SCIENCE CHINA
Mathematics

• ARTICLES •

January 2017 Vol. 60 No. 1: 165–176
doi: 10.1007/s11425-015-0807-8

A dynamic logistic regression for network link prediction

ZHOU Jing[1], HUANG DanYang[1,*] & WANG HanSheng[2]

[1]*Department of Statistics, Renmin University of China, Beijing* 100872*, China;*
[2]*Department of Statistics, Peking University, Beijing* 100871*, China*
*Email: jing.zhou@pku.edu.cn, dyhuang*89*@*126*.com, hansheng@gsm.pku.edu.cn*

Received December 7, 2015; accepted April 13, 2016; published online October 27, 2016

Abstract In social network analysis, link prediction is a problem of fundamental importance. How to conduct a comprehensive and principled link prediction, by taking various network structure information into consideration, is of great interest. To this end, we propose here a dynamic logistic regression method. Specifically, we assume that one has observed a time series of network structure. Then the proposed model dynamically predicts future links by studying the network structure in the past. To estimate the model, we find that the standard maximum likelihood estimation (MLE) is computationally forbidden. To solve the problem, we introduce a novel conditional maximum likelihood estimation (CMLE) method, which is computationally feasible for large-scale networks. We demonstrate the performance of the proposed method by extensive numerical studies.

Keywords conditional likelihood, dynamic logistic regression, link prediction, social networks

MSC(2010) 05C90, 94C15

Citation: Zhou J, Huang D Y, Wang H S. A dynamic logistic regression for network link prediction. Sci China Math, 2017, 60: 165–176, doi: 10.1007/s11425-015-0807-8

1 Introduction

For the past decades, various online social networks have undergoing a rapid development. It is estimated by Statista (www.statista.com) that there will be around 2.13 billion social network users worldwide in 2016. Popular social networks such as Facebook, Twitter, WeChat and Sina Weibo have attracted hundreds of millions of people to communicate with each other on their platforms. Facebook, for example, has attracted about 1.42 billion active users per month, while that of Twitter is about 0.3 billion. As one can see, social network is becoming an indispensable part of daily lives.

In the meanwhile, social network related business is also rapidly growing. For example, as reported by Facebook annual report, its revenue in 2014 is 12.47 billion US dollars and more than 90% of its revenue is from advertising sales. The reason for such a big advertising value is because there are a huge number of users actively communicating on the platform. Their active communication leads to tremendous amount of opportunities for advertising display. Intuitively, the more active a social network is, the more opportunities for advertising display, and thus more potential advertising revenue. As a result, increasing network activeness is always one of the top priorities for the firms' day-to-day operation.

Practically, for a social network with a given network size, the activeness typically refers to two different perspectives. The first one is a large amount of UGC (user generated content) with good quality. The

*Corresponding author

second one is fast and smooth UGC propagation. For the second perspective, a well connected network structure is essential. To improve network connectivity, one needs to recommend potentially interested friends to each user. In the meanwhile, one also has to prevent existing links from broken. Accurate recommendation may enhance the users' loyalty to the network and prevent customer churn. This makes link prediction a task of great importance [5,18]. Link prediction is a task about estimating the likelihood for a link to exist between two nodes, given the observed network structure [11]. Most existing methods are similarity-based algorithms [1,7,20,25,29]. They make link prediction according to various similarity measures, which are derived from the observed network structure. Then, how to combine various similarity measures under a unified framework is quite a challenging task [21]. See also [17, 30, 30, 32] for some relevant discussions.

To solve the problem, we propose here a dynamic logistic regression method for link prediction. It combines various similarity measures under a unified model framework. Specifically, assume we have a network with size n and its network structure is observed at a sequence of time points indexed by $\{t : 1 \leqslant t \leqslant T\}$. For any two arbitrary nodes i and j, define $a_{ij}^t = 1$ if i follows j and 0 otherwise at time point t. Then the relationships among the nodes over different time points are represented by a sequence of $n \times n$ adjacency matrices $\mathbb{A}_t = (a_{ij}^t)$ (see [16, 24, 34]). We then consider how to make accurate prediction for future link a_{ij}^t by carefully studying the historical network structure information $\mathcal{F}_{t-1} = \sigma\{\mathbb{A}_s : s < t\}$. This leads to a novel method of dynamic logistic regression [14]. The new method can flexibly take various network structure information into consideration.

It is remarkable that the new model allows the network structure to be extremely sparse. Furthermore, by introducing a novel binary random effect, a number of stylized network characteristics (e.g., reciprocation, transitivity) can be well accommodated [12]. However, as a side effect, the resulting likelihood function can be too complicated. This makes the standard maximum likelihood estimation (MLE) computationally forbidden. To alleviate the computational cost, we propose a novel conditional likelihood estimation (CMLE) method, which is computationally feasible for large-scale networks. We demonstrate the efficiency of our model with both simulation studies and a real data example.

The proposed new model makes two contributions to the existing literature. First, it is an interesting and novel time series model. Most classical time series literature focus on univariate time series [10]. Recently, there is an increasing interest in multivariate time series, with particular focus on high-dimensional data [4, 19, 27, 35]. However, to our best knowledge, the research about binary-matrix-valued time series (i.e., $\mathbb{A}_t$) is extremely limited. Second, in network analysis literature, much efforts have been made to understand the mechanism for network formation. The simplest Erdös-Rényi model is a pioneer work in this area. [9] assumed an independent condition for different links. To allow for reciprocal dependence, [13] proposed the p_1 model. Similar idea was further extended by [26,33], so that stochastic block structure can be modeled. Other related literatures include, for example, the exponential random graph model [15] and latent space model [12]. However, all these works focus on cross-sectional data, and none of them investigate network dynamics under a time series framework.

