LINK PREDICTION ON EVOLVING NETWORK USING TENSOR-BASED NODE SIMILARITY

Xiao Yang, Zhen Tian,Zhaoxin Zhang

School of Computer Science and Technology,
Harbin Institute of Technology, Harbin 150001, China
shawcsn@gmail.com, zhentian@hit.edu.cn, heart@hit.edu.cn

Abstract: Recently there has been increasing interest in researching links between objects in complex networks, which can be helpful in many data mining tasks. One of the fundamental researches of links between objects is link prediction. Many link prediction algorithms have been proposed and perform quite well, however, most of those algorithms only concerns network structure in terms of traditional graph theory, which lack information about evolving network. In this paper we proposed a novel tensor-based prediction method, which is designed through two steps: First, tracking time-dependent network snapshots in adjacency matrices which form a multi-way tensor by using exponential smoothing method. Second, apply Common Neighbor algorithm to compute the degree of similarity for each nodes. This algorithm is quite different from other tensor-based algorithms, which also mentioned in this paper. In order to estimate the accuracy of our link prediction algorithm, we employ various popular datasets of social networks and information platforms, such as Facebook and Wikipedia networks. The results show that our link prediction algorithm performances better than another tensor-based algorithms mentioned in this paper.

Keywords: Link prediction; Tensor; Node similarity; Temporal network analysis

1 Introduction

As the increase in online social network and information platform, people show more interest in complicated network. Despite that there are already numerous studies on snapshots of single networks, but these studies lack dynamic information of evolving network [1, 2].Extracting dynamic information which changes over time by analyzing the increase or decrease in amount of nodes in network is necessary to solve the problem of link prediction.

The link prediction problem we are solving is: when given link data of frontal T time periods, can we predict the link relationships at time $T+1$. Similar problems have been discussed under different backgrounds [3, 4].Time-based link prediction differs from missing link prediction problem[5] (times series problem doesn't exist in missing link prediction, and the aim of missing link prediction is to predict the missing links at a given time, thus get a comprehensive description about whole structure of links in the network).

Here we adopt tensor-based method to solve time series link prediction problem. First of all, experimental data comes from online network, which can be expressed as $i*j*t$ like high-dimensional array namely three-dimensional tensor, *i* and j respectively stand for users and items, t means the links between object *i* and object *j* at time *j*. In simplest condition, we define three-dimensional tensor *Z* as:

$$Z(i,j,t)=\begin{cases}1 & \textit{if object i links to object j at time t}\\ 0 & \textit{otherwise}\end{cases}$$

In order to make it possible to observe trends within time, by analyzing three-dimensional tensor link structure，we define time as an individual dimension. And dynamic information is used to predict the link structure at time $T+1$.

In this paper, we make use of link prediction algorithm based on three-dimensional tensor. The method consists of two independent parts: First, we process network data that expressed as a three-dimensional tensor *Z* through Holt-Winters model [6] and thus obtain time series information; which can be used to predict link structure in future. Second we adopt Common Neighbors (CN) algorithm to process data we get from the first step, and thus obtain the prediction result.

We further discuss how we design tensor-based link prediction method and explain its two parts in detail. On one hand, we describe how to obtain tensor data and network topology information over time; on the other hand, we explain how CN model works. We then make use of AUC [7] index to evaluate the effectiveness of the link prediction algorithm.

This paper is structured as follows: in Section 2 we talk about relevant researches on tensor-based link prediction; and Section 3, we introduce our link prediction method in detail; in Section 4, we explain our experimental result; Section 5 is summary of whole paper.

1.1 Notation

Vectors are expressed as lowercase letters, for example, *a*; matrices are expressed as bold capital letters, for example, *A*; the rth row of matrix is expressed as

kth a_r ; high-dimensional matrix is expressed as uppercase letter, for example, *Z*; the tth slice of tensor *Z* as z_t . The ith component of vector a is express as $a(i)$ and element (i, j) of matrix *A* as $A(i, j)$, element (i, j, k) of three-dimensional tensor *Z* as $z(i, j, k)$.

2 Related Work

Recently, a numerous methods based on tensor factorization have been proposed [8, 9], which nearly all base on CP tensor model [10, 11, 12].

