Send Orders for Reprints to reprints@benthamscience.ae

316 *The Open Cybernetics & Systemics Journal*, 2014, *8*, 316-321

Open Access

Link Prediction in Heterogeneous Networks Based on Tensor Factorization

Piao Yong[1,2*], Li Xiaodong[1] and Jiang He[1]

[1]*School of Software, Dalian University of Technology, Dalian, Liaoning, 116620, P.R. China;* [2]*School of Electronic and Information Engineering, Dalian University of Technology, Dalian, 116023, P.R. China*

Abstract: Link Prediction, that is, predicting the formation of links or interactions in a network in the future, is an important task in network analysis. Link prediction provides useful insights for other applications, such as recommendation system, disease-gene candidate detection and so on. Most link prediction methods assume that there is only one single type in the network. However, many real-world networks have heterogeneous interactions. Link prediction in such networks is challenging since (a) the network has a complicated dependency structure; and (b) the links of different types may carry different kinds of semantic meanings, which is important to distinguish the formation mechanisms of each link type. In this paper, we address these challenges by proposing a general method based on tensor factorization for link prediction in heterogeneous networks. Using a CANDECOMP/PARAFAC tensor factorization of the data, we illustrate the usefulness of exploring the natural three-dimensional structure of heterogeneous network. The experiment on real-world heterogeneous network demonstrates the effectiveness and efficiency of our methodology.

Keywords: Heterogeneous network, link prediction, tensor factorization.

1. INTRODUCTION

In recent years, complex network analysis has been receiving increasing attention by the scientific community due to the availability of massive network data from diverse domains. One fundamental task in network analysis is link prediction. Link prediction problem [1] can be broadly defined as: Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t'. It has been widely studied on homogeneous networks recently.

In reality, most networks are intrinsically multi-dimensional; there might be multiple connections between any pair of nodes. For example, the YouTube network contains user, video, tag as nodes, with links from types of follow-user, user-tag-video, and user-subscribe-video and so on. Those networks are broadly defined as heterogeneous information networks [2] (Fig. (**1**)). There are two typical ways of handling the link prediction problem in heterogeneous networks: treating all types of link equally; studying each type of link independently ignoring its correlation with other link types [3]. However, some information may be lost if different relations are not taken into account, which results in the problem: how to design an effective and general method for link prediction in heterogeneous network.

Heterogeneous networks can be organized as a third-order tensor ($Node \times Node \times Link\ type$) or multi-dimensional array. In this paper, we describe and evaluate one approach based on tensor factorization to the link prediction task in heterogeneous networks. To summarize, the following works have been done in this paper:

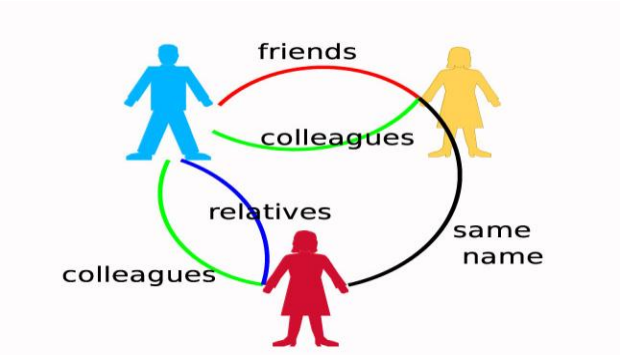

Fig. (1). Example of Heterogeneous Network.

1) We study the link prediction problem in networks involving heterogeneous interactions between actors.
2) We propose a method based on tensor factorization that can capture the correlation between different types of links for the link prediction problem without loss of information.
3) Experiments on real data sets have demonstrated the effectiveness of our approach compared with traditional link predictors.

