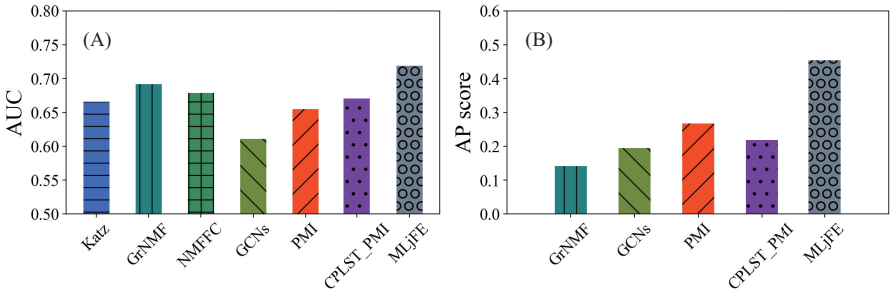

Figure 5: Performance of various algorithms on the Facebook social networks in terms of different measurements: (A) AUC, and (B) AP score.

5.4. CollegeMsg: message temporal networks

The third temporal networks [2] are derived from the University of California (Irvine), where each vertex denotes an individual and an edge represents there is at least on private message between a pair of vertices. There are in total 1828 users, 56912 edges and 6 time steps, where each time step corresponds to a month.

The accuracy of various algorithms on ColledgeMsg temporal networks is shown in Fig. 6, where panel A corresponds to AUCs and B to AP scores. We can conclude that *MLjFE* achieves the best performance, followed by NMF-FC and CPLST_PMI. Interestingly, although SNMF-FC independently extracts features for each time, it still has an excellent performance since the AUC of SNMF-FC is significantly higher than Katz, GrNMF, SVD and PMI. The possible reason is that the dynamics of ColledgeMsg temporal networks is not as fierce as Facebook temporal networks because the users in the same college have similar backgrounds. The AP score of *MLjFE* is also significant higher than others, demonstrating that the proposed algorithm precisely predicts the temporal links in social networks.

[2] http://snap.stanford.edu/data/CollegeMsg.html

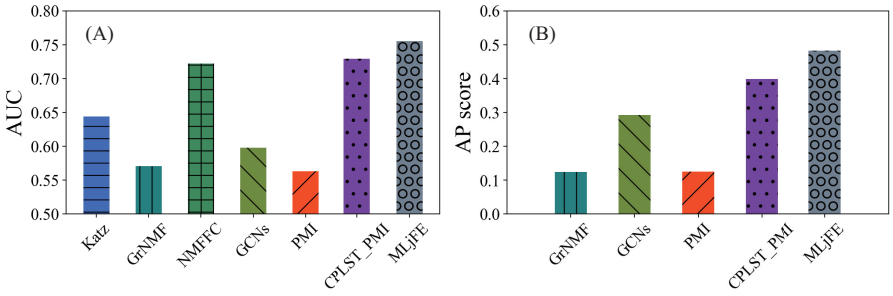

Figure 6: Performance of various algorithms on the CollegeMsg dataset in terms of different measurement: (A) AUC, and (B) AP score.

5.5. *Sx-mathoverflow*

The previous temporal networks are social networks for individuals. To fully validate performance of algorithms, we adopt the web temporal networks for stack exchange on *Math Overflow*, where each vertex corresponds to a user and each edge denotes a comment or answer between a pair of users. There are totally 3148 users and 62012 edges and 7 time steps [41].

The AUCs and AP scores of various algorithms on the web temporal networks are shown in Fig. 7, where MEjFE obviously outperforms the other algorithms in terms of both AUCs and AP scores. In details, the AUC of *MLjFE* is 0.755, while those of CPLST_PMI and SNMF-FC are 0.729 and 0.722, respectively. And, the AP score of *MLjFE* is 0.482, which improves 17.4% for CPLST_PMI. The results further demonstrate that joint multi-label learning and feature extraction can effectively capture the dynamics of networks.

5.6. *Email-Eu-core*

The email-EU-core network [3] is generated using email data from a large European research institution. The e-mails only represent communication between institution members (the core), so the network contains 986 nodes and 332334 edges with 803 days time span. We select 8 snapshots from 1970-01 to 1970-08 to construct the temporal networks.