The rest of this paper is organized as follows. Section 2 presents the model setup and the complete likelihood theory. This leads to the conditional likelihood theory and its estimation method. To demonstrate its finite sample performance, numerical studies based on both simulated and real datasets are conducted in Section 3. Lastly, the article is concluded with a short discussion and the limitations in Section 4.

2 The methodology

2.1 Model setup

We consider a network with n nodes, which are denoted as $\{1, \ldots, n\}$. We further assume that the network is consecutively observed on a sequence of equally spaced time points, indexed by $\{1, \ldots, T\}$. The relationships among the nodes over different time points are represented by a sequence of $n \times n$ adjacency matrices $\mathbb{A}_t = (a_{ij}^t)$ in which $a_{ij}^t = 1$ $(i \neq j)$ if there is a link from i to j at time t, and 0 otherwise. For an

undirected network we have $a_{ij}^t = a_{ji}^t$. However, throughout the rest of this article, we consider a directed network structure, which allows $a_{ij}^t \neq a_{ji}^t$. The goal is to model the network dynamic system, i.e., to model the conditional distribution of $\mathbb{A}_t$ given its lagged information $\mathcal{F}_{t-1} = \sigma\{\mathbb{A}_{t-1}, \mathbb{A}_{t-2}, \ldots, \mathbb{A}_0\}$.

In practice, the structure of a large-scale social network is typically extremely sparse. This implies that $P(a_{ij}^t = 1)$ is very small. Mathematically, this amounts to assume that $P(a_{ij}^t = 1) \to 0$ as $n \to \infty$. If one allows a_{ij}^t and a_{ji}^t to be mutually independent, we then should have $P(a_{ij}^t a_{ji}^t = 1)$ to be even smaller. Or equivalently, the conditional probability $P(a_{ij}^t = 1 \mid a_{ji}^t = 1) \to 0$ as $n \to \infty$. On the other side, most empirical studies suggest that the conditional probability $P(a_{ij}^t = 1 \mid a_{ji}^t = 1)$ should be much larger and theoretically should be treated as a fixed number. As a result, the two links (i.e., a_{ij}^t and a_{ji}^t) should not be modeled independently. Instead, they have to be modeled jointly [13].

To this end, define $z_{ij} = z_{ji} \in \{0, 1\}$ to be a binary indicator, which could be treated as a binary random effect. Define $Z = (z_{ij}) \in \mathbb{R}^{n \times n}$, which is a symmetric random effect matrix. We then assume $a_{ij}^t = z_{ij}\tilde{a}_{ij}^t$, where $\tilde{a}_{ij}^t \in \{0, 1\}$ is another independent binary indicator such that

$$P(\tilde{a}_{ij}^t = 1 \mid \mathcal{F}_{t-1}) = \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}, \tag{2.1}$$

$X_{ij}^{t-1} = (X_{ij,1}^{t-1}, \ldots, X_{ij,p}^{t-1})^{\mathrm{T}} \in \mathbb{R}^p$ is a p-dimensional predictor derived from $\mathcal{F}_{t-1}$, and $\beta = (\beta_1, \ldots, \beta_p)^{\mathrm{T}} \in \mathbb{R}^p$ is the associated regression coefficients and the primary parameters of interest. Lastly, define $\tilde{\mathbb{A}}_t = (\tilde{a}_{ij}^t) \in \mathbb{R}^{n \times n}$. As a consequence, we have

$$P(a_{ij}^t = 1 \mid \mathcal{F}_{t-1}) = P(z_{ij} = 1)P(\tilde{a}_{ij}^t = 1 \mid \mathcal{F}_{t-1}) = \alpha_{ij} \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}. \tag{2.2}$$

Because large-scale social network is typically extremely sparse, we should expect $P(a_{ij}^t = 1 \mid \mathcal{F}_{t-1})$ to be small. This implies that $\alpha_{ij} \to 0$ as $n \to \infty$. However, note that $a_{ji}^t = 1$ implies that $z_{ij} = z_{ji} = 1$. Consequently, conditional on $a_{ji}^t = 1$, whether $a_{ij}^t = 1$ is fully determined by $\tilde{a}_{ij}^t$. As a consequence, $P(a_{ij}^t = 1 \mid a_{ji}^t = 1) = \exp(\beta^{\mathrm{T}} X_{ij})/\{1 + \exp(\beta^{\mathrm{T}} X_{ij})\}$ is a fixed number. Thus, the *reciprocity* property is well accommodated.

2.2 Complete likelihood

Based on the above model setup, we can analytically spell out the complete likelihood function as follows:

$$\mathcal{L}(\theta) = \prod_{t=2}^{T} \prod_{i,j} \left[\alpha_{ij} \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}\right]^{a_{ij}^t} \left[\alpha_{ij} \frac{1}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})} + (1 - \alpha_{ij})\right]^{1 - a_{ij}^t}.$$

Then the log likelihood is

$$\begin{aligned} \ell(\theta) = \sum_{t=2}^{T} \sum_{i,j} [&(1 - a_{ij}^t) \log\{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1}) - \alpha_{ij} \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})\} \\ &+ a_{ij}^t \{(\alpha_{ij} + \beta^{\mathrm{T}} X_{ij}^{t-1}) + \log \alpha_{ij}\} - \log\{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})\}], \end{aligned}$$

where $\theta = (\beta^{\mathrm{T}}, \alpha_{ij}, i \neq j)^{\mathrm{T}} \in \mathbb{R}^{p+n(n-1)}$. As one can see, optimizing the above complete log likelihood function is computationally challenging for several reasons.

(1) The summations over t and each node pair (i, j) lead to a total of $Tn(n-1)$ terms to be computed. For a large-scale social network, the sample size n could be a huge number. As a consequence, the scale of $Tn(n-1)$ would be gigantic. This makes the computation cost of $\ell(\theta)$ extremely high.