The rest of the paper is organized as follows. In Section 2, we briefly review previous work related to our research. In Section 3, we propose a method based on tensor factorization for link prediction in heterogeneous networks. In Section 4,

*Address correspondence to this author at the School of Software. Dalian University of Technology, Dalian, Liaoning, 116620, P.R. China;
Tel: +86 0411 87571526; Fax: +86 0411 87571526;
E-mail: eric_piao@163.com

1874-110X/14

we report our experiments. We summarize this paper with our conclusion in Section 5.

Typically, we will use small letters, like *x*, *y z* to represent the node in a network and use the letter *e* to denote the edge. For a node x, $\Gamma(x)$ represents the set of neighbors of x. Tensors are denoted by calligraphic upper-case letters ($\mathcal{A}, \mathcal{B}$, …) matrices by uppercase letters (**A**, **B**, ...), vectors by bold lower case letters (**a**, **b**, ...), scalars by lower case letters (*a*, *b*, ...).

2. RELATED WORK

The literature on link prediction is vast; here we will only give a very brief interview. The seminal work of Liben-Norwell and Kleinberg [1] introduces numerous unsupervised methods for link prediction. That is by computing graph based similarity scores to predict the link between two vertices, the higher score, the more likely connected between two vertices. In addition to unsupervised approaches, recently many supervised methods have been raised [4-7] to extend the seminal work. Those methods treat link prediction as a binary classification problem by treating similarity metric as features. The challenge for those methods is feature engineer and it's difficult to extract good features about node and edge in network as privacy restriction. However most of the approaches introduced above are for homogeneous network, few works have been done for link prediction in heterogeneous network.

The link prediction problem has also been studied in the context of statistical relational learning [8, 9] and many proposed methods in those works can handle any relational data sets. The goal of these methods is to predict whether the type of links between a pair of objects exists. The advantages of those methods are that they are very general to data sets and can exploit the attributes of the objects. However those methods have a lot of parameters to tune and they are not very intuitive for us.

Perhaps the work by Evrim [10] is the closest to our work. However, they focus on temporal link prediction and model on homogeneous networks and have not considered how different relation types affect link formation.

Tensor factorizations have been widely applied to chemometrices and recently researches have applied tensor factorizations in web link analysis [11], network traffic analyzation [12] and also in social networks for analysis of chat room [13] and email communications [14] Moreover those works based on tensor factorization have achieved good results. More details of tensor factorizations and their applications can refer this excellent review paper [15].

3. LINK PREDICTION BASED ON TENSORFACTORIZATION

Since our method is based on tensor factorization, we first briefly review this technique. In this paper, we perform three-dimensional analysis on the data, attempting to discover the latent factors that govern the associations among these multi-type objects.

3.1. Preliminaries

Tensor. A tensor is a multidimensional array. More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system. A third tensor has three indices. A first order tensor is a vector, a second-order tensor is a matrix and tensors of order three or higher are called higher-order tensors.

Unfolding. Unfolding also known as matricization or flattening, is the process of reordering the elements of an *N*-way array into a matrix. The mode-*d* matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathcal{X}_n$. For a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, the mode-*n* unfolding are defined as follows [12]:

$$\mathcal{X}_{(1)}(i, p) = \mathcal{X}(i, j, k) \tag{1}$$

$$where\ p = j + (k-1)(J) \tag{2}$$

$$\mathcal{X}_{(2)}(j, p) = \mathcal{X}(i, j, k) \tag{3}$$

$$where\ p = i + (k-1)(I) \tag{4}$$

$$\mathcal{X}_{(3)}(k, p) = \mathcal{X}(i, j, k) \tag{5}$$

$$where\ p = i + (j-1)(I) \tag{6}$$

Kronecter Product. The symbol $\otimes$ denotes the Kronecter product of vectors, For example:

$$x = a \otimes b \Rightarrow x(k) = a(i)b(j) \tag{7}$$

$$where\ k = j + (i-1)J\ for\ all\ 1 \le i \le I, 1 \le j \le J \tag{8}$$

Khatri-Rao Product. The symbol $\odot$ denotes the Khatri-Rao product of two matrices. For example, let $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$:

