BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 02/06 pp19–33
Volume 1, Number 1, March 2018
DOI: 10.26599/BDMA.2017.9020002

DEEPEYE: Link Prediction in Dynamic Networks Based on Non-negative Matrix Factorization

Nahla Mohamed Ahmed, Ling Chen*, Yulong Wang, Bin Li, Yun Li, and Wei Liu

Abstract: A Non-negative Matrix Factorization (NMF)-based method is proposed to solve the link prediction problem in dynamic graphs. The method learns latent features from the temporal and topological structure of a dynamic network and can obtain higher prediction results. We present novel iterative rules to construct matrix factors that carry important network features and prove the convergence and correctness of these algorithms. Finally, we demonstrate how latent NMF features can express network dynamics efficiently rather than by static representation, thereby yielding better performance. The amalgamation of time and structural information makes the method achieve prediction results that are more accurate. Empirical results on real-world networks show that the proposed algorithm can achieve higher accuracy prediction results in dynamic networks in comparison to other algorithms.

Key words: dynamic network; link prediction; matrix factorization

1 Introduction

With the widespread reach of the Internet, social networks have become popular and people can use them to build wider connections[1−4]. Besides social networks, network structures are constantly observed in real-world systems, such as road traffic networks[5] and neural networks. When we build a network structure to topologically approximate complex systems, missing or redundant links may unavoidably occur owing to time and cost restrictions in experiments conducted for constructing the networks. Moreover, since the network links may be dynamically changing in nature, some potential links may appear in the future. Therefore, it is necessary to detect such hidden links, or future links, from the topological structure of the current network. This is a task of link prediction in complex networks[6−8].

Link prediction plays a key role in solving many real-world problems. For instance, in biological networks like disease-gene networks, protein-protein interaction networks, and metabolic networks[9], links indicate the interaction relations between the organism and the diseases represented by the nodes they connect to. Although such implicit interaction relations can be discovered by a biological experiment, their excessive cost makes large-scale experiments impossible. To reduce the huge cost of biological experiments, link prediction can be employed at the preprocessing stage to discover the potential interaction relations in biological networks. In diseases-gene networks, link prediction can detect the hidden links between disease and gene to discover the cause of the disease and to discover a treatment and new drugs for the disease[10].

Over the last decade, link prediction has been applied to recommendation[11,12], author-paper subject prediction[13], scientific paper impact estimation[14], biological network analysis[15], and protein interaction prediction[16], among others. Recently, the technique

• Nahla Mohamed Ahmed, Ling Chen, Bin Li, Yun Li, and Wei Liu are with College of Information Engineering, Yangzhou University, Yangzhou 225009, China. E-mail: nahlaibrahim@uofk.edu; yzulchen@163.com.
• Nahla Mohamed Ahmed is also with College of Mathematical Sciences, Khartoum University, Khartoum, Sudan.
• Ling Chen and Bin Li are also with State Key Lab of Novel Software Tech, Nanjing University, Nanjing 210093, China.
• Yulong Wang is with College of Agriculture, Yangzhou University, Yangzhou 225009, China.
∗ To whom correspondence should be addressed.
Manuscript received: 2017-07-27; accepted: 2017-11-28

of link prediction has become a useful tool in analyzing relations in social networks. By analyzing social relations using link prediction, we can discover potential interpersonal links[6,17−19]. Link prediction can reveal the potential friendship of users in social networks and can recommend possible friends to the users[20]. Link prediction has been exploited in monitoring criminal networks for detecting hidden connections between members of criminal organizations in order to stop their malefaction. Link prediction is often employed in analyzing author networks in scientific publications and to predict potential co-authors[21]. In e-commerce, merchants often use link prediction in order to recommend a commodity to the customers[12,22]. Such recommendation enables marketers to introduce their products to customers[23,24]. Link prediction can be used to analyze email communications and detect anomalous e-mails.

In addition to its important application value, link prediction also plays a significant role in theoretical investigations. For instance, link prediction research provides theoretical assistance to the study of complex network evolution mechanism[25,26]. For a given network, many models have been proposed in order to provide a possible network evolution mechanism. Due to the large number of statistics that characterize a network structure, it is hard to compare the precisions of various mechanisms. Link prediction can offer an easy, uniform, and relatively fair approach toward comparing the mechanisms of network evolution. It can greatly aid the theoretical investigation of the complex network evolution model.

With the development in network analysis research, there have been many link prediction methods reported recently. These methods can be classified into three main categories: similarity-based, probabilistic model-based, and machine learning-based methods.

The easiest and most widely used link prediction method is the similarity-based method. Many studies have shown that nodes with more similar topological features are more likely to be linked to each other. For instance, Aiello et al.[27] found that people with common interests and similar personalities tend to become friends. To predict potential links, the similarity-based method computes an index for each node pair in the network. Such index reflexes the topological similarity level between the two nodes. We can compute the similarity score according to the main topological features of the network. If a pair of nodes has correlated topological structures or more common features, they should be assigned a higher similarity score[28,29]. The non-existing links that have larger similarities are considered more likely to occur. There are three types of similarity indexes: local indexes based on the local topological structure, global indexes based on global topological structure, and quasi-local indexes, which exploit more topological information than local indexes but less than global indexes[30]. Computing quasi-local and global indexes usually involves the random walk technique. Wang et al.[31] proposed a random walk based link prediction method in directed networks exploiting the information of directed paths. Das Sarma et al.[32] advanced a distributed random walk algorithm, which can be used for parallel link prediction. Liu and Lü[33] showed that the finite steps of a random walk can achieve more precise prediction results, in comparison to the global random walk. Since random walk methods demand large amount of computation time, they are not applicable to solving many large-scale problems.

Probabilistic models are often used in some network link prediction methods. To predict individual relationships in a social network, Hu and Wong[34] suggested a probabilistic model; the model was optimized by genetic algorithm. To detect the hidden links in directed networks, Barbieri et al.[35] presented a stochastic topic model that can explain the reason why each link has been predicted. To compute the likelihood of the potential links, a probabilistic model was presented by Gao et al.[36] The model utilized information in the network from a variety of aspects.

Some methods for link prediction employ machine learning techniques. Based on a semi-supervised learning strategy, Zeng et al.[37] advanced a link prediction method that exploits the latent topological information in networks. Other machine learning techniques, such as supervised rank aggregation[38], principal component analysis[39], and ensemble[40], are also employed for network link prediction. To reduce the time cost of optimization, heuristic optimization approaches, such as evolutionary algorithm[41] and ant colony optimization[42], were also employed for network link prediction. To predict hidden links in large-scale networks, Chen et al.[43] presented a sampling based algorithm in order to predict the links involving a certain node. Ding et al.[44] presented a link prediction method according to community information in the network.

However, most methods aimed at link prediction in static networks, where the link connections do not change over time. In real-world applications, more systems are modeled as dynamic networks. In dynamic complex networks, temporal topology information is one of the main sources for designing the similarity function between entities. However, existing link prediction algorithms do not apply temporal information sufficiently. Therefore, it is necessary to study effective methods for predicting links in dynamic networks.

Recently, works for link prediction in dynamic networks have been reported. Ahmed et al.[1] presented an algorithm for predicting potential links in uncertain dynamic social networks. The algorithm constructs a deterministic network, where a random walk is performed in order to compute similarity scores between the nodes. A continuous-time regression model was presented by Vu et al.[45] for link prediction in dynamic networks. The model combines temporal regression coefficients with the time-dependent network statistics in order to improve the quality of prediction results.

In this paper, we propose a method for link prediction based on non-NMF in dynamic networks. New iterative rules are proposed in order to construct the matrix factors that carry important features of the network. We prove the convergence and correctness of the proposed non-NMF algorithm. Additionally, we show how latent NMF features can express network dynamics efficiently, instead of static representation, which yields better performances than using static representation. The amalgamation of time and structural information makes the method achieve prediction results with higher precision. Empirical results on real-world networks show that the proposed algorithm can achieve higher quality prediction results in dynamic networks, in comparison to other algorithms.

The remainder of this paper is structured as follows: Section 2 provides the problem definition and creates a Non-negative Matrix Factorization (NMF) model for dynamic networks. Section 3 proposes an iterative method for NMF, and introduces the framework of the algorithm. Section 4 provides the time complexity analysis of the algorithm, while Section 5 illustrates the convergence and correctness of our iterative method. Experimental results are shown and analyzed in Section 6. In Section 7, we conclude the paper.