(2) The dimension of θ (i.e., the parameters need to be optimized) is also extraordinarily high. It is squared order of the network size n. This makes the already difficult computation even more challenging.

(3) Even if one has the demanded computational power, optimizing $\ell(\theta)$ directly is not cost-effective. This is because most large-scale social networks are extremely sparse. As a consequence, most node pairs do not follow each other throughout the entire observational period. This implies that $a_{ij}^t = 0$ for every $1 \leqslant t \leqslant T$ with no variabilities at all. Consequently, their contribution to the log likelihood function should be minimum. However, their associated computational cost is quite significant. This makes optimizing $\ell(\theta)$ cost-ineffective.

As a consequence, this motivates us to further develop a novel estimation method, which is computationally more efficient.

2.3 Conditional likelihood

As we have explained before, network structure $\mathbb{A}_t$ is extremely sparse and $a_{ij}^t = 0$ for most t and (i,j) pairs. Obviously, those unconnected node pairs should carry minimum information about the regression coefficient β. This motivates us to focus on only those pairs which are connected. Specifically, for a given time point $t \in \{2,\ldots,T\}$, we define $\mathcal{S}_{t-1} = \{(i,j) : a_{ij}^{t-1} + a_{ij}^t > 0\}$. In other words, $\mathcal{S}_{t-1}$ collects all the node pairs, which are at least connected either at time $t-1$ or t. Accordingly, for any $(i,j) \in \mathcal{S}_{t-1}$, we can only observe three possibilities. They are, respectively, $(a_{ij}^{t-1}, a_{ij}^t) = (1,0)$, $(0,1)$ and $(1,1)$. Their likelihood function is given by, respectively,

$$\begin{aligned} P\{(a_{ij}^{t-1}, a_{ij}^t) = (1,0)\} &= P(z_{ij} = z_{ji} = 1) P\{(\tilde{a}_{ij}^{t-1}, \tilde{a}_{ij}^t) = (1,0)\} \\ &= \alpha_{ij} \left\{ \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-2})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-2})} \right\} \left\{ \frac{1}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})} \right\}. \end{aligned}$$

Similarly, we can have

$$\begin{aligned} P\{(a_{ij}^{t-1}, a_{ij}^t) = (0,1)\} &= \alpha_{ij} \left\{ \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})} \right\} \left\{ \frac{1}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-2})} \right\}, \\ P\{(a_{ij}^{t-1}, a_{ij}^t) = (1,1)\} &= \alpha_{ij} \left\{ \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-2})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-2})} \right\} \left\{ \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})} \right\}. \end{aligned}$$

Then conditional on $a_{ij}^{t-1} = 1$, the likelihood for $a_{ij}^t = 1$ is given by

$$P(a_{ij}^t = 1 \mid a_{ij}^{t-1} = 1) = \frac{\exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}{1 + \exp(\beta^{\mathrm{T}} X_{ij}^{t-1})}. \tag{2.3}$$

It is remarkable that (2.3) is a standard logistic regression form with a_{ij}^t as the binary response, X_{ij}^{t-1} as the predictor, and conducted on the sample $\mathcal{S}'_{t-1} = \{(i,j) : a_{ij}^{t-1} = 1\}$. Specifically, the log likelihood function that needs to be optimized is

$$\ell^*(\beta) = \sum_{t=2}^{T} \sum_{i,j} [a_{ij}^t \beta^T X_{ij}^{t-1} - \log\{1 + \exp(\beta^T X_{ij}^{t-1})\}].$$

The corresponding estimator is given by $\hat{\beta} = \mathrm{argmax}_\beta \ell^*(\beta)$, which is referred to as the conditional maximum likelihood estimator (CMLE). To numerically compute $\hat{\beta}$, a standard Newton-Raphson algorithm can be used. Specifically, define $\dot{\ell}^*(\beta) \in \mathbb{R}^p$ and $\ddot{\ell}^*(\beta) \in \mathbb{R}^{p \times p}$ to be the first and second order derivatives of $\ell^*(\beta)$ with respect to β, respectively. Define $\hat{\beta}^{(0)} = 0$ be the initial estimator. We then compute $\hat{\beta}^{(s+1)} = \hat{\beta}^{(s)} - [\ddot{\ell}^*\{\hat{\beta}^{(s)}\}]^{-1} \dot{\ell}^*\{\hat{\beta}^{(s)}\}$ for $s > 0$. We update $\hat{\beta}^{(s)}$ till it numerically converges. This leads to the final estimate of β.

3 Numerical studies

3.1 Simulation models

To demonstrate the finite sample performance of the proposed methodology, we present in this subsection three simulation examples. These three examples are similar to each other. The only difference is the way by which Z is generated. In the meanwhile, we can generate network structure as follows. First, we generate $n \times n$ matrix, where each element (denoted by U_{ij}) is independently generated from a standard uniform distribution. We next define $a_{ij}^0 = I(U_{ij} < 30/n)$, where $I(\cdot)$ is the indicator function. This leads to the initial network structure $\mathbb{A}_0 = (a_{ij}^0) \in \mathbb{R}^{n\times n}$. Once Z and $\mathbb{A}_0$ are given, $\tilde{\mathbb{A}}_t$ and $\mathbb{A}_t$ can be generated according to (2.1) and (2.2) sequentially. This leads to the network structure sequence $\{\mathbb{A}_t\}$. It is remarkable that in order to dynamically generate $\tilde{\mathbb{A}}_t$, a number of predictors summarized from the historical network structure need to be constructed, i.e., X_{ij}^{t-1} in (2.1). To this end, three interesting and popularly used network statistics are considered. Their construction details are to be given in the next subsection for illustration purpose. Lastly, the detailed generation process for Z is to be presented according to different simulation examples.