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{A}(:,1) \otimes \mathbf{B}(:1) \quad \mathbf{A}(:,2) \otimes \mathbf{B}(:,2) \quad \cdots \quad \mathbf{A}(:,K) \otimes \mathbf{B}(:,K)] \tag{9}$$

Rank-One Tensors. An N-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is rank one if it can be written as the outer product of *N* vectors, *i.e.*

$$\mathcal{X} = a^{(1)} \circ a^{(2)} \circ \cdots \circ a^{(N)} \tag{10}$$

The symbol $\circ$ represents the vector outer product Tensor Rank. The rank of a tensor $\mathcal{X}$, denoted rank ($\mathcal{X}$), is defined as the smallest number of rank-one tensors that generate $\mathcal{X}$ as their sum

3.2. Data Representation

A tensor is a multidimensional array. If there are N nodes and K link types, then the data can be represented as a three-way tensor of size $N \times N \times K$ where the (*i*, *j*, *k*) entry is nonzero if node *i* is connected to node *j* by link type *k*. For the heterogeneous network example discussed at Sec.1, there are four links types exist: friendship connection, colleague connection, relative connection and the same name connection.

3.3. Tensor Decomposition based Link Prediction Model

We employ CANDECOMP/PARAFAC(CP) tensor decomposition [16] to capture the underlying patterns in the node-relationship-node tensor. The idea of the CP decomposition is factorizing a tensor into a sum of component rank one tensors, it is a higher-order analog of the matrix singular value decomposition (SVD). Given a third-order tensor $\mathcal{X} \in R^{I \times J \times K}$ we can write it as:

$$\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r \tag{11}$$

The $\circ$ denotes outer product of vectors, R is a positive integer and $a_r \in \mathbb{R}^I : a_r^T a_r = 1, b_r \in \mathbb{R}^J :$ $b_r^T b_r = 1$ and $c_r \in \mathbb{R}^K :$ $c_r^T c_r = 1$, $\lambda \in \mathbb{R}^+$ is used to normalize the columns of the matrices **A**, **B**, **C** to length one. In contrast to the solution matrix of SVD, the columns of factor the matrices **A**, **B**, **C** need not to be orthonormal. Fig. (**2**) Illustrates the decomposition.

\illustrates the decomposition.

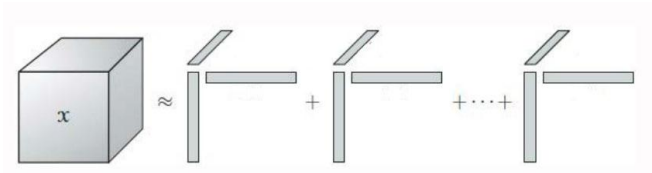

Fig. (2). CP Tensor Decomposition.

The CP decomposition can generate feature vectors for the nodes in the graph, which can be computed to get a similarity score that combines the multiple types of the graph. After CP decomposition, we get three factor matrices: node matrix *A*, relationship matrix *B*, and node matrix *C*. Link prediction can be computed according to the captured associations. We define a score matrix *S* as follows:

$$S = \sum_{r=1}^{R} \lambda_r a_r c_r^T \tag{12}$$

S Matrix measures the latent relations between different nodes in which S_{ij} represents the likelihood that node *i* will connect node *j*.