2 Link Prediction and Non-negative Matrix Factorization

2.1 Link prediction

We use a set of undirected binary graphs to represent a given dynamic network. Let $V = \{v_1, v_2, ..., v_n\}$ be a set of vertices, and $G_{t_0}, G_{t_0+1}, ..., G_{t_0+T-1}$ be the networks appearing at different times $t = t_0, t_0 + 1, ..., t_0 + T - 1$. We define a list of symmetric matrices $A_{t_0}, A_{t_0+1}, ..., A_{t_0+T-1}$ to be the adjacency network matrices $G_{t_0}, G_{t_0+1}, ..., G_{t_0+T-1}$. The element $A_t(i, j)$ in the adjacent matrix A implies the occurrence of a link between vertexes v_i and $v_j, i, j = 1, 2, ..., n$, during the time period $t = t_0,\ t_0 + 1, ...,\ t_0 + T - 1$. Given the graph series $G_{t_0},\ G_{t_0+1}, ...,\ G_{t_0+T-1}$, the goal of link prediction in a dynamic network is to forecast the appearance of the future links at time $t_0 + T$, where T is the time interval. Our link prediction method outputs an $n \times n$ matrix $S_{t_0,T}$ with each element $S_{t_0,T}(i, j)$ as a similarity score indicating the likelihood of the link (v_i, v_j) at time $t_0 + T$.

2.2 Non-negative matrix factorization

Let non-negative matrix$A$ be the $n \times n$ adjacent matrix of the network. The goal of the non-NMF is to find two non-negative matrices $U \in \mathbf{R}_+^{n\times k}$ and $V \in \mathbf{R}_+^{n\times k}$, so that their product will be very close to matrix $\mathbf{A}$:

$$A \approx UV^{\mathrm{T}}.$$

Here k is the dimension of the latent space ($k < n$). U consists of latent space bases and is called the base matrix. V represents the combination coefficients of the bases for reconstructing matrix A and is called the coefficient matrix. This matrix factorization problem can be modeled as a problem of optimizing the following Frobenius norm:

$$\min_{U,V} ||X - UV^{\mathbf{T}}||_{\mathrm{F}}^2, \text{s.t.} \quad U \geqslant 0, V \geqslant 0 \qquad (1)$$

Here, $||\cdot||_{\mathrm{F}}$ is the Frobenius norm, and constrains $U \geqslant 0$ and $V \geqslant 0$ require that all elements in matrices $U$and $V$are non-negative. The similarity between nodes i and j in the latent space can be represented by the similarity between the i-th and j-th row vectors in matrix V.

Let $\{A^{(1)}, A^{(2)}, \cdots, A^{(T)}\}$ be the adjacent matrix of the dynamic network at time from 1 to T, where $A^{(t)} \in \mathbf{R}_+^{n\times n}$ $(1 \leqslant t \leqslant T)$ is the adjacent matrix at time t. Suppose the adjacent matrix $A^{(t)}$ can be decomposed into $A^{(t)} \approx U^{(t)}(V^{(t)})^{\mathrm{T}}$ $(1 \leqslant t \leqslant T)$, where vector $V_j^{(t)}$ represents the coefficient of node

j in k-dimensional latent space represented by $U^{(t)}$. The values of $V^{(t)}$ are changing for different times t. In order to take exploit the topological information at different times, we effectively integrate the adjacent matrices in the low dimensional latent space. We attempt to find a consensus latent space matrix U^* and a coefficient matrix V^*, so that for every consensus latent space matrix $U^{(t)}$ and coefficient matrix $V^{(t)}$, the differences between V^* and $V^{(t)}$, U^* and $U^{(t)}$ will be minimized. In other words, the Frobenius norm of the differences between $V^{(t)}$ and V^*, U^* and $U^{(t)}$ will be minimized:

$$D(U^{(t)}, U^*, V^{(t)}, V^*) = ||V^{(t)} - V^*||_{\mathrm{F}}^2 + ||U^{(t)} - U^*||_{\mathrm{F}}^2 \tag{2}$$

Therefore, our goal was to obtain a proper $U^{(t)}$, $V^{(t)} (\forall 1 \leqslant t \leqslant T)$, and V^*, in order to minimize the following objective function J:

$$J = \sum_{t=1}^{T} \lambda^{T-t} ||A^{(t)} - U^{(t)}(V^{(t)})^{\mathrm{T}}||_{\mathrm{F}}^2 + \sum_{t=1}^{T} \lambda^{T-t} ||U^{(t)} - U^*||_{\mathrm{F}}^2 + \sum_{t=1}^{T} \lambda^{T-t} ||V^{(t)} - V^*||_{\mathrm{F}}^2 \tag{3}$$

$$\text{s.t. } \forall 1 \leqslant t \leqslant T, U^{(t)}, V^{(t)}, U^*, V^* \geqslant 0$$

Since the links in recent networks carry more reliable information than old networks, the attenuation coefficient λ in Eq. (3) is used to assign more weight to the recent network topological information. At each time, the weights of the nodes are decreased by the attenuation coefficient.

3 Iterative Method for NMF

To solve the optimization problem (24), we present an iterative method. Since $U^{(t)}$, $V^{(t)}$, U^*, and V^* are variables in Eq. (3), we fix three of the variables at each time in the iteration, and obtain the best value for the fourth variable, in order to minimize the objective function J. After setting the proper initial values for matrices V^*, $U*$, $U^{(t)}$, and $V^{(t)}$, for t=1, ..., T, each iteration consists of four steps:

Step 1. Fix matrices $V^{(t)}$, U^*, and V^*, update $U^{(t)}$ to minimize J;

Step 2. Fix matrices $U^{(t)}$, U^*, and V^*, update $V^{(t)}$ to minimize J;

Step 3. Fix matrices $U^{(t)}$, V^*, and $V^{(t)}$, update U^* to minimize J.

Step 4. Fix matrices $U^{(t)}$, U^*, and $V^{(t)}$, update V^* to minimize J.

3.1 Updating $U^{(t)}$

By fixing matrices $V^{(t)}$,V^*, and U^*, we update $U^{(t)}$ in order to minimize the objective function of $J(U^{(t)})$, t=1, ..., T:

$$J(U^{(t)}) = \lambda^{T-t} ||A^{(t)} - U^{(t)}(V^{(t)})^{\mathrm{T}}||_{\mathrm{F}}^2 + \lambda^{T-t} ||U^{(t)} - U^*||_{\mathrm{F}}^2 \tag{4}$$

Since

$$\frac{\partial J(U^{(t)})}{\partial U^{(t)}} = -\lambda^{T-t}[-2A^{(t)}V^{(t)} + 2U^{(t)}V^{(t)\mathrm{T}}V^{(t)} + 2U^{(t)} - 2U^*] \tag{5}$$

by the Karush–Kuhn–Tucker (KKT) condition, we know that

$$-2A^{(t)}V^{(t)} + 2U^{(t)}V^{(t)\mathrm{T}}V^{(t)} + 2U^{(t)} - 2U^* = 0 \tag{6}$$

By Eq. (6), we can obtain the following formula for updating $U_{ij}^{(t)}$:

$$u_{ij}^{(t)} \leftarrow u_{ij}^{(t)} \frac{(A^{(t)}V^{(t)})_{ij} + u_{ij}^*}{(U^{(t)}V^{(t)\mathrm{T}}V^{(t)})_{ij} + u_{ij}^{(t)}},$$
$$t = 1,\ 2,\ ...,\ T;\ i = 1,\ 2, ...,\ n;\ j=1, 2, ..., k \tag{7}$$

3.2 Updating $V^{(t)}$

By fixing matrices $U^{(t)}$, U^*, and V^*, we update $V^{(t)}$ in order to minimize the objective function of $J(V^{(t)})$:

$$J(V^{(t)}) = \lambda^{T-t} ||A^{(t)} - U^{(t)}(V^{(t)})^{\mathrm{T}}||_{\mathrm{F}}^2 + \lambda^{T-t} ||V^{(t)} - V^*||_{\mathrm{F}}^2 \tag{8}$$

Since

$$\frac{\partial J(V^{(t)})}{\partial V^{(t)}} = -\lambda^{T-t}[2U^{(t)T}U^{(t)}V^{(t)\mathrm{T}} - 2U^{(t)\mathrm{T}}A^{(t)} + 2V^{(t)} - 2V^*] \tag{9}$$

by the KKT condition, we know that

$$2U^{(t)\mathrm{T}}U^{(t)}V^{(t)\mathrm{T}} - 2U^{(t)\mathrm{T}}A^{(t)} + 2V^{(t)} - 2V^* = 0 \tag{10}$$

By Eq. (10), we can obtain the following formula for updating $v_{ij}^{(t)}$:

$$v_{ij}^{(t)} \leftarrow v_{ij}^{(t)} \frac{(U^{(t)\mathrm{T}}A^{(t)})_{ij}^{\mathrm{T}} + v_{ij}^*}{(U^{(t)\mathrm{T}}U^{(t)}V^{(t)\mathrm{T}})_{ij}^{\mathrm{T}} + v_{ij}^{(t)}},$$
$$t = 1, 2, ..., T;\ i = 1, 2, ..., k;\ j = 1, 2, ..., n \tag{11}$$