Example 3.1 (Pseudo dyad independence model)**.** Following [13], we generate $Z_0 = (z_{0,ij}) \in \mathbb{R}^{n\times n}$ as follows. We define a dyad as $D_{0,ij} = (z_{0,ij}, z_{0,ji})$ for any $1 \leqslant i < j \leqslant n$. In the dyad independence model, it is also assumed that different $D_{0,ij}$s are independent. To allow for the network sparsity, we define $P(D_{0,ij} = (1,1)) = 20n^{-1}$. This leads to the expected number for the mutually connected dyad (i.e., $D_{0,ij} = (1,1)$) is of $O(n)$. In the next step, set

$$P(D_{0,ij} = (1,0)) = P(D_{0,ij} = (0,1)) = 0.5n^{-0.8}.$$

Accordingly, we should have $P(D_{0,ij} = (0,0)) = 1 - 20n^{-1} - n^{-0.8}$, which is very close to 1 for large n. Recall that $z_{ij} = z_{ji}$, thus we let $z_{ij} = I(z_{0,ji} + z_{0,ij} > 0)$. This leads to a symmetric Z. Define $d_i = \sum_{j=1}^n z_{ji}$ to be the meet in-degree of node i in Z. See Figure 1 for an example of meet in-degree histogram in the pseudo dyad independence model in one replication. We fix $T = 10$, $T = 15$ and $T = 20$, respectively in this example.

Example 3.2 (Pseudo stochastic block model)**.** We next consider the structure generated from the stochastic block model [26, 33], which is another popular network topology. Define $K \in \{10, 20, 50\}$ to be the total number of blocks and fix $T = 20$. Specifically, we follow [26], and randomly assign a block label ($k = 1, \ldots, K$) for each node with equal probability $1/K$. Next, let $P(z_{0,ij} = 1) = 0.03n^{-0.03}$ if i and j are in the same block, and $P(z_{0,ij} = 1) = 0.01n^{-1}$ otherwise. Correspondingly, the nodes in the same block are more likely to meet each other compared with nodes from different blocks. Let $z_{ij} = I(z_{0,ji} + z_{0,ij} > 0)$ and thus we obtain Z. See Figure 2 for the histogram of meet in-degree in one replication under the pseudo stochastic block model.

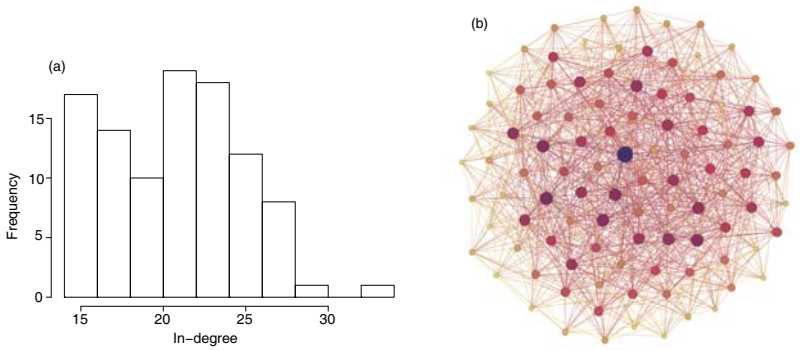

Figure 1 A particular example of the pseudo dyad independence model. (a) Histogram of in-degree in the pseudo dyad independence mode. (b) Visualization of the pseudo dyad independence model

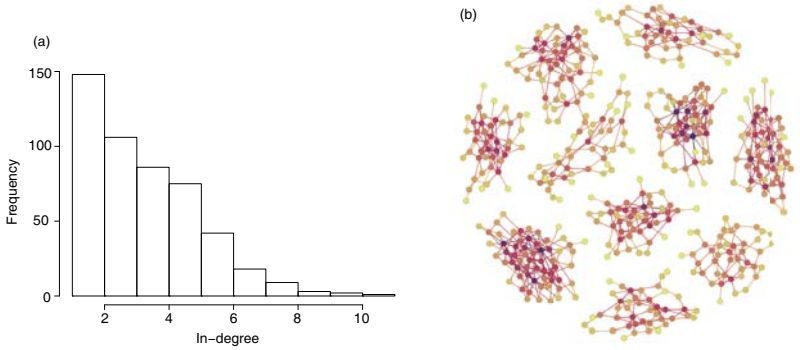

Figure 2 A particular example of the pseudo stochastic block model. (a) Histogram of in-degree in the pseudo stochastic block model. (b) Visualization of the pseudo stochastic block model

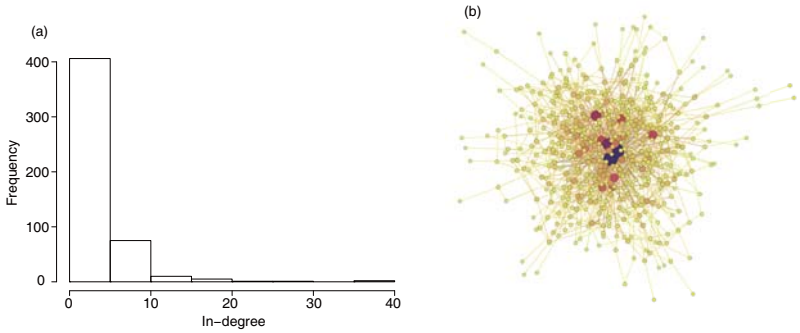

Figure 3 A particular example of the pseudo power-law distribution model. (a) Histogram of in-degree in the pseudo power-law distribution model. (b) Visualization of the pseudo power-law distribution model

Example 3.3 (Pseudo power-law distribution)**.** According to [2], a power-law distribution reflects a popular network phenomenon. The majority of nodes have few edges but a small amount have huge number of edges. We follow [8] and simulate $Z_0 = (z_{0,ij}) \in \mathbb{R}^{n\times n}$ as follows. First, for each node, $d_{0,i} = \sum_j z_{0,ji}$ is generated according to the discrete power-law distribution, which is $P(d_{0,i} = k) = ck^{-\alpha}$, c is a normalizing constant and the exponent parameter $\alpha \in \{2.0, 2.3, 2.5\}$. It is remarkable that smaller α value implies the heavier distribution tail. Second, for the i-th node, $d_{0,i}$ nodes are randomly selected to be node i's potential friends to meet. Correspondingly, define $z_{ij} = I(z_{0,ji} + z_{0,ij} > 0)$ and thus one could get Z. The histogram of meet in-degree in one replication under the pseudo power-law distribution is presented in Figure 3. We fix $T = 20$ in this example.