[3] http://snap.stanford.edu/data/email-Eu-core-temporal.html

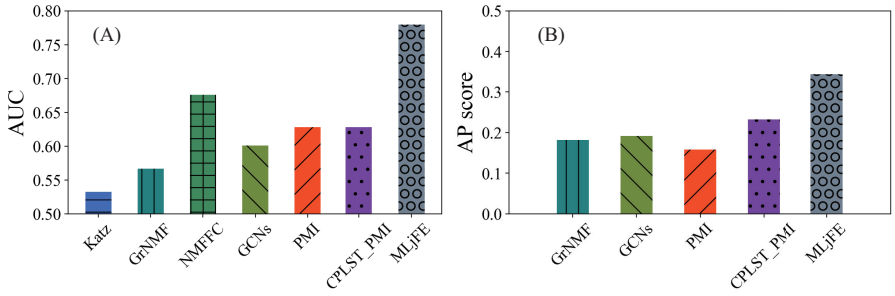

Figure 7: Performance of various algorithms on the sx-mathoverflow dataset in terms of different measurements: (A) AUC, and (B) AP score.

The performance of various algorithms on the email-EU-Core networks is shown in Fig. 8. From panel A, it is easy to assert that all the algorithms, except Katz, achieve an excellent performance and *MLjFE* is the best in terms of AUC. The possible reason why Katz is worst is that topological information cannot fully characterize dynamics of networks, while the latent features are more accurate for describe the structure of temporal networks. Fig.8 B demonstrates that *MLjFE* is inferior to GrNMF and CPLST_PMI. One of the possible reasons is that the Email temporal networks are much denser, i.e. the average degree of vertices is 65, which is 8 times higher than that of the other temporal networks. In all, the performance of *MLjFE* is also acceptable.

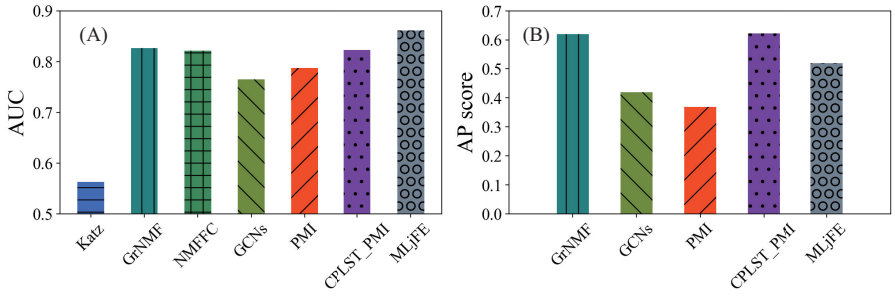

Figure 8: Comparison of various algorithms on the Email-EU-Core networks in terms of different measurements: (A) AUC, and (B) AP score.

5.7. DBLP networks and sx-askubuntu

The sizes of the previous networks are too small (≤ 5000) to fully evaluate the performance of the *MLjFE* algorithm. Therefore, we adapt the large-scale DBLP dataset[4] and sx-askubuntu, which have nodes 31,855 and 28,371. Each vertex on DBLP corresponds to an author and each link denotes co-author relationship between authors. Each vertex on sx-askubuntu corresponds to a user and each edge denote a comment or answer between a pair of users.

Due to the large number of nodes, only algorithms without networks collapsing could be adapted in large-scale datasets, like DBLP and sx-askubuntu. So, we only compare *MLjFE* with GrNMF, shown in Fig. 9.

Although DBLP and sx-askubuntu are both low degree networks with large number of nodes, our algorithms works well in these datasets. This is because feature extraction term is good at dealing with sparse networks, like DBLP and sx-askubuntu. In conclusion, we could tell that our proposed method works well in both artificial networks, real-world networks, and large-scale networks.

5.8. Performance on static link prediction

In addition, we testify the possibility of extending *MLjFE* for the static link prediction, where the temporal networks are collapsed into a static one. The 5-fold cross validation strategy is employed to validate the performance of *MLjFE*. We remove the temporality item in the objective function in Eq.(20). Thus, six static networks are generated for SYN-FIX, SYN-VAR, facebook, email-EU-core, CollegeMsg, and sx-mathoverflow. Six algorithms, including Katz, NMF, GCNs, DeepWalk, node2vec, and *MLjFE_static*, are selected for a comparison. The results of various algorithms are shown in Figs. 10. From these panels, we assert that DeepWalk achieves the best performance, whereas GCNs has the worst performance. *MLjFE* is inferior to DeepWalk, and superior to the others. These results demonstrate that *MLjFE* is also promising for static link prediction.