3.3 Updating U^*

By fixing matrices $U^{(t)}$, $V^{(t)}$, and V^*, we update U^* in order to minimize the objective function of $J(U^*)$:

$$J(U^*) = \sum_{t=1}^{T} \lambda^{T-t} ||A^{(t)} - U^{(t)}(V^{(t)})^{\mathrm{T}}||_{\mathrm{F}}^2 +$$

$$\sum_{t=1}^{T} \lambda^{T-t} ||U^{(t)} - U^*||_F^2 \tag{12}$$

Since

$$\frac{\partial J(U^*)}{\partial U^*} = \frac{\partial}{\partial U^*} \sum_{t=1}^{T} \lambda^{T-t} ||U^{(t)} - U^*||_F^2 =$$

$$\sum_{t=1}^{T} \lambda^{T-t} \frac{\partial}{\partial U^*} ||U^{(t)} - U^*||_F^2 = 2 \sum_{t=1}^{T} \lambda^{T-t} (U^{(t)} - U^*) \tag{13}$$

by the KKT condition, we know that

$$\sum_{t=1}^{T} \lambda^{T-t} \left(U^{(t)} - U^*\right) = 0 \tag{14}$$

By Eq. (14), we obtain the following formula for updating U^*:

$$U^* = \frac{1}{\sum_{t=1}^{T} \lambda^{T-t}} \sum_{t=1}^{T} \lambda^{T-t} U^{(t)} \tag{15}$$

3.4 Updating V^*

By fixing the matrices $U^{(t)}$, $V^{(t)}$, and U^*, we update V^* in order to minimize the objective function of $J(V^*)$:

$$J(V^*) = \sum_{t=1}^{T} \lambda^{T-t} ||A^{(t)} - U^{(t)} (V^{(t)})^{\mathrm{T}}||_F^2 + \sum_{t=1}^{T} \lambda^{T-t} ||V^{(t)} - V^*||_F^2 \tag{16}$$

Since

$$\frac{\partial J(V^*)}{\partial V^*} = \frac{\partial}{\partial V^*} \sum_{t=1}^{T} \lambda^{T-t} ||V^{(t)} - V^*||_F^2 =$$

$$\sum_{t=1}^{T} \lambda^{T-t} \frac{\partial}{\partial V^*} ||V^{(t)} - V^*||_F^2 = 2 \sum_{t=1}^{T} \lambda^{T-t} (V^{(t)} - V^*),$$

by the KKT condition, we know that

$$\sum_{t=1}^{T} \lambda^{T-t} (V^{(t)} - V^*) = 0 \tag{17}$$

By Eq. (17), we obtain the following formula for updating V^*:

$$V^* = \frac{1}{\sum_{t=1}^{T} \lambda^{T-t}} \sum_{t=1}^{T} \lambda^{T-t} V^{(t)} \tag{18}$$

3.5 Algorithm

The frame work of our NMF based algorithm for link prediction in dynamic networks is as shown in Algorithm 1.

In Algorithm 1, Step 3.1.1 fixes matrices $V^{(t)}$, U^*, and V^*, and updates $U^{(t)}$. Step 3.1.2 fixes matrices $U^{(t)}$, U^*, V^*, and updates $V^{(t)}$. Step 3.2 fixes matrices $U^{(t)}$, V^*, $V^{(t)}$, and updates U^*. Step 3.3 fixes matrices $U^{(t)}$, U^*, $V^{(t)}$, and updates V^*. In Step 4, the similarities between the row vectors in V^* are computed and stored in matrix S as the output score of link prediction. Similarity measures, such as correlation coefficient and cosine similarity, can be used as similarity scores.

Algorithm 1 MF (NMF based Link Prediction)

Input :
$\{A^{(1)}, A^{(2)}, \cdots, A^{(T)}\}$: adjacent matrices of the dynamic network in times $1, \ldots, T$;
K: dimension of the latent space;
λ: The damping factor;
Output:
$\{U^{(1)}, U^{(2)}, \cdots, U^{(T)}\}$: latent space matrices;
$\{V^{(1)}, V^{(2)}, \cdots, V^{(T)}\}$: coefficient matrices;
V^* : integrated latent space matrices;
$U*$: integrated latent space matrices;
S: similarity score matrix of link prediction;
Begin
1. Initialize the matrices U^*, V^*, $U^{(t)}$ and $V^{(t)}$ for $t = 1, \ldots, T$;
2. $M = \sum_{t=1}^{T} \lambda^{T-t}$;
3. While not convergence **do**
  3.1 for t=1 **to** T **do**
    3.1.1 compute $A_V^{(t)} = A^{(t)} V^{(t)}, \widetilde{A} V^{(t)} = U^{(t)} V^{(t)\mathrm{T}} V^{(t)}$;
    for each element $u_{ij}^{(t)}$ in $U^{(t)}$ **do**
      $u_{ij}^{(t)} \leftarrow u_{ij}^{(t)} \frac{(A_V^{(t)})_{ij} + u_{ij}^*}{(\widetilde{A}_V^{(t)})_{ij} + u_{ij}^{(t)}}$;
    End for;
    3.1.2 compute $A_U^{(t)} = U^{(t)\mathrm{T}} A^{(t)}, \widetilde{A}_U^{(t)} = U^{(t)\mathrm{T}} U^{(t)} V^{(t)\mathrm{T}}$;
    for each element $v_{ij}^{(t)}$ in $V^{(t)}$ **do**
      $v_{ij}^{(t)} \leftarrow v_{ij}^{(t)} \frac{(A_U^{(t)})_{ij}^{\mathrm{T}} + v_{ij}^*}{(\widetilde{A}_U^{(t)})_{ij}^{\mathrm{T}} + v_{ij}^{(t)}}$;
    End for;
  End for t;
  3.2 $U^* = \frac{1}{M} \sum_{t=1}^{T} \lambda^{T-t} U^{(t)}$;
  3.3 $V^* = \frac{1}{M} \sum_{t=1}^{T} \lambda^{T-t} V^{(t)}$;
  End while;
4. For i=1 **to** n **do**
    For j = 1 **to** n **do**
      $S(i, j)$=similarity between the i-th and j-th row vectors in V^*;
    End for i
  End for j;
5. Output(S)
End

4 Time Complexity Analysis

In each iteration, Steps 3.1.1 and 3.1.2 require $O(n^2 \cdot k)$ time for matrix multiplications. Steps 3.2 and 3.3 require $O(t.n.k)$ time for the summation of the matrices, where t is the number of iterations. Since each row in matrix V^* is a k-dimensional vector, it takes $O(k)$ time to compute the similarity between such vectors. Therefore, Step 4 requires $O(n^2.k)$ time to compute the similarities for all pairs of the row vectors in V^*. Since k and t can be treated as constants, the complexity of the algorithm is $O(n^2)$. Since there are $\frac{n(n-1)}{2}$ node pairs in the network, similarity-based link prediction methods require at least $O(n^2)$ time. Therefore, algorithm MF reaches the lower bound of time complexity for a similarity-based link prediction method.

5 Convergence and Correctness Analysis

In this section, the convergence of the iterative formulas for updating U, V, U^*, and V^* is proven, and the correctness of converged solutions is verified. First, we give the following lemmas.

Lemma 1 Let $F \in \mathbf{R}_+^{n\times n}$, $G \in \mathbf{R}_+^{k\times k}$ be symmetric matrices, and $S \in \mathbf{R}_+^{n\times k}$, $S' \in \mathbf{R}_+^{n\times k}$ be two $n \times k$ matrices, then, the following inequality holds.

$$\sum_{i=1}^{n}\sum_{j=1}^{k}\frac{(FS'G)_{ij}S_{ij}^2}{S'_{ij}} \geqslant \mathrm{tr}[SF^{\mathrm{T}}SG] \tag{19}$$

Proof Let $S_{ij} = S'_{ij}p_{ij}$. By using the explicit index, the difference between LHS and RHS is

$$\Delta = \sum_{i,x=1}^{n}\sum_{j,y=1}^{k} F_{ix}S'_{xy}G_{yj}S'_{ij}(p_{ij}^2 - p_{ij}p_{xy}).$$

Since F and G are symmetric matrices, this is equal to

$$\Delta = \sum_{i,x=1}^{n}\sum_{j,y=1}^{k} F_{ix}S'_{xy}G_{yj}S'_{ij}\left(\frac{p_{ij}^2+p_{xy}^2}{2} - p_{ij}p_{xy}\right) = \frac{1}{2}\sum_{i,x=1}^{n}\sum_{j,y=1}^{k} F_{ix}S'_{xy}G_{yj}S'_{ij}(p_{ij}-p_{xy})^2 \geqslant 0$$

Q.E.D.

Lemma 2 Let $L(H)$ be a function of matrix H. $Z(H,\tilde{H})$ is a function for $L(H)$ which satisfies $Z(H,\tilde{H}) \geqslant L(H)$ and $Z(H,H) = L(H)$ for any H and $\tilde{H}$.

If the following iteration is performed in order to obtain a sequence of $H(\tau)$:

$$H(\tau+1) = \arg\min_H Z\,(H, H(\tau)) \tag{20}$$

Then $L(H(\tau))$ is non-increasing.

Proof By Lemma 2, we have

$$L(H(\tau+1)) \leqslant Z(H(\tau+1),$$
$$H(\tau)) \leqslant Z(H(\tau), H(\tau)) = L(H(\tau)).$$

Q.E.D

Theorem 1 By fixing any three matrices in $U^{(t)}$, $V^{(t)}$,V^*, and U^*, and by using the updating rules (7), (11), (15), and (18) in each iteration of algorithm MF, the value of the objective function$J$ is non-increasing.