3.2 Three network statistics

As we have discussed before, in order to simulate $\tilde{\mathbb{A}}_t$ by (2.1), we need to construct a predictor vector X_{ij}^t. For illustration purpose, we consider here three (i.e., $p = 3$) popularly used network statistics. Specifically, define $\Gamma_i^t = \{j : a_{ij}^t = 1\}$ to be the set of nodes that i follows at time point t, $\Gamma_i'^t = \{j : a_{ji}^t = 1\}$ to be the set of nodes those follow i at time point t. We then define

$$X_{ij}^t = (X_{ij,1}^t, X_{ij,2}^t, X_{ij,3}^t)^{\mathrm{T}} \in \mathbb{R}^3$$

as follows:

(1) The momentum effect. Past empirical research suggests that people tend to behave in accordance to what they have done before [3, 6]. As a result, we define a predictor that indicates the previous link status between two nodes, which is

$$X_{ij,1}^t = a_{ij}^t. \tag{3.1}$$

Thus we know the corresponding coefficient represents a momentum effect.

(2) The homophily effect. Classic homophily theory in sociology field suggests that people who are connected with each other are more likely to have similar preference [23, 31]. For two arbitrary nodes i and j, this implies that we can recommend i's friends to j. Because grounded on the theory of homophily, i and j should be similar to some extent. Technically, we define $X^t_{ij,2}$ to be the number of pathways of length two between node i and node j. For example, a typical indirect connection between i and j with length 2 could be $i \to k \to j$ for some $1 \leqslant k \leqslant n$, i.e.,

$$X^t_{ij,2} = \sum_{k=1}^{n} a^t_{ik} a^t_{kj} = |\Gamma^t_i \cap \Gamma'^t_j|, \tag{3.2}$$

where $|\Gamma^t_i \cap \Gamma'^t_j|$ denotes the total number in the set $\Gamma^t_i \cap \Gamma'^t$. It is remarkable that $X^t_{ij,2}$ is just the corresponding element of $\mathbb{A}^2_t$.

(3) The common factor effect. Previous study also suggests that people can be influenced by some common exogenous factors [22]. Practically, the number of mutual friends is always an important factor in assessing link formation. So the last predictor we examine here is the number of mutual friends between i and j. For example, $i \to k$ and $j \to k$ for some $1 \leqslant k \leqslant n$. Define

$$X^t_{ij,3} = \sum_{k=1}^{n} a_{ik} a_{jk} = |\Gamma^t_i \cap \Gamma^t_j|. \tag{3.3}$$

Notice that $X^t_{ij,3}$ is the corresponding element of $\mathbb{A}_t \mathbb{A}^{\mathrm{T}}_t$.

It is remarkable that other network features could also be adopted in the unified dynamic logistic model. For example, *reverse homophily effect,* which is defined as

$$X^t_{ij,4} = \sum_{k=1}^{n} a^t_{jk} a^t_{ki} = |\Gamma^t_j \cap \Gamma'^t_i|$$

and *common follower effect,* which is

$$X^t_{ij,5} = \sum_{k=1}^{n} a^t_{kj} a^t_{ki} = |\Gamma'^t_i \cap \Gamma'^t_j|.$$

Higher order network characteristics could also be involved. Once $\hat{\beta}$ is obtained, we can construct a conditional likelihood index (CLI) based on the network features at time point t,

$$\mathrm{CLI}^t(i,j) = \frac{\exp(\hat{\beta}^{\mathrm{T}} X^{t-1}_{ij})}{1 + \exp(\hat{\beta}^{\mathrm{T}} X^{t-1}_{ij})}. \tag{3.4}$$

We can then predict $a^t_{ij} = 1$ if $\mathrm{CLI}^t(i,j) > c$ for some pre-specified threshold value c. Different choice of c leads to different true and false positive rates, which are to be comprehensively evaluated by AUC (see [28]) in Subsection 3.4.

3.3 Estimation results

We fix

$$(\beta_1, \beta_2, \beta_3)^{\mathrm{T}} = (1.0, 0.3, 0.8)^{\mathrm{T}}$$

throughout the whole simulation studies. For each simulation example, various network sizes ($n = 1,000, n = 5,000$ and $n = 10,000$) are considered, and the experiment is randomly replicated for $M = 500$ times. Let

$$\hat{\beta}^{(m)} = (\hat{\beta}^{(m)}_k)^{\mathrm{T}} = (\hat{\beta}^{(m)}_1, \hat{\beta}^{(m)}_2, \hat{\beta}^{(m)}_3)^{\mathrm{T}}$$

be the estimator obtained in the m-th ($m \in \{1, \ldots, M\}$) replication. We consider two measures to gauge the finite sample performance. First, for a given parameter β_k with $1 \leqslant k \leqslant 3$, the root mean square error is evaluated by

$$\mathrm{RMSE}_k = \left\{ M^{-1} \sum_{m=1}^{M} (\hat{\beta}_k^{(m)} - \beta_k)^2 \right\}^{1/2}.$$