3.4. Computing the CP Decomposition

In this paper, we use alternating least-squares (ALS) with weighted-λ-regularization algorithm to fit the CP decomposition. Let $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ be a third-order tensor. The goal is to compute a CP decomposition with *R* components that best approximates $\mathcal{X}$, *i.e.*, to find.

$$\min_{\bar{\mathcal{X}}} \| \mathcal{X} - \bar{\mathcal{X}} \| \; with \quad \bar{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r a_r \circ b_r \circ c_r = [[\lambda; A, B, C]] \tag{13}$$

The problem of computing the CP decomposition that best approximates $\mathcal{X}$ can be formulated as a least squares optimization problem:

$$min\ f(A,B,C) = \frac{1}{2} \| \mathcal{X} - [[A,B,C]] \|_F^2 \Leftrightarrow \frac{1}{2} \| \mathbf{X}(1) - A(B \odot C)^T \|_F^2 \tag{14}$$

The ALS approach fixes **B** and **C** to solve **A** by minimizing the objective function (the sum of squared errors) the optimal **A** solution can be solved using the pseudo-inverse of a Khatri-Rao product:

$$A = X_{(1)}(C \odot B)((C^T C).*(B^T B))^{\dagger} \tag{15}$$

We can solve **B** and **C** using the same pattern. The ALS algorithm for CP decomposition is presented in Algorithm **1**:

4. EXPERIMENTS

In this section, we show that our proposed method can improve link prediction accuracy compared with the methods that only use homogeneous object and link information. First we will review some homogeneous methods as baseline.

4.1. Baseline Methods

All the methods can be viewed as computing a measure of proximity of similarity between nodes *x* and *y*. They assign a weight $score(x, y)$ to pairs of nodes $< x, y >$, based on the network topology, and then produce a ranked list in decreasing order of $score(x, y)$.

Node Neighborhood Methods. For a node *x*, let $\Gamma(x)$ denotes the set of neighbors of *x*. If two nodes neighbors have large overlap, they are more likely to form a link in the future.

Algorithm 1: ALS method for CP decomposition

Input: tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, $N > 0$ is the maximum number of iterations, $\sigma > 0$ is the stopping critical point.
Output: matrix $\mathbf{A} \in \mathbb{R}^{I \times R}; \mathbf{B} \in \mathbb{R}^{J \times R}; \mathbf{C} \in \mathbb{R}^{K \times R}, \lambda \in \mathbb{R}^R$

Compute CP decomposition: $cp(\mathcal{X}, N, \sigma)$;
$n = 0$;
while $n \leq N$ *and* $||\mathcal{X} - [[\lambda; A, B, C]]|| > \sigma$ **do**
    n = n + 1 ;
    $\mathbf{A} = X_{(1)}(\mathbf{C} \odot \mathbf{B})((\mathbf{C}^T\mathbf{C}). * (\mathbf{B}^T\mathbf{B}))^{\dagger}$;
    Normalize the columns of **A** to length one ;
    $\mathbf{B} = X_{(1)}(\mathbf{C} \odot \mathbf{A})((\mathbf{C}^T\mathbf{C}). * (\mathbf{A}^T\mathbf{A}))^{\dagger}$;
    Normalize the columns of **B** to length one ;
    $\mathbf{C} = X_{(1)}(\mathbf{B} \odot \mathbf{A})((\mathbf{B}^T\mathbf{B}). * (\mathbf{A}^T\mathbf{A}))^{\dagger}$;
    Normalize the columns of **C** to length one ;
end

Common neighbors. The more common neighbors between x and y, the chance that x and y have a link between them increases. The score defined as below:

$$score(x,y) := |\Gamma(x) \cap \Gamma(y)| \tag{16}$$

Newman [17] has computed this quantity in the context of collaboration networks to show that a correlation exists between the numbers of common neighbors of x and y at time t, and the probability that they will collaborate in the future.