Proof When we fix matrices $V^{(t)}$ and U^*, in order to update $U^{(t)}$, updating the formula for $U^{(t)}$ is derived by minimizing the objective function of $J(U^{(t)})$ given in Eq. (4). Then, $J(U^{(t)})$ can be rewritten as

$$J(U^{(t)}) = \mathrm{tr}[\lambda^{T-t}(A^{(t)} - U^{(t)}V^{(t)\mathrm{T}})(A^{(t)} - U^{(t)}V^{(t)\mathrm{T}})^{\mathrm{T}} + \lambda^{T-t}(U^{(t)} - U^*)(U^{(t)} - U^*)^{\mathrm{T}}].$$

It is easy to obtain

$$J(U^{(t)}) = \lambda^{T-t}\mathrm{tr}[A^{(t)\mathrm{T}}A^{(t)} - 2U^{(t)\mathrm{T}}A^{(t)}V^{(t)} + V^{(t)\mathrm{T}}V^{(t)}U^{(t)\mathrm{T}}U^{(t)} - 2U^{(t)\mathrm{T}}U^* + U^{*\mathrm{T}}U^*] \tag{21}$$

By ignoring constant terms with respect to $U^{(t)}$, Eq. (21) can be written as

$$L(U^{(t)}) = \mathrm{tr}[-2U^{(t)\mathrm{T}}(A^{(t)}V^{(t)} + U^*) + (V^{(t)\mathrm{T}}V^{(t)} + I)U^{(t)\mathrm{T}}U^{(t)}] \tag{22}$$

Let

$$Z(U^{(t)}, \tilde{U}^{(t)}) = -2\sum_{ij}((A^{(t)}V^{(t)})_{ij} + U_{ij}^*)U_{ij}^{(t)} + \sum_{ij}\frac{(\tilde{U}^{(t)}(V^{(t)\mathrm{T}}V^{(t)} + I))_{ij}U_{ij}^{(t)2}}{\tilde{U}_{ij}^{(t)}} \tag{23}$$

Then $Z\left(U^{(t)}, \tilde{U}^{(t)}\right)$ is a function for $L(U^{(t)})$. First, when $\tilde{U}^{(t)} = U^{(t)}$, it is obvious that $Z\left(U^{(t)}, \tilde{U}^{(t)}\right) = L(U^{(t)})$. By Lemma 1, the second term in Eq. (22) is always less than that in Eq. (23); thus, inequality $Z(U, U') \geqslant L(U)$ holds. Therefore, $Z(U^{(t)}, \tilde{U}^{(t)})$ is an auxiliary function for $L(U^{(t)})$.

By the definition of the function in Lemma 2, we know that if we find that the $U^{(t)}$ value reaches the local minimum of function $Z(U^{(t)}, \tilde{U}^{(t)})$, then the value of $L(U^{(t)})$ is non-increasing over t; therefore, we find the $U^{(t)}$value to minimize $Z(U^{(t)}, \tilde{U}^{(t)})$ by fixing $\tilde{U}^{(t)}$.

Since $\frac{\partial Z}{\partial U_{ij}^{(t)}} = -2((A^{(t)}V^{(t)})_{ij} + U_{ij}^*) + 2\frac{(\tilde{U}^{(t)}(V^{(t)\mathrm{T}}V^{(t)} + I))_{ij}U_{ij}^{(t)}}{\tilde{U}_{ij}^{(t)}}$, let $\frac{\partial Z}{\partial U_{ij}^{(t)}} = 0$, then we know that the value of $U_{ij}^{(t)}$ for minimizing

$Z(U^{(t)}, \tilde{U}^{(t)})$ is

$$U_{ij}^{(t)} = \tilde{U}_{ij}^{(t)} \frac{(A^{(t)}V^{(t)})_{ij} + U_{ij}^*}{(\tilde{U}^{(t)}V^{(t)\mathrm{T}}V_{ij}^{(t)}) + \tilde{U}_{ij}^{(t)}} \tag{24}$$

According to Eq. (20), we can obtain updating rule (7) from Eq. (24) after replacing $U^{(t)}$ in Eq. (24) by $U^{(t)}(\tau+1)$, and $\tilde{U}^{(t)}$ by $U^{(t)}(\tau)$. Therefore, by using Eq. (7) in each iteration of algorithm MF, the value of the objective function J is non-increasing.

Similarly, we can prove that by using rules (11), (15), and (18) in each iteration of algorithm MF, in order to update $V_{ij}^{(t)}$, U^*, and V^*, respectively, the value of the objective function J is also non-increasing. Q.E.D.

From Eq. (20), we know that the value of $U_{ij}^{(t)}$ by Eq. (24) is also non-increasing. Since $U_{ij}^{(t)} > 0$, it is bounded and converges. The correctness of the converged solution is ensured by the fact that at convergence from Eq. (24) the solution will satisfy:

$$2\left(-A^{(t)}V^{(t)} - U^* + U^{(t)}V^{(t)\mathrm{T}}V^{(t)} + U^{(t)}\right)_{ij} U_{ij}^{(t)} = 0 \tag{25}$$

This is the same as the fixed-point condition of Eq. (6). In a similar way, the convergence of the updating formula of $V^{(t)}$ in Eq. (11) can be proven.

To show the convergence of U^*, by using the updating formula for U^* in Eq. (15), we obtain

$$U^*(\tau+1) - U^*(\tau) = \frac{1}{\sum_{t=1}^{T} \lambda^{T-t}} \sum_{t=1}^{T} \lambda^{T-t} \left(U^{(t)}(\tau+1) - U^{(t)}(\tau)\right) \tag{26}$$

Since $U^{(t)}$ is a non-increasing and bounded function, it converges; namely, $U^{(t)}(\tau+1) - U^{(t)}(\tau) \leqslant 0$; hence, from Eq. (26), U^* is non-increasing, bounded, and converges. Since the updating formula of U^* in Eq. (15) satisfies the KKT condition, therefore, U^* converged to the optimal value that minimizes the objective function given in Eq. (12). This shows the correctness of updating U^*. Similarly, the convergence and correctness proves that V^* can be derived.

6 Experimental Results and Analysis

We conducted experiments on some real-world datasets in order to test our MF algorithm for link prediction. In the experiments, we tested the similarity-based predicting methods, CN, RA, PA, LP, and LHN2, to compare them with the proposed link prediction algorithm MF and the analytical experiment results. All the tests were conducted on an Intel Core i3 processor with 4 GB memory, under the Windows 7 operating system. The algorithms were coded in MATLAB.

6.1 Datasets

In our experiments, we tested five real-world network datasets, which are commonly used in link prediction system tests.

(1) *Irvine Message*[46]

The *Irvine Message* network consists of users representing students in the University of California, in Irvine. The network represents a web community, where the nodes denote the users and the links represent the communications between them.

(2) *Enron Email*[47]

In the *Enron* email network, the nodes represent the employees in Enron, and the links represent the email communications between the employees.

(3) *News Words*

In the *News Words* network, each node represents a word from a newspaper, and the edge between the two words indicates that they appear in the same sentence.

(4) *Nodobo*[48]

Nodobo is a network of cell phone calls between high-school students. In this network, the nodes represent the cell phone users, and the edges represent the cell phone communications between the users. In our experiments, we experimented with a sub-dataset over a two-month period, and predicted the phone calls based on daily phone calls.

(5) *Infectious SocioPatterns (Inf.Pattern)*[49]

Infectious SocioPatterns is a human maneuverability network. In this network, the nodes are the visitors in the Science Gallery, and the connections represent a face-to-face encounter between the visitors. In our experiment, we experimented on a sub-dataset consisting of 8-hour records in 96 time-frames. The length of each time-frame was 5 minutes.

Table 1 shows the statistical characters of five datasets, where #Nodes is the number of nodes, #Links is the number of edges, T_N is the length of the time series sequence dataset sparse, and P.length is the period length in hours.

Table 1 Statistical characters of datasets.

Dataset	#Nodes	#Links	P.Length (h)	T_N
Irvine Msg	896	201	55.556	25
Enron	704	925	240	33
News Words	503	15638	24	60
Nodobo	395	453	24	61
Inf.Pattern	200	714	0.0833	93

6.2 Experimental setup

On each dataset test, T_N snapshot graphs $G_1, G_2, ..., G_{T_N}$ were used. At each time step $t_0, t_0 = 1, 2, ..., T_N - T$, we predicted the links in G_{t_0+T} by using the previous T graphs, $G_{t_0}, G_{t_0+1}, ..., G_{t_0+T-1}$, and compared the performances of the link prediction algorithms. Since we already knew the topological structure of G_{t_0+T}, the performance of the algorithms could be estimated by comparing the set of predicted links to the actual ones in G_{t_0+T}.