[4] http://wwww.informatik.uni-trier.ed/~ley/db/index.html

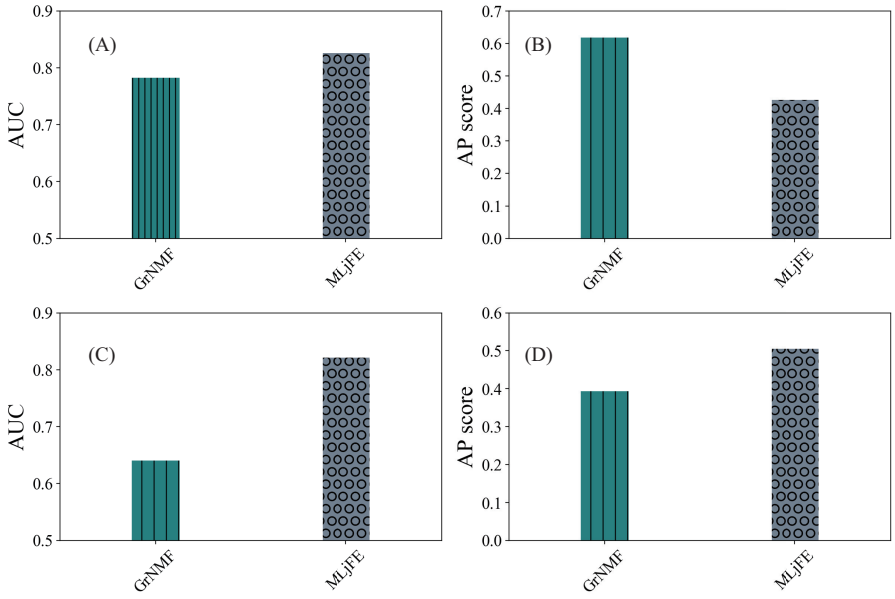

Figure 9: Comparison of various algorithms on the large-scale real-world networks in term of different measurements: (A) AUC on the DBLP networks, (B) AP score on the DBLP networks, (C) AUC on the sx-askubuntu networks, and (D) AP score on the sx-askubuntu networks.

6. Conclusion

Temporal link prediction is of great significance since the incomplete of observation of complex networks. Compared with the traditional link prediction in static networks, temporal link prediction takes into account both topological structure and temporality of networks, imposing a great challenge on designing effective algorithms. Although great efforts have been devoted to this issue, many unsolved problems, such as time complexity and accuracy of algorithms, remain. For example, the available algorithms assume the independence of feature extraction and prediction, resulting in an undesirable performance.

In this study, a novel joint learning-based algorithm (*MLjFE*) is proposed, where the feature of vertices and temporality are jointly learned. *MLjFE* consists of two major components, i.e., feature extraction and prediction. Specifically, the multi-label learning serves as feature extraction and link prediction, and temporal smoothing is employed to address the temporality of networks.

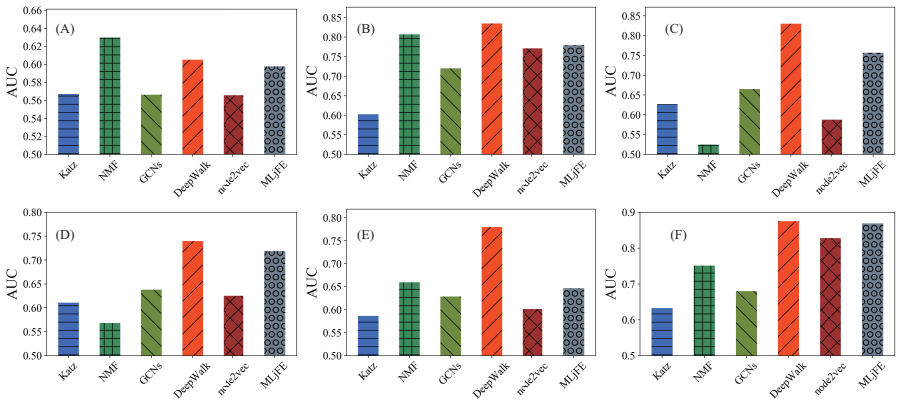

Figure 10: Comparison of static link prediction algorithms on static networks in term of AUC score: (A) SYNFIX (B) SYNVAR, (C) facebook (D) CollegeMsg, (E) sx-mathoverflow, (F) Email-Eu-Core.