Jaccard's coefficient:

$$score(x,y) := \frac{\Gamma(x) \cap \Gamma(y)}{\Gamma(x) \cup \Gamma(y)} \tag{17}$$

Adamic/Adar measure. Adamic and Adar [18] proposed this score as a metric of similarity between two web pages.

$$\sum_{z:\, feature shared by\, x,y} \frac{1}{log(frequency(z))} \tag{18}$$

For link prediction, Liben-Nowell and Kleinberg [1] refined this metric and assigned large weight to common neighbors z of x and y which themselves have few neighbors $|\Gamma(z)|$ The score defined as below:

$$score(x,y) := \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{log|\Gamma(z)|} \tag{19}$$

Preferential attachment: The premise of preferential attachment is that nodes like to form ties with popular nodes. A new edge has node x as its endpoint is proportional to $|\Gamma(x)|$. The score defined as below:

$$score(x,y) := |\Gamma(x)||\Gamma(y)| \tag{20}$$

Path Methods:

katz measure Sums over all possible paths between x and y, giving higher weights to shorter paths. The score defined as below:

$$score(x,y) = \sum_{l=1}^{\infty} \beta^l |paths_{x,y}^{(l)}| \tag{21}$$

where $\beta > 0$ and $paths_{x,y}^{(l)}$ is the set of all length-l paths from x to y.

SimRank is defined by the following recursive procedure: two nodes are similar to the extent that their neighbors.

Specially, $similarity(x,x) = 1$

$$similarity(x,y) = \gamma \frac{\sum_{a \in \Gamma(x)} \sum_{b \in \Gamma(y)} similarity(a,b)}{|\Gamma(x)||\Gamma(y)|} \tag{22}$$

where $\gamma \in [0,1]$, for link prediction *score*(*x, y*) = *similarity* (*x, y*).

4.2. Data Sets

UMLS. This data set contains data from the Unified Medical Language System semantic work gathered by [19]. This consists of 135 entities and 54 relationships. The entities are high-level concepts like 'Disease or Syndrome', 'Diagnostic Procedure', or 'Mammal'. The relation verbs illustrate the relationship between those concepts.

4.3. Parameter Settings

We use the top-k recommendations metric [20] to evaluate link prediction results. That is, we will predict top k links for each user. The procedure is as below: for each user, we randomly select one linked user and k-1 unlinked to form the test set and the remaining linked user form the training set. We repeat this procedure 60 times for different samplings. The goal is to find the position of the linked user in the recommendation list. There are k possible ranks for the linked user and the best result is that no unlinked users appear before it.

4.4. Dealing with High Computing Cost

It's cost intensively to compute tensor factorization. So in this paper, we conducted our experiments on a Linux Ubuntu server with four Intel Xeon CPUs, each CPU has two cores, and each core has 13.5 MB cache. We parallel [21] our Matlab code, and this significantly improve the speed of the algorithm. We found that the ALS based tensor factorization performs better.

4.5. Results and Discussions

To demonstrate the usefulness and effectiveness of our model, we offer comparison experiments on unsupervised link prediction methods. According to [1, 22]. Adamic/Adar measure and Katz measure performs well both in theoretical and practical experiments. So here, we just compare these two measures with our methods. Figs. (**3** and **4**) show the cumulative distributions of ranks for the connected users in the test set (k=81). 0% means that the connected user is at the first place in the ordered list, while 100% means that it is in the last position.

Comparing the results, in Fig. (**3**) our method provides better precision than unsupervised ones on the data sets. And in Fig. (**4**) our method provides competitive effect to Adamic/Adar measure and both those two methods beat Katz measure. The reason for that can be explained that our model can capture the latent multi-relationships of data.

5. CONCLUSION

In this paper, we study the problem of link prediction in heterogeneous networks. Heterogeneous networks contain multi-relational links between entities and traditional matrix based methods are not expressive enough to model heterogeneous networks without information loss. We propose a new general and effective method based on tensor factorization to address this problem. Experiments on the real world network show that by considering multiple relations, the link prediction accuracy can be significantly improved. In the future, we will try to scale our model for very large scale heteroge-