Second, for each $1 \leqslant k \leqslant 3$, a 95% confidence interval is constructed for β_k as

$$\mathrm{CI}_k^{(m)} = (\hat{\beta}_k^{(m)} - z_{0.975}\widehat{\mathrm{SE}}_k^{(m)}, \hat{\beta}_k^{(m)} + z_{0.975}\widehat{\mathrm{SE}}_k^{(m)}),$$

where $\widehat{\mathrm{SE}}_k^{(m)}$ is the estimated standard error in the m-th replication, and z_α is the αth quantile of a standard normal distribution. Then, the coverage probability is computed as

$$\mathrm{CP}_k = M^{-1} \sum_{m=1}^{M} I(\beta_k \in \mathrm{CI}_k^{(m)}).$$

We report the estimation results of RMSE_k and CP_k for parameter β_k in each example. Additionally, we report the average CPU time for each replication (CPU in our study is 3.2GHz). See Tables 1 –3 for detailed results respectively. Consider for example Table 1. We find that, when n is fixed but T increases, the RMSE values decrease. This is expected because in this case larger T leads to large sample size. Similarly, if T is fixed, then RMSE value drops as n increases. Furthermore, the reported coverage probabilities (i.e., CP) are always close to their nominal level 95%. This suggests that the estimated standard error (i.e., $\widehat{\mathrm{SE}}$) should approximate the true SE well. Similar results are obtained for Examples 3.2 and 3.3 from Tables 2 and 3. Lastly, by Figure 4, we find that the CPU time demanded by our method increases as the network size n increases. Its increasing pattern is approximately linear.

3.4 Link prediction results

For illustration purpose, CLI is compared with four well known local similarity indices listed in Table 4. Those indices were developed for link prediction. They have been popularly used in practice due to their computational feasibility for large-scale social networks. Consider for example the first index,

Table 1 Simulation results for Example 3.1 with 500 replications. The RMSE values ($\times 10^{-2}$) are reported for every β estimates. The CP (in %) of every estimate is given in parentheses. The average CPU computation time (s) is reported

	$T = 10$			$T = 15$			$T = 20$		
	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$
β_1	0.75 (94.2)	0.29 (94.6)	0.20 (95.4)	0.59 (94.8)	0.23 (95.0)	0.16 (94.0)	0.53 (92.2)	0.19 (95.6)	0.14 (92.0)
β_2	3.12 (95.0)	2.84 (92.6)	2.56 (96.8)	2.55 (94.0)	2.18 (95.2)	2.12 (94.4)	2.22 (94.0)	1.86 (95.0)	1.78 (96.2)
β_3	3.18 (94.8)	2.84 (95.8)	2.72 (94.8)	2.59 (95.0)	2.30 (95.4)	2.13 (95.6)	2.24 (95.0)	1.94 (94.4)	1.82 (94.4)
Time	0.1753	0.8683	1.8687	0.2765	1.4386	2.8475	0.3690	1.9012	3.8898

Table 2 Simulation results for Example 3.2 with 500 replications. The RMSE values ($\times 10^{-2}$) are reported for every β estimates. The CP (in %) of every estimate is given in parentheses.The average CPU computation time (s) is reported

	$K = 10$			$K = 20$			$K = 50$		
	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$
β_1	0.91 (94.8)	0.37 (95.0)	0.37 (96.4)	1.12 (95.6)	0.32 (95.4)	0.28 (93.8)	1.77 (93.0)	0.39 (95.6)	0.24 (94.2)
β_2	3.57 (95.6)	0.65 (95.6)	0.59 (94.0)	6.64 (96.4)	0.85 (94.2)	0.47 (95.0)	16.1 (95.0)	1.62 (94.2)	0.68 (94.0)
β_3	3.70 (96.6)	0.66 (95.0)	0.58 (93.8)	6.76 (95.4)	0.82 (94.8)	0.45 (96.6)	17.2 (95.4)	1.62 (95.6)	0.68 (94.4)
Time	0.1031	3.9293	16.9881	0.0515	1.6816	8.3904	0.0221	0.6067	3.0084

Table 3 Simulation results for Example 3.3 with 500 replications. The RMSE values ($\times 10^{-2}$) are reported for every β estimates. The CP (in %) of every estimate is given in parentheses. The average CPU computation time (s) is reported

	$\alpha = 2.0$			$\alpha = 2.3$			$\alpha = 2.5$		
	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$	$n = 1{,}000$	$n = 5{,}000$	$n = 10{,}000$
β_1	1.06 (95.2)	0.42 (95.4)	0.28 (94.2)	1.00 (95.6)	0.44 (93.2)	0.28 (96.0)	1.08 (95.0)	0.47 (94.2)	0.33 (93.8)
β_2	3.97 (93.4)	1.58 (94.8)	1.06 (97.0)	9.07 (96.4)	5.49 (95.6)	4.52 (95.0)	16.3 (94.0)	11.4 (95.6)	9.94 (95.8)
β_3	3.98 (94.0)	1.61 (95.8)	1.13 (96.0)	8.74 (95.8)	5.89 (95.4)	4.72 (94.8)	16.2 (95.0)	12.4 (96.0)	9.98 (96.2)
Time	0.1957	1.0381	2.5820	0.0773	0.4377	0.9370	0.0547	0.3210	0.6474