In the experiments, we first tested algorithm MF, and compared its performance with the methods based on a compressed static network, which is often applied in methods for link prediction in dynamic networks. In these methods, a static network is constructed by compressing the dynamic networks at different times. Based on such a static network, the potential links are predicted; namely, dynamic network sequence $G_{t_0}, G_{t_0+1}, ..., G_{t_0+T-1}$ is compressed to one static network $G_{t_0,T}$. The adjacency matrix $G_{t_0,T}$ is denoted as $A_{t_0,T}$, with the entries in $A_{t_0,T}$ given by

$$A_{t_0,T}(i,j) = \begin{cases} 1, & \text{if } \exists k \in [t_0, t_0 + T - 1],\ A_k(i,j) = 1; \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

Then, a link prediction algorithm was performed on the compressed network, and the predicted results on $G_{t_0,T}$ were treated as the solution of the dynamic network.

To test the compressed static network method, we employed the similarity indexes of *Common Neighbor*, *Resource Allocation*, *Preferential Attachment*, *Local Path,* and *leicht-Holme-Newman,* denoted as CN, RA, PA, LP, and LHN2, respectively. We also compared the performance of our algorithm MF to the quality of these similarity indices in the compressed static network method.

In our tests, we use two models to represent the dynamic networks and the performances of similarity indices based on the two models were computed and compared. The first model was the binary static network described in Eq. (27). The second one used the NMF dynamic weighted network, which exploits latent features of non-negative matrix factorization. The approximated matrix $A^* = U^* V^{*\mathrm{T}}$ was used in order to provide weights for the second model, where $U^*$and $V^*$are the best fitting factors that minimize the objective distance function given in Eq. (2). For each of the similarity indices CN, RA, PA, LP, and LHN2, we compared the quality of the results obtained by using two different models. In all experiments, we referred to those methods respectively by CN, RA, PA, LP, and LHN2, under the binary model, and by MF-CN, MF-RA, MF-PA, MF-LP, and MF-LHN2, under the NMF dynamic weighted model.

In our experiments, the quality of prediction results by the algorithms was measured by Area Under Curve (AUC). AUC is a commonly used score for estimating the quality of the prediction results. To compute the AUC score, we ranked the similarity indexes of existing and non-existing edges. Then, we compared the similarity indexes of the n pairs of existing and non-existing edges. If among these n pairs of edges, the score of the existing edge was larger than the non-existing one in n' comparisons, and the score of existing edge was equal to the non-existing one in n'' comparisons, then, the AUC score was given by

$$\mathrm{AUC} = \frac{n' + 0.5n''}{n} \tag{28}$$

A higher AUC score commonly represents a better quality of predicted results. From Eq. (28) we can see that the highest AUC is 1, which indicates a correct result. The AUC score of a completely random prediction is 0.5.

6.3 Experimental results

In our tests, we set the widow size $T = 5$, and $\lambda = 0.8$, for all datasets. Since the dimension of the latent space k depends on the number of nodes in the dataset, we considered k as 5% of the number of nodes of each dataset.

In the first experiment of our tests, we tested and compared the qualities of the results predicted by our matrix factorization method MF to those of the CN, RA, PA, LP, and LHN2 methods. Figure 1 demonstrates the AUCs of the results on the six datasets, at each time, by MF and the compared methods. It can be clearly observed from Fig. 1 that our MF algorithm obtained the largest AUC value among all the tested methods, including LP and LHN2, on the Irvine Msg, Enron, and Nodobo datasets. It also demonstrated higher performance with the other two datasets. The reason for our MF obtaining the best quality prediction results was that the non-NMF technique used could efficiently amalgamate temporal and structural information with latent features.

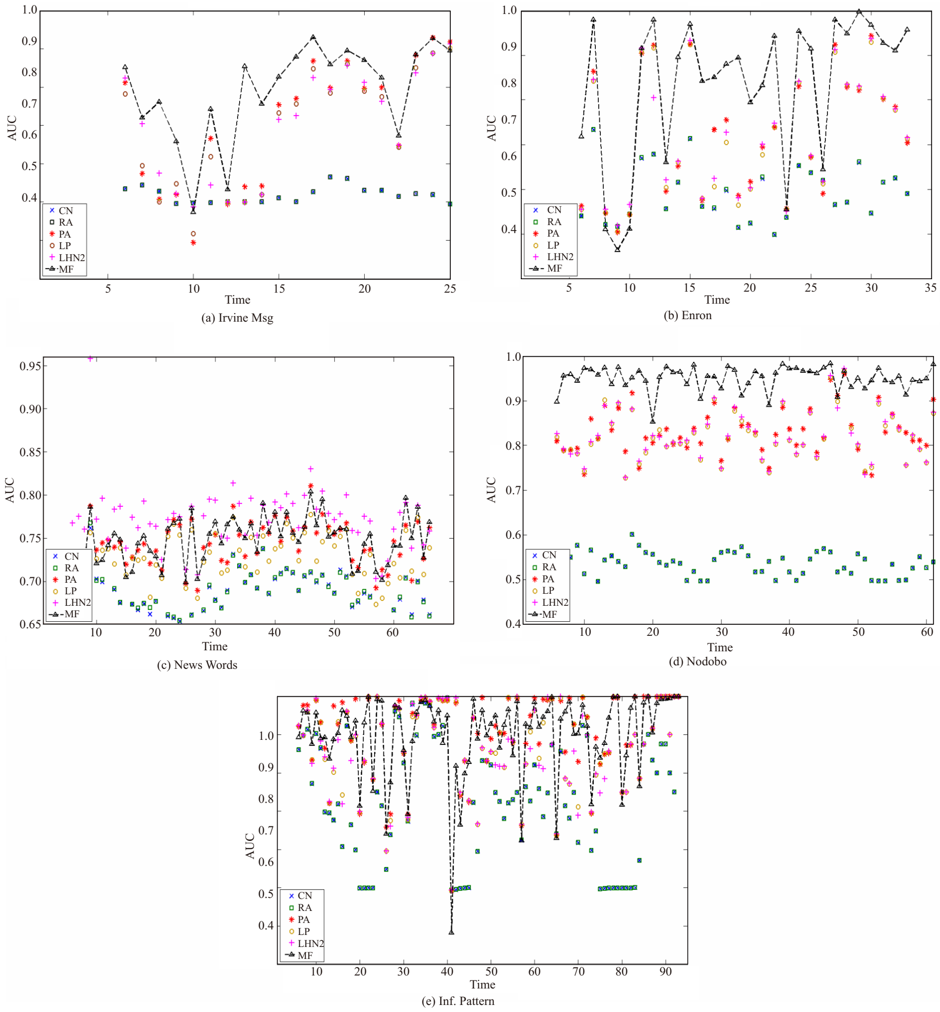

Fig. 1 Performance of MF and other methods.

Figure 2 shows the AUCs of the predicting results by algorithms MF and CN, RA, PA, LP, and LHN2 on the 5 datasets. From Fig. 2, we can see clearly that MF obtained the highest average performance on all 5 datasets.

In the second experiment, we compared the performances of algorithms CN, RA, PA, LP, and LHN2, based on two types of network models, namely, static binary network and NMF dynamic weighted network. Figures 3–7 show the AUC values of Irvine Msg, Enron, News Words, Nodobo, and Inf.Patterns datasets, respectively. From the figures, we can observe that among all methods, the NMF dynamic weighted network provides much better quality for prediction results, in comparison to the binary network for all datasets. The reason for our NMF dynamic weighted graph obtaining the highest quality results is that our model efficiently represents network dynamics and topology.

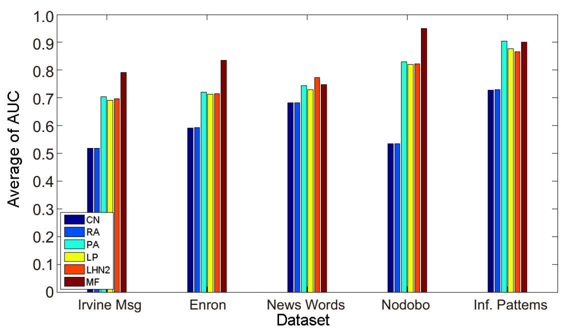

Fig. 2 AUC comparison for MF and other methods.