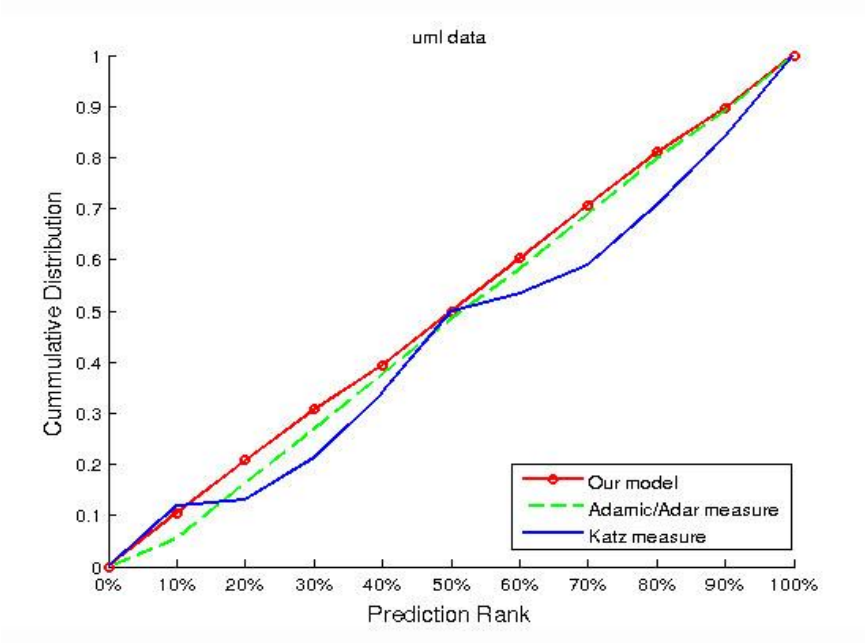

Fig. (3). Top-81 Link Prediction Performance for UML Data (Result 1).

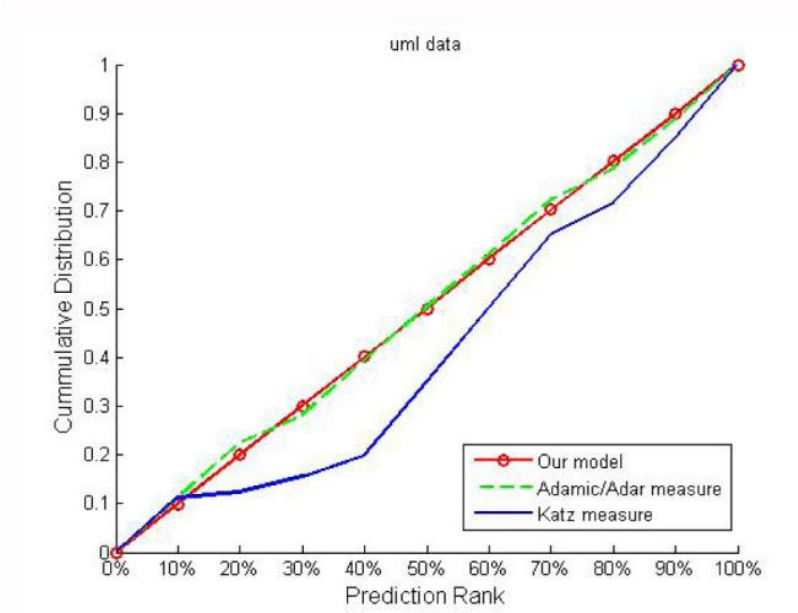

Fig. (4). Top-81 Link Prediction Performance or UML Data (Result 2).

neous networks, such as utilizing the Hadoop platform to speed up tensor factorization.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the National Science Foundation of software bug warehouse priority mining project (No.61370144).

ABOUT THE AUTHORS

First Author Zhao Jing, University lecturer of Shenyang University of Technology, Ph.D. The author's major is Fluid Machinery and Engineering. 16 papers received by EI or SCI and 2 patents had been published.

Second Author Wang Jing, master degree in engineering, studying for PhDs in Beijing University of Chemical Technology. The author's major is Mechatronic Engineering.

Third Author Wang Shijie is a member of the IEEE and the IEEE Computer Society.