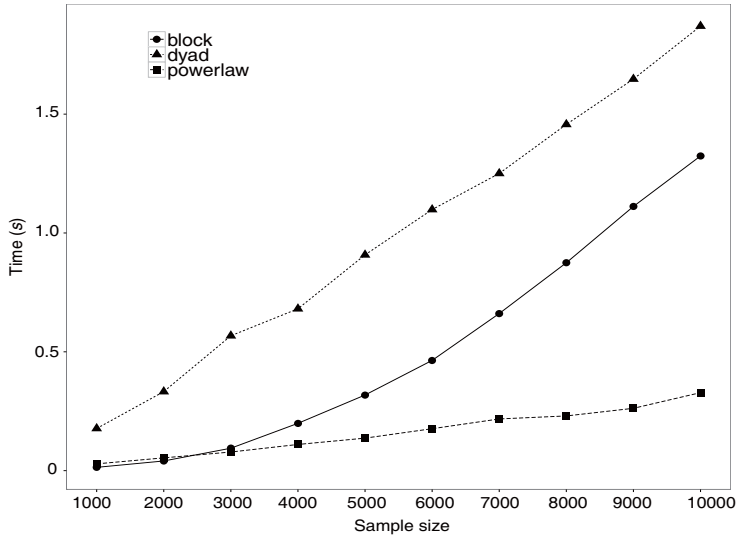

Figure 4 Average CPU time for each example across different sample size. The triangular line corresponds to Example 3.1 with $T = 10$. The circle line corresponds to Example 3.2 with $K = 50$ and the square line corresponds to Example 3.3 with $\alpha = 2.5$

Table 4 Local similarity indices

Method	Index
Common neighbors index (CNI)	$\lvert\Gamma_i^T \cap \Gamma_j'^T\rvert$
Salton index (SI)	$\frac{\lvert\Gamma_i^T \cap \Gamma_j'^T\rvert}{\sqrt{d_i^T \times d_j^T}}$
Hub promoted index (HPI)	$\frac{\lvert\Gamma_i^T \cap \Gamma_j'^T\rvert}{\min\{d_i^T, d_j^T\}}$
Hub depressed index (HDI)	$\frac{\lvert\Gamma_i^T \cap \Gamma_j'^T\rvert}{\max\{d_i^T, d_j^T\}}$

i.e., the common neighbors index (CNI). By Table 4, CNI is defined to be $|\Gamma_i^T \cap \Gamma_j'^T|$. Recall that $\Gamma_i^T = \{j : a_{ij}^T = 1\}$ is defined to be the set of nodes that i follows, and $\Gamma_i'^T = \{j : a_{ji}^T = 1\}$ is defined to be the set of nodes those follow i at time point T. Thus, CNI counts the number of common followees between i and j. Then, all the (i, j)-pairs should be sorted according to their CNI values in a descending order. Those pairs with top CNI values should be predicted as 1, while the rest to be 0. Other indices are various modifications about CNI, according to their in-degree $d_i^T = \sum_j a_{ji}^T = |\Gamma_i'^T|$. For a more detailed discussion about various local similarity indices, we refer to [21].

For each simulation replication, n is fixed to be 1,000. We use data from time points 1 to $T - 1$ to estimate CMLE $\hat{\beta}$. Its prediction accuracy is then evaluated by data from time point T in terms of AUC (see [28]). To save computation time, only those pairs with $a_{ij}^{T-1} = 1$ are used for prediction evaluation. The resulting AUC values are average across different simulation replications, and then reported in Table 5. We find that CLI performs better than its competitors in terms of AUC. This is reasonable because CLI is a more comprehensive estimate and takes information from different sources into consideration.

Table 5 AUC results in the scenario of $a_{ij}^{T-1}=1$ in different examples

Method	CLI	CNI	HDI	HPI	SI
Example 3.1					
$T=10$	**0.5786**	0.5612	0.5612	0.5611	0.5611
$T=15$	**0.5785**	0.5631	0.5630	0.5630	0.5630
$T=20$	**0.5782**	0.5640	0.5639	0.5639	0.5639
Example 3.2					
$K=10$	**0.5836**	0.5681	0.5677	0.5457	0.5459
$K=20$	**0.5482**	0.5385	0.5384	0.5112	0.5113
$K=50$	**0.5211**	0.5164	0.5163	0.5071	0.5071
Example 3.3					
$\alpha=2.0$	**0.6288**	0.6173	0.5767	0.5600	0.5450
$\alpha=2.3$	**0.5455**	0.5389	0.5346	0.5137	0.5134
$\alpha=2.5$	**0.5209**	0.5173	0.5162	0.5074	0.5073

Table 6 Real data results with $T=20$

Estimation			
Coefficient	Estimate	$\widehat{\mathrm{SE}}$	P-value
Momentum effect	7.3633	0.0464	0
Homophily effect	0.4373	0.0269	0
Common factor effect	−0.0108	0.0241	0.6549
Reverse homophily effect	0.3045	0.0238	0
Common follower effect	−0.2606	0.0247	0

Prediction: Conditional on $a_{ij}^{T-1}=1$					
Method	CLI	CNI	HDI	HPI	SI
$\widehat{\mathrm{AUC}}$	0.7063	0.6822	0.6162	0.5003	0.5003

Prediction: Conditional on $a_{ij}^{T-1}=0$					
Method	CLI	CNI	HDI	HPI	SI
$\widehat{\mathrm{AUC}}$	0.8160	0.8136	0.7033	0.6632	0.6481

3.5 Sina Weibo network analysis

We next illustrate the performance of the proposed method by a real data analysis example. The data are collected from Sina Weibo (www.weibo.com), which can be viewed as a Twitter-type social media in Chinese. For illustration purpose, our dataset contains $n=8{,}591$ active followers of an official Weibo account. Their follower-followee relationships within these n nodes are also recorded for a total of $T=20$ days. The detailed estimation and prediction results are given in Table 6. Except for the proposed three useful network statistics in the simulation, we also include two more effects. They are, respectively, *reverse homophily effect*, and *common follower effect* as defined in the previous section. By Table 6, we find that all of the coefficients are significant except common factor effect. Moreover, the momentum effect is the most important factor. Lastly, the CLI method always predicts best in terms of AUC. However, its relative advantage over CNI is practically ignorable, if $a_{ij}^{T-1}=0$.