2.1 CP Tensor Model

CP tensor model is one of the most commonly used and most effective tensor models, to a three-dimensional tensor *Z* defined as *M*N*T*, its K-component CP factorization is defined as:

$$z \approx \sum_{k=1}^{K} \lambda_k a_k \circ b_k \circ c_k \qquad (1)$$

symbol ◦ stands for outer product, $k = 1,...,K$. $\lambda_k \in \mathbb{R}_+$, $a_k \in \mathbb{R}^M$, $b_k \in \mathbb{R}^N$,and $c_k \in \mathbb{R}^T$, where $k = 1,.....,K.$ each summand $(\lambda_k a_k \circ b_k \circ c_k)$ is called a component, each vector called a factor. We assume $\|a_k\| = \|b_k\| = \|c_k\| = 1$ and therefore λ_k contains the scalar weight of the kth component. A three way outer product is defined as follows:

$$\chi = a \circ b \circ c \Leftrightarrow \chi(i, j, k) = a(i)b(j)c(k)$$

2.2 CP Scoring Using Temporal Forecasting

We assign scores to each pair (i, j) according to the probability of linking at time $T+1$ by using components extracted by CP model. The outer product c_k captures the temporal profiles. a_k , b_k quantify the relationship between objects in component K. In order to obtain the similarity score between object pairs, we use prediction method given by Holt-Winters [13], the model is appropriate for dealing with time series data, in which the input is simply the data to process, and the effectiveness of the model have been validated over and over, thus, it can be used as a prediction tool.

Through the model, we can evaluate the $(K+1)th$ frontal slice of tensor *Z*, the slice contains the future correlation of objects pairs (such as users or items). Via Holt-Winter model, we can figure out vector

$$x_{i,j,k+1} \approx \sum_{r=1}^{R} \lambda_r (a_{i,r} \cdot b_{j,r} \cdot c_{K+1,r}) \qquad (2)$$

and not only can we evaluate the k+1th frontal slice (see equation 2),but also we can get the similarity score between object i and j in the future:

$$x_{i,j,k+1} \approx \sum_{r=1}^{R} \lambda_r (a_{i,r} \cdot b_{j,r} \cdot c_{K+1,r}) \qquad (3)$$

3 Description of Tensor Based Prediction models

In this paper, we proposed anther tensor-based link prediction algorithm. Different from the link prediction algorithm based on tensor factorization mentioned in Section 2, the algorithm doesn't include tensor factorization and three-dimensional tensor calculation. And different from traditional link prediction techniques: the method includes two independent steps, while at the same time, it can efficiently solve the problem of link prediction in evolving network, while past techniques usually finish the work in one step, and only aim at topology of single network snapshot.

We use a tensor *Z*, which is a set of slices which represents the relational data of frontal T time periods. Two-dimensional tensor is adjacency matrix of relation between the network nodes at a given time period. Hence we map three-dimensional tensor *Z* to a two-dimensional matrix which contains trends within times of each edge using Holt-Winters model, the higher the weight in the matrix, the closer the relationship between two objects. We then make use of Common Neighbor method to quantize the similarity score between two objects.

We take two-dimensional matrix as adjacency matrix in Common Neighbor model, thus, figure out another two-dimensional matrix S. The values in the matrix show the intimacy between nodes at time $T+1$, the greater the value, the higher the similarity. In other words, the higher the similarity between two nodes, the greater possibility that they will have relation at time $T+1$.

3.1 Turn three-dimensional tensor into two-dimensional matrix

First we store the data into tensor *Z*. Every slice $z_t(i, j)$ in *Z* represents the state of object *i* and object *j* at time period *t*. If $z_t(i, j)$ equals 1, there is an edge that links object *i* and object *j* at *t*, otherwise there is no. Tensor is a set made up of slices $(\{z_1, z_2, ..., z_k\})$, in which z_k represents the adjacency matrix of the network at time k, we adopt improved exponential smoothing model to turn three-dimensional tensor *Z* into two-dimensional matrix.

Exponential smoothing is a method which can be applied to process time series data while at the same time produce smoothed data for prediction. Time series data is a sequence of observed data, which is usually

expressed as x_t , and the result of Exponential smoothing algorithm is usually expressed as s_t .

When time series data starts from 0, the simplest exponential smoothing algorithm can be displayed as following equations:

$$s_1 = x_0 \tag{4}$$

$$s_{t+1} = \alpha x_t + (1-\alpha)s_t \tag{5}$$

α is called smooth factor whose value range is $0<\alpha<1$. In our link prediction algorithm, we make use of exponential smoothing method to turn three-dimensional tensor into two-dimensional matrix. The value in the matrix can be applied to predict the future relation of variables. Formula used to produce the two-dimensional matrix is described as below:

$$\begin{aligned} z_{t+1} &= \alpha * z_t + \alpha * (1-\alpha) * z_{t-1} \\ &+ \alpha * (1-\alpha)^2 * z_{t-2} + ... + (1-\alpha)^t * z_1 \end{aligned} \tag{6}$$

z_{t+1} stands for two-dimensional matrix with weight that transformed by exponential smoothing method, and the value of $z_{t+1}(i, j)$ is intimacy of object *i* and object *j* at time $T+1$.