REFERENCES

[1] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. American Society for Information Science, vol.58 no. 7, pp. 1019-1031, 2007.

[2] J. Han. Mining heterogeneous information networks by exploring the power of links. In Discovery Science, vol. 12, pp. 13-30, 2009.

[3] D. Davis, R. Lichtenwalter, and N. Chawla. Multi-relational link prediction in heterogeneous information networks. In Advances in Social Networks Analysis and Mining, vol. 18, pp. 281-288, 2011.

[4] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11, New York, 2011, pp. 635-644.

[5] A. Popescul and L. Ungar. Statistical relational learning for link prediction. In IJCAI workshop on learning statistical models from relational data, vol. 2003, pp. 348-355, 2003.

[6] M. Pujari and R. Kanawati. Supervised rank aggregation approach for link prediction in complex networks. In Proceedings of the 21st international conference companion on World Wide Web, Hong Kong, 2012, pp. 1189-1196.

[7] N. Benchettara, R. Kanawati, and C. Rouveirol. Supervised machine learning applied to link prediction in bipartite social networks. In Advances in Social Networks Analysis and Mining, Chicago, 2010, pp. 326-330.

[8] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In Proc. of SDM 06 workshop on Link Analysis, Counterterrorism and Security, New York, 2006, pp. 120-128.

[9] Z. Xu, V. Tresp, S. Yu, and K. Yu. Nonparametric relational learning for social network analysis. In KDD 2008 Workshop on Social Network Mining and Analysis, Washington, 2008, pp. 60-66.

[10] E. Acar, D. M. Dunlavy, and T. G. Kolda. Link prediction on evolving data using matrix and tensor factorizations. In ICDMW'09: Proceedings of the 2009 IEEE International Conference on Data Mining Workshops, New York, 2009, pp. 262-269.

[11] T. G. Kolda, B. W. Bader, and J. P. Kenny. Higher-order web link analysis using multilinear algebra. In IEEE International Conference On Data Mining, 2005, Chicago, pp. 242-249.

[12] J. Sun, D. Tao, and C. Faloutsos. Beyond streams and graphs: dynamic tensor analysis. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, Beijing, 2006, pp. 374-383.

[13] E. Acar, S. A. ampere. Collective sampling and analysis of high order tensors for chartroom communications. In ISI 2006: IEEE International Conference on Intelligence and Security Informatics, Hong Kong, 2006, pp. 213-224.

[14] B. Bader, M. Berry, and M. Browne. Discussion tracking in enron email using parafac. Survey of Text Mining, vol. 2, pp. 147-163, 2008.

[15] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. SIAM Review, vol.51, no. 3, pp. 455-500, Sep 2009.

[16] J. Carroll and J. Chang. Analysis of individual divergences in multidimensional scaling *via* an n-way generalization of "eckart-young" decomposition. Psychometrika, vol. 35, no. 3, pp. 283-319, 1970.

[17] M. Newman. Clustering and preferential attachment in growing networks. Physical Review E, vol.64, no. 2, pp. 25-102, 2001.

[18] L. Adamic and E. Adar. Friends and neighbors on the web. Social networks, vol. 25, no. 3, pp. 211-230, 2003.

[19] A. T. McCray. An upper-level ontology for the biomedical domain. Comparative and Functional Genomics, vol. 4, no. 1, pp. 80-84, 2003.

[20] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative ltering model. In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, Hong Kong, 2008, pp. 426-434.

[21] J. Kepner. Parallel MATLAB for multicore and multinode computers, Society for Industrial Mathematics, vol.21, pp. 150-165, 2009.

[22] P. Sarkar, D. Chakrabarti, and A. Moore. Theoretical justification of popular link prediction heuristics. In Proceedings of the Twenty-Second international joint conference on Artificial Intelligence, New York, 2011, pp. 2722-2727.

Received: September 22, 2014 Revised: November 30, 2014 Accepted: December 02, 2014