7 Conclusion

In this paper, we presented a new method for link prediction in dynamic networks. By this method, we obtained better link prediction results by proposing an algorithm using non-negative matrix factorization. The algorithm learned latent features from the dynamic and topological structure of a graph, and was shown

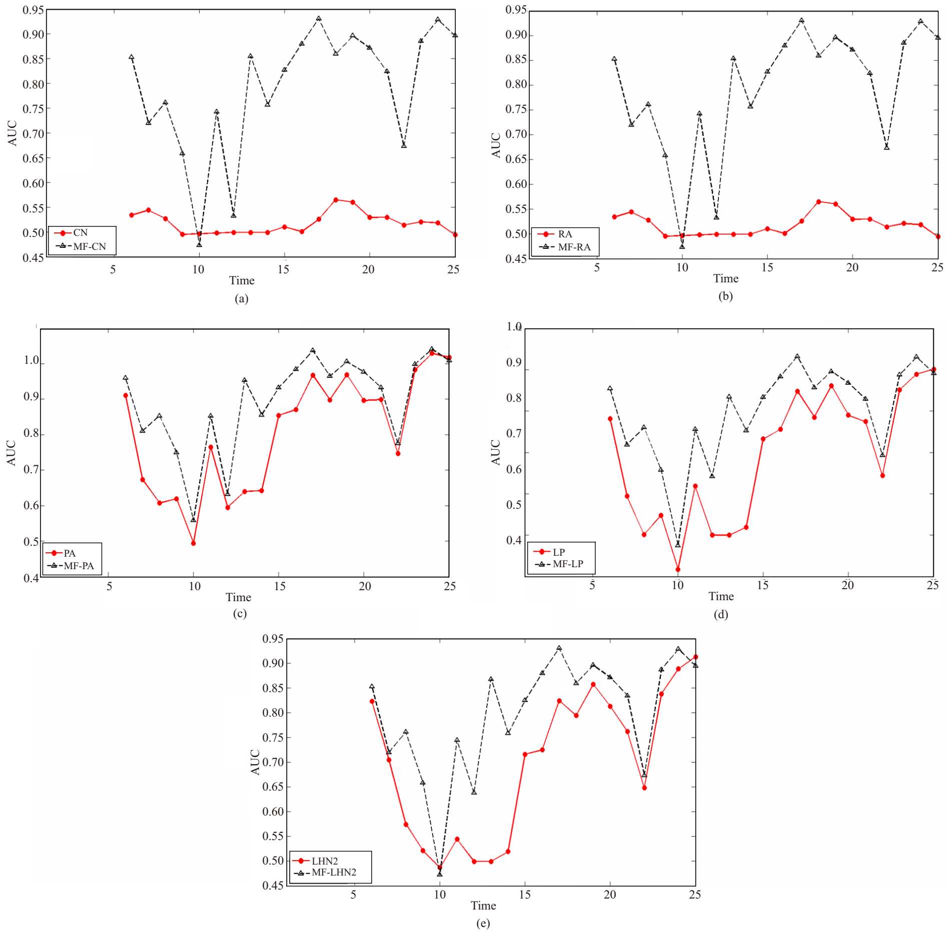

Fig. 3 Performance of different methods on Irvine Msg.

to make better predictions than other similarity-based methods. We provided new iterative rules in order to construct the NMF factors, and proved their convergence and correctness. We also demonstrated the better performance of similarity indices based on NMF weighted representation, rather than on static representation. Empirical results on real-world networks show that the proposed algorithm can achieve higher quality prediction results in dynamic networks, in comparison to other algorithms.

Acknowledgment

This research was supported in part by the National Natural Science Foundation of China (Nos. 61379066, 61702441, 61379064, 61472344, 61402395, 61702441, and 61602202); the Natural Science Foundation of Jiangsu Province (Nos. BK20130452, BK2012672, BK2012128, and BK20140492); the Natural Science Foundation of Education Department of Jiangsu Province (Nos. 12KJB520019, 13KJB520026, and 09KJB20013), and Six-talent-peak Project in Jiangsu Province (No. 2011-DZXX-032).

References

[1] N. M. Ahmed and L. Chen, An efficient algorithm for link prediction in temporal uncertain social networks, *Inf. Sci.*, vol. 331, pp. 120–136, 2016.

[2] F. Buccafurri, L. Fotia, G. Lax, and V. Saraswat, Analysis-preserving protection of user privacy against information leakage of social-network Likes, *Inf. Sci.*, vol. 328, pp. 340–358, 2016.

[3] M. Jankowski-Lorek, S. Jaroszewicz, Ł. Ostrowski, and A. Wierzbicki, Verifying social network models of Wikipedia knowledge community, *Inf. Sci.*, vol. 339, pp. 158–174, 2016.

[4] K. Saito, M. Kimura, K. Ohara, and H. Motoda, Super mediator—A new centrality measure of node importance for information diffusion over social network, *Inf. Sci.*, vol. 329, pp. 985–1000, 2016.

[5] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, A multi-step outlier-based anomaly detection approach to network-wide traffic, *Inf. Sci.*, vol. 348, pp. 243–271, 2016.

[6] H. Liao, A. Zeng, and Y. C. Zhang, Predicting missing links via correlation between nodes, *Physica A Stat. Mech. Appl.*, vol. 436, pp. 216–223, 2015.

[7] P. Wang, B. W. Xu, Y. R. Wu, and X. Y. Zhou, Link prediction in social networks: The state-of-the-art, *Sci. China Inf. Sci.*, vol. 58, no. 1, pp. 1–38, 2015.

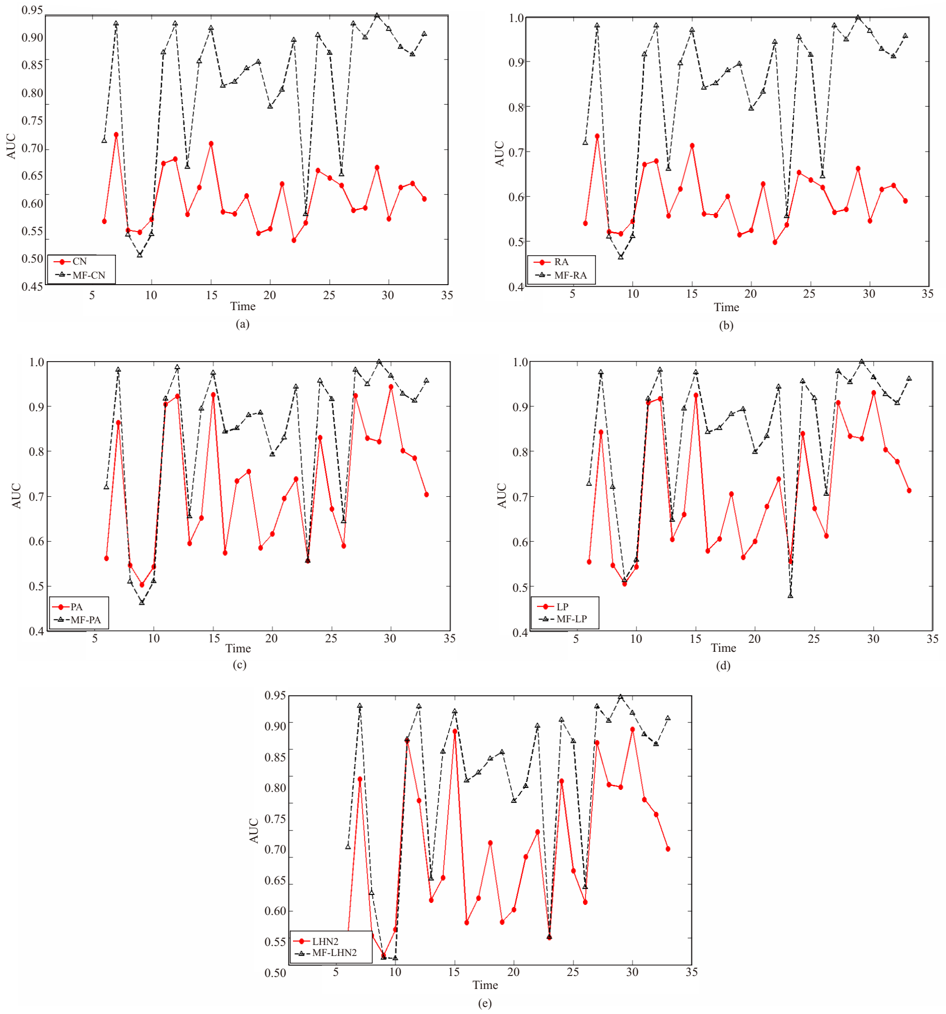

Fig. 4 Performance of different methods on Enron.

[8] X. M. Wang, Y. Liu, and F. Xiong, Improved personalized recommendation based on a similarity network, *Physica A Stat. Mech. Appl.*, vol. 456, pp. 271–280, 2016.

[9] B. Kaya and M. Poyraz, Age-series based link prediction in evolving disease networks, *Comput. Biol. Med.*, vol. 63, pp. 1–10, 2015.

[10] R. Guimerà, M. Sales-Pardo, and H. E. Stanley, Missing and spurious interactions and the reconstruction of complex networks, *Proc. Natl. Acad. Sci. USA*, vol. 106, no. 52, pp. 22073–22078, 2009.

[11] A. Nigam and N. V. Chawla, Link prediction in a semi-bipartite network for recommendation, in *Intelligent Information and Database Systems*, N. T. Nguyen, B. Trawi ki, H. Fujita, and T. P. Hong, eds. Springer, 2016, pp. 127–135.

[12] F. Xie, Z. Chen, J. X. Shang, X. P. Feng, and J. Li, A link prediction approach for item recommendation with complex number, *Knowl-Based Syst.*, vol. 81, pp. 148–158, 2015.

[13] Z. Sun, Q. K. Peng, J. Lv, and J. Zhang, A prediction model of post subjects based on information lifecycle in forum, *Inf. Sci.*, vols. 337&338, pp. 59–71, 2016.

[14] P. Klimek, A. S. Jovanovic, R. Egloff, and R. Schneider, Successful fish go with the flow: Citation impact prediction based on centrality measures for term–document networks, *Scientometrics*, vol. 107, no. 3, pp. 1265–1282, 2016.