3.2 Common Neighbor

As a general rule, the greater the number of common neighbors that object x and object y have, the greater possibility that there will be a link between them in the future.

To an object x, $\Gamma(x)$ represents a set of neighbors of object x. The simplest measure of this neighborhood overlap is directed count, namely

$$s_{xy}^{CN} = \left|\Gamma(x) \cap \Gamma(y)\right| \tag{7}$$

In formula (7), we obviously have $s_{xy} = (A^2)_{xy}$, A represents adjacency matrix, in the method, A is equivalent to z_{t+1} we obtain in 3.1, the value of $A(i, j)$ means the intimacy between i and j at time $t+1$. We can interpret adjacency matrix A as matrix A^1 with weight: $A^1_{xy} = 1$ indicates that there is a direct link between x and y, otherwise $A^1_{xy} = 0$. The correctness of the method has been repeatedly validated under different backgrounds. [13,14].

s_{ij} represents the similarity between object *i* and object *j*, the greater the value is, the greater possibility that there is a link between them.

4 Evaluation

In order to evaluate the accuracy of our Tensor-based Node Similarity (TBNS) link prediction algorithm, we have to divide the dataset into training set and test set. *U* stands for dataset, and E^{tr} stands for training set, and $U - E^{tr}$ is the test set. Training set is used as the input data of three-dimensional tensor analysis, and test set is used to compare with result of prediction to evaluate the accuracy of the prediction.

Dataset can be divided by percentage settled beforehand or by time. In order to better analyze and process time series data, we divided the dataset by time. After dividing the dataset, E^{tr} represents the link relation between object at frontal T time periods. $U - E^{tr}$ means the link relations between object we want to predict at $T+1$.

We express the network data from the Internet as three-dimensional tensor Z in form of $X * Y * T$, because what we need is the time series data trend information, $Z(i, j, t) = 1$ means there is a link between i and j at time t, otherwise there is no link. We divide dataset by time in table 1 into training set and test set.

Table 1 Properties of Examined Datasets

Dataset	Structure	X	Y	Spasity
Autonomous system	*router*router*time*	65535	65535	0.023
Wiki-edit-Esperanto	*user*page*time*	7211	296542	0.13
Facebook-wall posts	*user*user*time*	63890	63891	0.021

These datasets we select to examine are Autonomous system, Wiki-edit-Esperanto and Facebook-wall posts. Autonomous system is sub-graphs of the graph of routers comprising the internet. Each node exchanges traffic flows with some neighbors. Wiki-edit-Esperanto is a User-article edit network, in which user can edit article and become one of the editors of this articles. Facebook-wall posts dataset is abstracted from Facebook New Orleans networks. Each line contains two anonymous user identifiers, meaning user appeared in the first user's friend list.

The accuracy of link prediction is relevant to the evaluation method, therefore we adopt regularly used area under the receiver operating characteristic curve (AUC) to evaluate the accuracy of our link predictions and meanwhile compare with the tensor-based algorithm mentioned in Section 2. In Figure 1, we show the ROC(receiver operating characteristic)curves. We estimate the AS dataset and Figure 1(a) shows that all methods performing. And estimate Wiki-edit-Esperanto dataset and Facebook-wall posts dataset, and the performing of all methods can be observed in Figure 1(b) and Figure 1(c) respectively.

Table 2 Evalution of the accuracy using AUC

	Autonomous system	Wiki-edit-Esperanto	Facebook-wall posts
Tensor	0.895	0.872	0.917
TBNS	0.933	0.927	0.938