4 Conclusion

To conclude the paper, we discuss here a number of interesting topics for future research. First, in order to predict $\mathbb{A}_t$, only network statistics derived from $\mathbb{A}_{t-1}$ are numerical studied in this work. Higher order network features (e.g., statistics derived based on $\mathbb{A}_{t-1}$ and $\mathbb{A}_{t-2}$) are to be investigated in the future. Second, we find CLI performs the best in presence of strong momentum effect. This is true if the network structure changes slowly. However, for rapidly changed network structure (e.g., a telecommunication

network), whether CLI can still perform competitively is not clear at this moment. Lastly, we assume here that β is fixed across different time points. In fact, one can reasonably expect that β should dynamically change as t increases. Then, how to model such a time series dynamics for β is another interesting topic for future study.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 11131002, 11271031, 71532001, 11525101, 71271210 and 714711730), the Business Intelligence Research Center at Peking University, the Center for Statistical Science at Peking University, the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (Grant No. 16XNLF01), Ministry of Education Humanities Social Science Key Research Institute in University Foundation (Grant No. 14JJD910002), the Center for Applied Statistics, School of Statistics, Renmin University of China and China Postdoctoral Science Foundation (Grant No. 2016M600155).

References

1 Adamic L A, Adar E. Friends and neighbors on the web. Soc Netw, 2003, 25: 211–230
2 Barabasi A L, Albert R. Emergence of scaling in random networks. Science, 1999, 286: 509–512
3 Bawa K. Modeling inertia and variety seeking tendencies in brand choice behavior. Market Sci, 1990, 9: 263–278
4 Bedrick E J, Tsai C-L. Model selection for multivariate regression in small samples. Biometrics, 1994, 50: 226–231
5 Butts C T. Network inference, error, and informant (in) accuracy: A bayesian approach. Soc Netw, 2003, 25: 103–140
6 Chintagunta P K. Inertia and variety seeking in a model of brand-purchase timing. Market Sci, 1998, 17: 253–270
7 Chowdhury G. Introduction to Modern Information Retrieval. London: Facet Publishing, 2010
8 Clauset A, Shalizi C R, Newman M E. Power-law distributions in empirical data. SIAM Rev, 2009, 51: 661–703
9 Erdös P, Rënyi A. On the evolution of random graphs. Publ Math Inst Hung Acad Sci, 1960, 5: 17–61
10 Fan J, Yao Q. Nonlinear Time Series: Nonparametric and Parametric Methods. New York: Springer, 2003
11 Getoor L, Diehl C P. Link mining: A survey. ACM Sigkdd Explor Newslett, 2005, 7: 3–12
12 Hoff P D, Raftery A E, Handcock M S. Latent space approaches to social network analysis. J Amer Statist Assoc, 2002, 97: 1090–1098
13 Holland P W, Leinhardt S. An exponential family of probability distributions for directed graphs. J Amer Statist Assoc, 1981, 76: 33–50
14 Hunter D R, Goodreau S M, Handcock M S. Goodness of fit of social network models. J Amer Statist Assoc, 2008, 103: 248–258
15 Hunter D R, Handcock M S, Butts C T, et al. Ergm: A package to fit, simulate and diagnose exponential-family models for networks. J Statist Softw, 2008, 24: 54860
16 Knoke D, Yang S. Social Network Analysis. Encyclopedia Soc Netw Anal Min, 2008, 22: 109–127
17 Koren Y. Collaborative filtering with temporal dynamics. ACM Sigkdd Int Conf Knowl Discov Data Min, 2009, 53: 89–97
18 Kossinets G. Effects of missing data in social networks. Soc Netw, 2006, 28: 247–268
19 Lam C, Yao Q. Factor modeling for high-dimensional time series: Inference for the number of factors. Ann Statist, 2012, 40: 694–726
20 Leicht E A, Holme P, Newman M E. Vertex similarity in networks. Phys Rev E, 2006, 73: 026120
21 Lü L, Zhou T. Link prediction in complex networks: A survey. Phys A, 2011, 390: 1150–1170
22 Ma L, Krishnan R, Montgomery A L. Latent homophily or social influence? An empirical analysis of purchase within a social network. Manag Sci, 2014, 61: 454–473
23 McPherson J M, Smith-Lovin L. Homophily in voluntary organizations: Status distance and the composition of face-to-face groups. Amer Soc Rev, 1987, 52: 370–379
24 Newman M. Networks: An Introduction. Oxford: Oxford University Press, 2010
25 Newman M E. Clustering and preferential attachment in growing networks. Phys Rev E, 2001, 64: 025102
26 Nowicki K, Snijders T A B. Estimation and prediction for stochastic blockstructures. J Amer Statist Assoc, 2001, 96: 1077–1087
27 Pan J, Yao Q. Modelling multiple time series via common factors. Biometrika, 2008, 95: 365–379
28 Pepe M S, Cai T, Longton G. Combining predictors for classification using the area under the receiver operating characteristic curve. Biometrics, 2006, 62: 221–229
29 Ravasz E, Somera A L, Mongru D A, et al. Hierarchical organization of modularity in metabolic networks. Science, 2002, 297: 1551–1555
30 Richard E, Gaiffas S, Vayatis N. Link prediction in graphs with autore gressive features. J Mach Learn Res, 2014, 15: 489–517

31 Shalizi C R, Thomas A C. Homophily and contagion are generically confounded in observational social network studies. Sociol Methods Res, 2011, 40: 211–239
32 Vu D Q, Hunter D, Smyth P, et al. Continuous-time regression models for longitudinal networks. Adv Neural Inf Process Syst, 2012, 24: 2492–2500
33 Wang Y J, Wong G Y. Stochastic blockmodels for directed graphs. J Amer Statist Assoc, 1987, 82: 8–19
34 Wasserman S, Faust K. Social Network Analysis: Methods and Applications. Cambridge: Cambridge University Press, 1994
35 Yuan M, Ekici A, Lu Z, et al. Dimension reduction and coefficient estimation in multivariate linear regression. J Roy Statist Soc Ser B, 2007, 69: 329–346