[15] B. Kaya and M. Poyraz, Unsupervised link prediction in evolving abnormal medical parameter networks, *Int.J. Mach. Learn. Cybern.*, vol. 7, no. 1, pp. 145–155, 2016.

[16] M. Q. Ge, A. Li, and M. H. Wang, A bipartite network-based method for prediction of long non-coding RNA–protein interactions, *Genomics Proteomics Bioinformatics*, vol. 14, no. 1, pp. 62–71, 2016.

[17] F. Buccafurri, G. Lax, A. Nocera, and D. Ursino, Discovering missing me edges across social networks, *Inf. Sci.*, vol. 319, pp. 18–37, 2015.

[18] J. Fournet and A. Barrat, Contact patterns among high school students, *PLoS One*, vol. 9, no. 9, p. 107878, 2014.

[19] N. M. A. Ibrahim and L. Chen, Link prediction in dynamic social networks by integrating different types of information, *Appl. Intell.*, vol. 42, no. 4, pp. 738–750, 2015.

[20] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, Fast and accurate link prediction in social networking

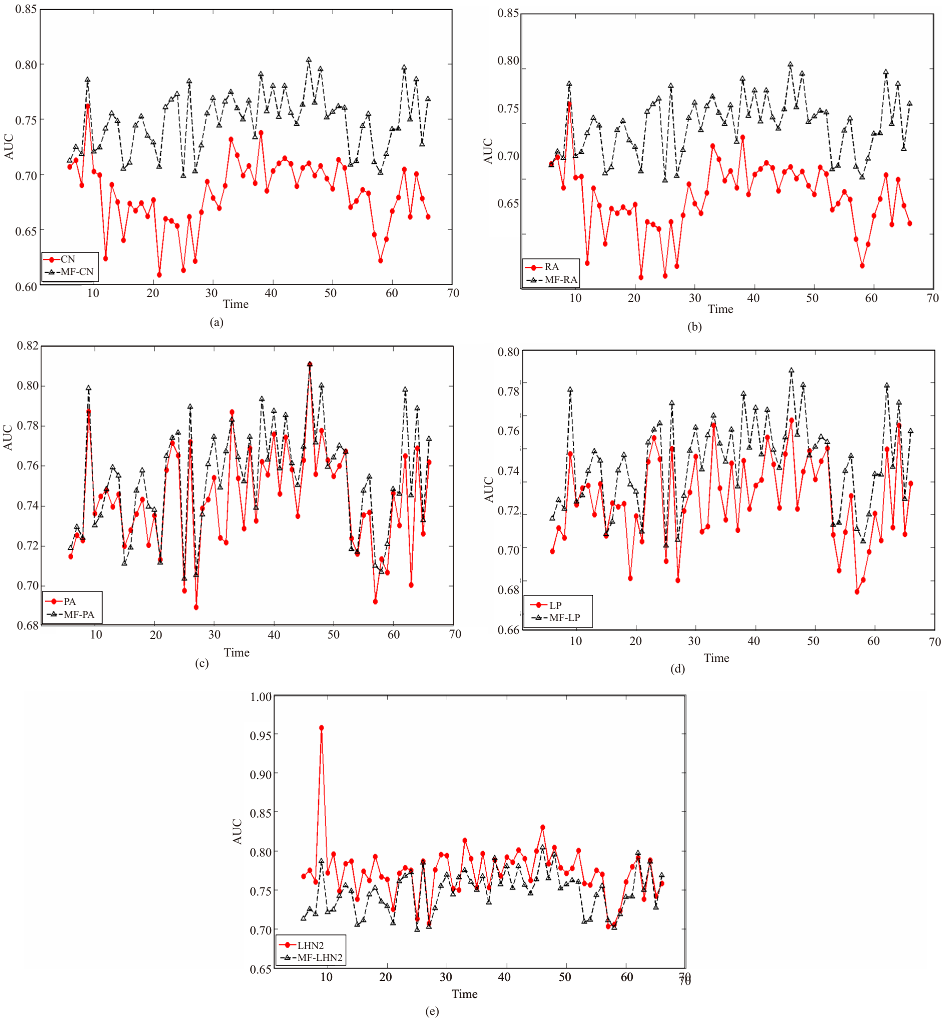

Fig. 5 Performance of different methods on News Words.

systems, *J. Syst. Softw.*, vol. 85, no. 9, pp. 2119–2132, 2012.

[21] Y. Z. Sun, R. Barber, M. Gupta, C. C. Aggarwal, and J. W. Han, Co-author relationship prediction in heterogeneous bibliographic networks, in *Proc. 2011 Int. Conf. Advances in Social Networks Analysis and Mining (ASONAM 2011)*, Kaohsiung, China, 2011, pp. 121–128.

[22] J. Li, L. L. Zhang, F. Meng, and F. H. Li, Recommendation algorithm based on link prediction and domain knowledge in retail transactions, *Procedia Comput. Sci.*, vol. 31, pp. 875–881, 2014.

[23] X. Li and H. Chen, Recommendation as link prediction in bipartite graphs: A graph kernel-based machine learning approach, *Decis. Supp. Syst.*, vol. 54, no. 2, pp. 880–890, 2013.

[24] A. Vidmer, A. Zeng, M. Medo, and Y. C. Zhang, Prediction in complex systems: The case of the international trade network, *Physica A Stat. Mech. Appl.*, vol. 436, pp. 188–199, 2015.

[25] B. Kaya and M. Poyraz, Supervised link prediction in symptom networks with evolving case, *Measurement*, vol. 56, pp. 231–238, 2014.

[26] X. Ma, J. L. Liao, S. M. Djouadi, and Q. Cao, LIPS: Link prediction as a service for data aggregation applications, *Ad Hoc Networks*, vol. 19, pp. 43–58, 2014.

[27] L. M. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, Friendship prediction and homophily in social media, *ACM Trans. Web (TWEB)*, vol. 6, no. 2, p. 9, 2012.

[28] M. W. Ahn and W. S. Jung, Accuracy test for link prediction in terms of similarity index: The case of WS and BA models, *Phys. A Stat. Mech. Appl.*, vol. 429, pp. 177–183, 2015.

[29] M. Hoffman, D. Steinley, and M. J. Brusco, A note on using the adjusted rand index for link prediction in networks, *Soc. Networks*, vol. 42, pp. 72–79, 2015.

[30] L. Y. Lü and T. Zhou, Link prediction in complex networks: A survey, *Phys. A Stat. Mech. Appl.*, vol. 390, no. 6, pp. 1150–1170, 2011.

[31] X. J. Wang, X. Zhang, C. L. Zhao, Z. Xie, S. J. Zhang, and D. Y. Yi, Predicting link directions using local directed path, *Phys. A Stat. Mech. Appl.*, vol. 419, pp. 260–267, 2015.

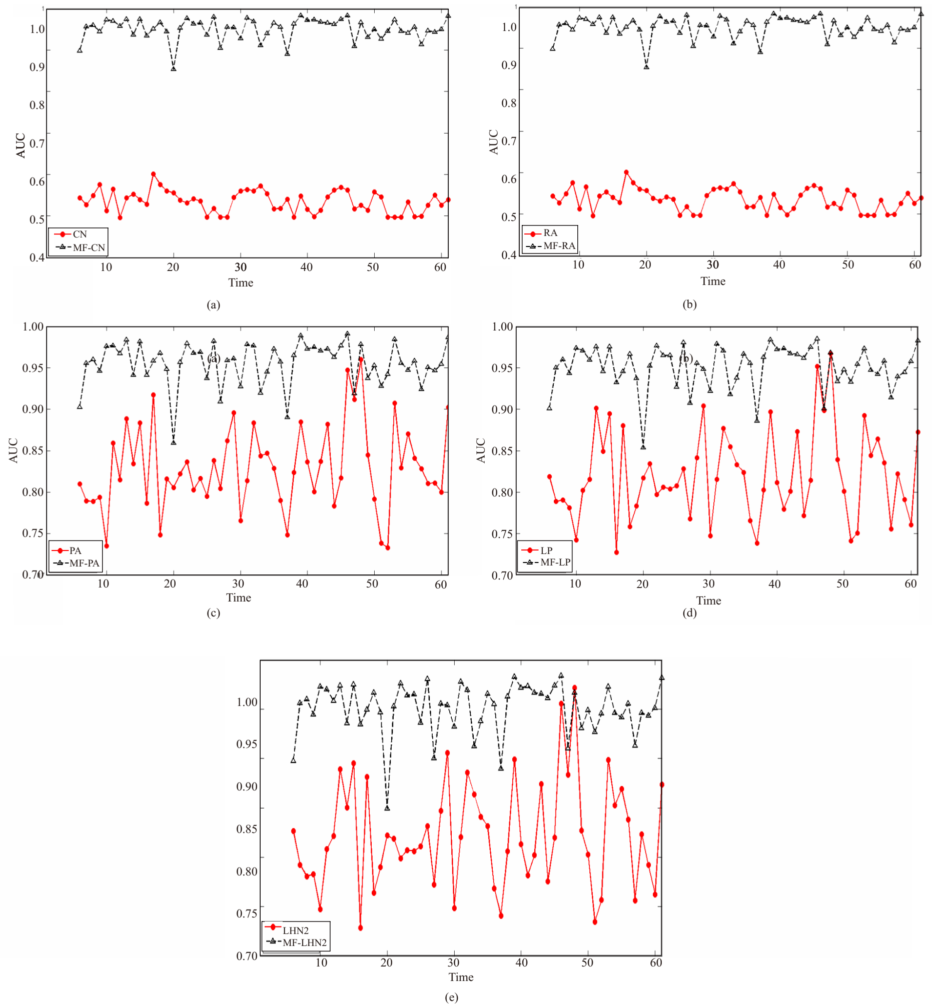

Fig. 6 Performance of different methods on Nodobo.