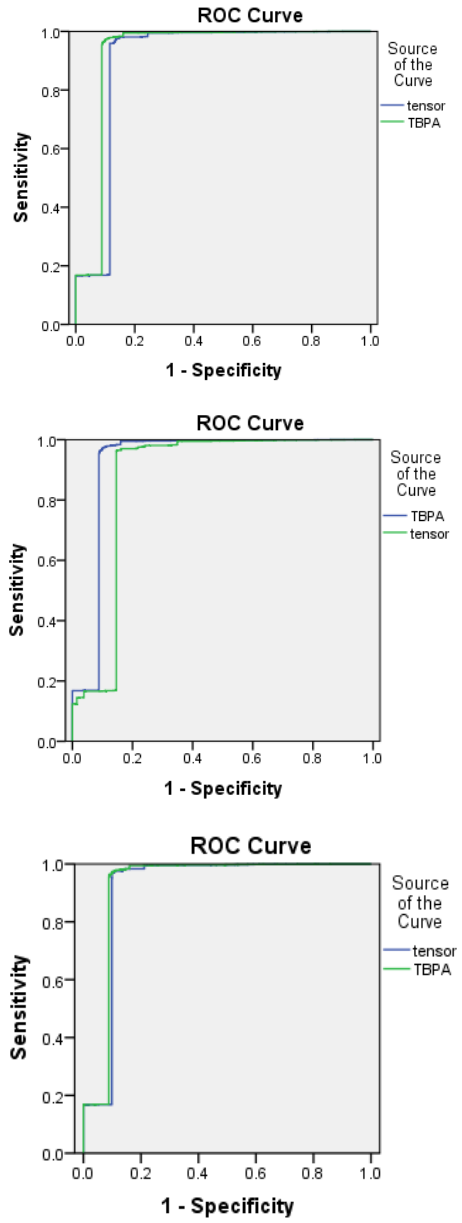

Figure 1 Average ROC curves showing the performance of link prediction methods

We can conclude from the evaluation result of AUC that TBNS have achieved better performance in selected datasets for experiment than algorithm based on CP model. And because TBNS has no calculations of three-dimensional tensors, it possesses higher link prediction efficiency than algorithm based on CP model.

5 Conclusions and Future Work

The paper proposed a novel tensor-based prediction algorithm. Past algorithms concentrate on solving topology of a single network without consideration of dynamic information in evolving network, but the dynamic data is so vital in solving the problem of link prediction. What the TBNS algorithm in our paper differs from the former ones is that it considered and well processed dynamic information in evolving network and solved the problem of time series link prediction.

Recently several methods based on tensor factorization to solve problem of time series link prediction have been proposed, unlike these methods, TBNS link prediction algorithm consists of two independent steps without calculations of three-dimensional tensor, which improves efficiency, and have better performance in experimental dataset than algorithm based on tensor factorization. TBNS is made up of two independent steps: first, store dataset with dynamic information into three-dimensional tensor; next, take advantage of exponential smoothing algorithm to process time series data and obtain the output for prediction, and then utilize CN algorithm to predict the possibility that links will emerge between two objects in the future.

In the Evaluation, we made use of TBNS algorithm to deal with AS, Wiki-edit，Facebook time series dataset, and we achieved satisfying result with higher accuracy than similar algorithms based on tensor factorization, this indicates that TBNS can preferably solve the problem of link prediction of time series data.

In our future work, we will further study the link prediction performance of TBNS in complicated evolving networks with diverse features, and improve its accuracy of link prediction in these networks. Simultaneously, we will study the change of which features in diverse network features has greater effect on accuracy of link prediction, and then research how to adjust it to get higher prediction accuracy in different network structures.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No.61100189, 61003261). We also appreciate the reviewers' precious comments.

References

[1] Leskovec, J. 2009. Networks, communities and kronecker products. Proceeding of the 1st ACM international workshop on Complex networks meet information & knowledge management (2009), 1–2.

[2] Leskovec, J., Kleinberg, J. and Faloutsos, C. 2005. Graphs over time: densification laws, shrinking diameters and possible explanations. Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining (2005), 177–187.

[3] Liben-Nowell, D. and Kleinberg, J. 2007. The link-prediction problem for social networks. Journal of the American society for information science and

technology. 58, 7 (2007), 1019–1031.

[4] Al Hasan, M., Chaoji, V., Salem, S. and Zaki, M. 2006. Link prediction using supervised learning. SDM'06: Workshop on Link Analysis, Counter-terrorism and Security (2006).

[5] Clauset, A., Moore, C. and Newman, M.E.J. 2008. Hierarchical structure and the prediction of missing links in networks. Nature. 453, 7191 (2008), 98–101.

[6] Yar, M. and Chatfield, C. 1990. Prediction intervals for the Holt-Winters forecasting procedure. International Journal of Forecasting. 6, 1 (1990), 127–137.

[7] Hanley, J.A. 1982. The meaning and use of area under a receiver operating characteristic characteristic (ROC) curvel. Radiology. 743, (1982), 29–36.

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

[9] Spiegel, S., Clausen, J., Albayrak, S. and Kunegis, J. 2012. Link prediction on evolving data using tensor factorization. New Frontiers in Applied Data Mining. (2012), 100–110.

[10] Carroll, J.D. and Chang, J.J. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. Psychometrika. 35, 3 (1970), 283–319.

[11] Acar, E. and Yener, B. 2009. Unsupervised multiway data analysis: A literature survey. Knowledge and Data Engineering, IEEE Transactions on. 21, 1 (2009), 6–20.

[12] Kolda, T.G. and Bader, B.W. 2009. Tensor decompositions and applications. SIAM review. 51, 3 (2009), 455.

[13] Newman, M.E.J. 2001. Clustering and preferential attachment in growing networks. Physical Review E. 64, 2 (2001), 025102.

[14] Kossinets, G. 2006. Effects of missing data in social networks. Social networks. 28, 3 (2006), 247–268.