Experimental results on the artificial and real-world temporal networks demonstrate that the proposed model outperforms state-of-the-art approaches. Different from the current methods, *MLjFE* has noteworthy advantages. First, the equivalence relation between graph embedding and matrix factorization is proved, laying the solid foundation for the proposed algorithm. Second, *MLjFE* makes use of the latent features of temporal networks based on the matrix factorization, which is more accurate to characterize the structure of networks. Finally, the features extraction and link prediction are jointly learned, improving the discriminative ability of features since the features are under the guidance of link prediction. Extensive experiments demonstrate that *MLjFE* is superior to state-of-the-art methods on various datasets. What we want to point out is that MFjFE provides a flexible joint learning framework for temporal link prediction, which can be easily extended to various applications.

Although the proposed model and algorithm overcomes several drawbacks of current methods. There are still some unsolved problems involving the complexity, accuracy and quantification of patterns, which are summarized as

- Even though the time complexity of *MLjFE* is the same as that of NMF and accelerated by PGD with automatically step size searching. However, it is still unacceptable for the large-scale networks with millions of vertices.

The possible alternative is network reduction, which aims to reduce the size of networks by preserving the topological structure of networks. However, it is non-trivial to perform network reduction on temporal networks since temporality and topological structure are difficult to balance. How to reduce sizes of networks without destroying the temporality of vertices is promising for addressing the complexity of algorithms.

- Currently, the clear definition of dynamic patterns in temporal networks is lack. The available algorithms, including the proposed algorithm, implicitly address the dynamic features by utilizing the smoothing framework to balance structure and temporality. How to describe and quantification of dynamic patterns in temporal paves a way to design alternatives for temporal link prediction.

- The application of temporal link prediction is also an interesting topic. There are so many dynamic systems, such as progression of cancers. How to apply the proposed algorithm to the cancer temporal networks and identify potential dynamic patterns are critical for revealing the underlying mechanisms of cancers. However, the interpretability of dynamic patterns for cancer is also highly non-trivial. Currently, *MLjFE* ignores the underlying backgrounds of temporal networks. How to incorporate priori information into algorithms is one of the possible strategies for this issue.

Acknowledgement

This work was supported by the NSFC (Grant No. 61772394), Key Research and Development Program of Shaanxi (Program No. 2021ZDLGY02-02), and open funding of State Key Laboratory for Novel Software Technology, China (Grant No. KFKT2020B14)..

References

References

[1] D. Watts, P. Dodds, M. Newman, Identity and search in social networks, Science 296 (2002) 1302–1305.

[2] X. Ma, L. Gao, K. Tan, Modeling disease progression using dynamics of pathway connectivity, Bioinformatics 30 (16) (2014) 2343–2350.

[3] A. Nishi, H. Shirado, D. Rand, N. Christakis, Inequality and visibility of wealth experimental social networks, Nature 526 (2015) 426–429.

[4] P. Cramer, Organization and regulation of gene transcription, Nature 573 (2019) 45–54.

[5] J. Banavar, A. Maritan, A. Rinaldo, Size and form in efficient transportation networks, Nature 399 (1999) 130–132.

[6] S. Fraiberǵer, R. Sinatra, M. Resch, et al., Quantifying reputation and success in art, Science 362 (6416) (2018) 825–829.

[7] X. Ma, L. Gao, Predicting protein complexes in protein interaction networks using a core-attachment algorithm based on graph communicability, Information Sciences 189 (2012) 233–254.

[8] X. Ma, D. Dong, Q. Wang, Community detection in multi-layer networks using joint nonnegative matrix factorization, IEEE Transactions Knowledge and Data Engineering 31 (2) (2019) 273–286.

[9] M. Stumpf, T. Thorne, E. Silva, et al., Estimating the size of the human interactome, PNAS 105 (19) (2008) 6959–C6964.

[10] A. Clauset, C. Moore, M. Newman, Hierarchical structure and the prediction of missing links in networks, Nature 453 (2008) 98–101.

[11] J. Zhao, L. Miao, J. Yang, et al., Prediction of links and weights in networks by reliable routes, Scientific Reports 5 (2015) 12261.

[12] Z. Cao, L. Wang, G. de Melo, Link prediction via subgraph embedding-based convex matrix completion, in: Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 2803–2810.