[32] A. Das Sarma, A. R. Molla, and G. Pandurangan, Distributed computation in dynamic networks via random walks, *Theor. Comput. Sci.*, vol. 581, pp. 45–66, 2015.

[33] W. P. Liu and L. Y. Lü, Link prediction based on local random walk, *Europhys Lett*, vol. 89, no. 5, p. 58007, 2010.

[34] F. Y. Hu and H. S. Wong, Labeling of human motion based on cbga and probabilistic model, *Int. J. Smart Sens. Intell. Syst.*, vol. 6, no. 2, pp. 583–609, 2013.

[35] N. Barbieri, F. Bonchi, and G. Manco, Who to follow and why: Link prediction with explanations, in *Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, New York, NY, USA, 2014, pp. 1266–1275.

[36] S. Gao, L. Denoyer, and P. Gallinari, Temporal link prediction by integrating content and structure information, in *Proc. 20th ACM Int. Conf. Information and Knowledge Management*, 2011, pp. 1169–1174.

[37] Z. Z. Zeng, K. J. Chen, S. B. Zhang, and H. J. Zhang, A link prediction approach using semi-supervised learning in dynamic networks, in *Proc. Sixth Int. Conf. Advanced Computational Intelligence (ICACI)*, Hangzhou, China, 2013, pp. 276–280.

[38] M. Pujari and R. Kanawati, Supervised rank aggregation approach for link prediction in complex networks, in *Proc. 21st Int. Conf. World Wide Web*, Lyon, France, 2012, pp. 1189–1196.

[39] Z. F. Bao, Y. Zeng, and Y. C. Tay, sonLP: Social network link prediction by principal component regression, in *Proc. 2013 IEEE/ACM Int. Conf. Advances in Social Networks Analysis and Mining*, Niagara Falls, Canada, 2013, pp. 364–371.

[40] Y. L. He, J. N. K. Liu, Y. X. Hu, and X. Z. Wang, OWA operator based link prediction ensemble for social network, *Expert Syst. Appl.*, vol. 42, no. 1, pp. 21–50, 2015.

[41] C. A. Bliss, M. R. Frank, C. M. Danforth, and P. S. Dodds, An evolutionary algorithm approach to link prediction in dynamic social networks, *J. Comput. Sci.*, vol. 5, no. 5, pp. 750–764, 2014.

[42] E. Sherkat, M. Rahgozar, and M. Asadpour, Structural link prediction based on ant colony approach in social networks, *Phys. A Stat. Mech. Appl.*, vol. 419, pp. 80–94, 2015.

[43] B. L. Chen, L. Chen, and B. Li, A fast algorithm for

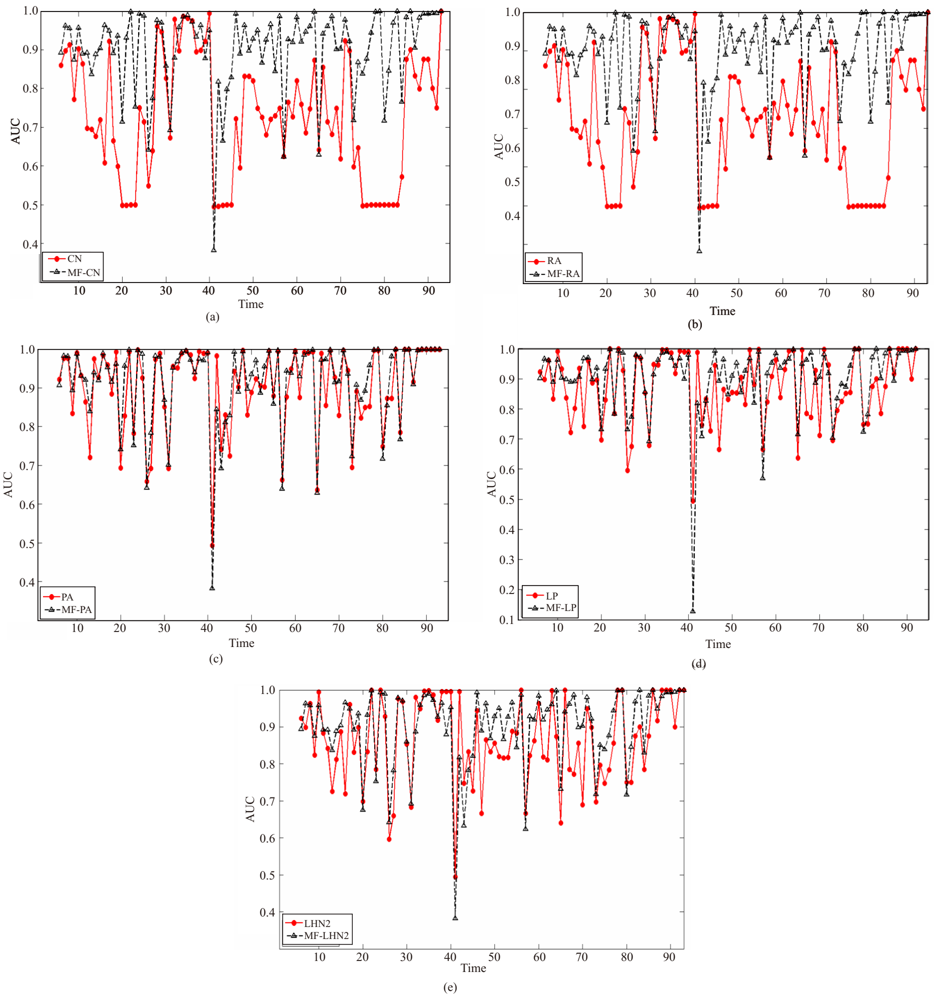

Fig. 7 Performance of different methods on Inf.Patterns.

predicting links to nodes of interest, *Inf. Sci.*, vol. 329, pp. 552–567, 2016.

[44] J. Y. Ding, L. C. Jiao, J. S. Wu, Y. T. Hou, and Y. T. Qi, Prediction of missing links based on multi-resolution community division, *Phys. A Stat. Mech. Appl.*, vol. 417, pp. 76–85, 2015.

[45] D. Q. Vu, A. U. Asuncion, D. R. Hunter, and P. Smyth, Continuous-time regression models for longitudinal networks, in *Advances in Neural Information Processing Systems 24, Proc. 25th Annual Conf. Neural Information Processing Systems*, 2011, pp. 1–9.

[46] http://www.cs.cmu.edu/∼enron/, 2017.

[47] http://konect.uni-koblenz.de/networks/enron, 2017.

[48] http://nodobo.com/release.html, 2017.

[49] SocioPatterns. DATASET: INFECTIOUS SocioPatterns dynamic contact networks, http://www.sociopatterns.org/datasets/infectious-sociopatterns-dynamic-contact-networks, 2011

Ling Chen got the BS degree from Yangzhou Teachers' College, China, in 1976. He is currently a professor in the Computer Science Department, Yangzhou University, China. His research interests include bioinformatics, data mining, and computational intelligence.

Yulong Wang got the PhD degree in agriculture from Ehime University, Japan, in 1996. He is currently a professor in the Agriculture Department, Yangzhou University, China. His research interests include bioinformatics and cultivation physiology.

Nahla Mohamed Ahmed got the master degree in mathematics from Institute of Mathematical Sciences, Cape town, South Africa in 2010, and the PhD degree in information and computing Science from Yangzhou University, China in 2016. She is currently a post-doc reseacher in Yangzhou University, China. She is an assistant professor in College of Mathematical Sciences, Khartoum University, Sudan. Her research interest is in complex network analysis.

Bin Li got the PhD degree in computer science from Nanjing University of Aeronautics and Astronautics, China, in 2001. He is currently a professor in the Computer Science Department, Yangzhou University, China. His research interests include computational intelligence and service computing.

Wei Liu got the PhD degree in computer science from Nanjing University of Aeronautics and Astronautics, China, in 2010. She is currently an associate professor in the Computer Science Department, Yangzhou University, China. Her research interests include bioinformatics and computational intelligence.

Yun Li got the PhD degree in computer science from Shanghai University, China in 2005. He is currently a professor in the Computer Science Department, Yangzhou University, China. His research interests include data mining and computational intelligence.