[13] L. L*ü*, C. Jin, T. Zhou, Predicting missing links via local information, Physical Review E 30 (2009) 046122.

[14] X. Cao, H. Chen, X. Wang, et al., Neural link prediction over aligned networks, in: Proceedings of Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 249–256.

[15] G. Palla, A. Barab*á*si, T. Vicsek, Quantifying social group evolution, Nature 446 (2007) 664–667.

[16] S. Haghani, M. Keyvanpour, A systemic analysis of link prediction in social network, Artificial Intelligence Review 52 (3) (2019) 1961–1995.

[17] X. Ma, P. Sun, G. Qin, Nonnegative matrix factorization algorithms for link prediction in temporal networks using graph communicability, Pattern Recognition 71 (2017) 361–374.

[18] D. Dunlavy, T. Kolda, E. Acar, Temporal link prediction using matrix and tensor factorizations, ACM Transactions on Knowledge Discovery from Data 5 (2) (2011) 10.

[19] L. Zhu, D. Guo, J. Yin, et al., Scalable temporal latent space inference for link prediction in dynamic social networks, IEEE Transactions on Knowledge Data Engineering 28 (10) (2016) 2765–2777.

[20] X. Ma, P. Sun, Y. Wang, Graph regularized nonnegative matrix factorization for temporal link prediction in dynamic networks, Physica A 496 (2018) 121–136.

[21] H. Lakkaraju, S. Bach, J. Leskovec, Interpretable decision sets: a joint framework for description and prediction, in: Proceedings of ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 1675–1684.

[22] Y. Zhao, L. Li, X. Wu, Link prediction-based multi-label classification on networked data, in: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2016, pp. 61–68.

[23] D. Liben-Nowell, J. M. Kleinberg, The link prediction problem for social networks, in: CIKM, 2003, pp. 556–559.

[24] U. Sharan, J. Neville, Temporal-relational classifiers for prediction in evolving domains, in: ICDM, 2008, pp. 540–549.

[25] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1) (1953) 39–43.

[26] X. Ma, L. Gao, X. Yong, Eigenspaces of networks reveal the overlapping and hierarchical community structure more precisely, Journal of Statistical Mechanics: Theory and Experiment 34 (2) (2013) 243–265.

[27] B. Perozzi, R. Al-Rfou, S. Skiena, Deepwalk: online learning of social representations, in: Conference on Knowledge Discovery and Data Mining, KDD, 2014, pp. 701–710.

[28] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, Q. Mei, LINE: large-scale information network embedding, in: Conference on World Wide Web, WWW, 2015, pp. 1067–1077.

[29] A. Grover, J. Leskovec, node2vec: Scalable feature learning for networks, in: Conference on Knowledge Discovery and Data Mining,KDD, 2016, pp. 855–864.

[30] Z. Chen, M. Chen, K. Weinberger, W. Zhang, Marginalized denoising for link prediction and multi-label learning, in: Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2289–2296.

[31] H. Yu, P. Jain, P. Kar, I. S. Dhillon, Large-scale multi-label learning with missing labels, in: International Conference on Machine Learning, 2014, pp. 593–601.

[32] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.

[33] O. Levy, Y. Goldberg, Neural word embedding as implicit matrix factorization, in: Conference on Neural Information Processing Systems, 2014, pp. 2177–2185.

[34] C. Yang, Z. Liu, D. Zhao, M. Sun, E. Y. Chang, Network representation learning with rich text information, in: International Joint Conference on Artificial Intelligence, IJCAI, 2015, pp. 2111–2117.

[35] C. H. Q. Ding, T. Li, W. Peng, H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in: Proceedings of the Twelfth International Conference on Knowledge Discovery and Data Mining, 2006, pp. 126–135.

[36] C. Lin, Projected gradient methods for nonnegative matrix factorization, Neural Computation 19 (10) (2007) 2756–2779.

[37] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907.

[38] K. Lei, M. Qin, B. Bai, G. Zhang, M. Yang, Gcn-gan: A non-linear temporal link prediction model for weighted dynamic networks, in: IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, 2019, pp. 388–396.

[39] M. Kim, J. Han, A particle-and-density based evolutionary clustering method for dynamic networks, PVLDB 2 (1) (2009) 622–633.

[40] R. A. Rossi, N. K. Ahmed, The network data repository with interactive graph analytics and visualization, in: AAAI, 2015.

[41] A. Paranjape, A. 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.