Intelligent Data Analysis 15 (2011) 633–652
DOI 10.3233/IDA-2011-0486
IOS Press

Learning information diffusion model in a social network for predicting influence of nodes

Masahiro Kimura[a,*], Kazumi Saito[b], Kouzou Ohara[c] and Hiroshi Motoda[d]
[a]*Department of Electronics and Informatics, Ryukoku University, Seta, Otsu, Japan*
[b]*School of Administration and Informatics, University of Shizuoka, Shizuoka, Japan*
[c]*Department of Integrated Information Technology, Aoyama Gakuin University, Kanagawa, Japan*
[d]*Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan*

Abstract. We address the problem of estimating the parameters, from observed data in a complex social network, for an information diffusion model that takes time-delay into account, based on the popular independent cascade (IC) model. For this purpose we formulate the likelihood to obtain the observed data which is a set of time-sequence data of infected (active) nodes, and propose an iterative method to search for the parameters (time-delay and diffusion) that maximize this likelihood. We first show by using a synthetic network that the proposed method outperforms the similar existing method. Next, we apply this method to problems of both 1) predicting the influence of nodes for the considered information diffusion model and 2) ranking the influential nodes. Using three large social networks, we demonstrate the effectiveness of the proposed method.

Keywords: Social network analysis, information diffusion model with time-delay, parameter learning, influence degree prediction, node ranking

1. Introduction

Investigating the structure and function of complex networks such as biochemical and social networks is a hot research subject [2,5,15]. From a functional view, innovation, topics and even malicious rumors can propagate through social networks among people in the form of so-called "word-of-mouth" communications. Therefore, considerable attention has recently been devoted to social networks as an important medium for the spread of information [4,7,13,14,21].

There are several models that simulate information diffusion through a network. A widely-used fundamental probabilistic model is the *independent cascade (IC) model* [6,8], which can be regarded as the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease [15]. This model has been used to solve the problem of finding a limited number of nodes that are influential for the spread of information [8,9]. This combinatorial optimization problem is called the *influence maximization problem*, and is one of the important application problems in sociology and "viral marketing" [1]. Further, this model has been used to solve yet another problem of minimizing the spread of undesirable

*Corresponding author: Department of Electronics and Informatics, Ryukoku University, Seta, Otsu 520-2194, Japan. Tel.: +81 77 543 7406; Fax: +81 77 543 7428; E-mail: kimura@rins.ryukoku.ac.jp.

information by blocking a limited number of links in a social network [10], and to visualize a complex network in terms of information flow [19]. The IC model requires the parameters that represent diffusion probabilities through links to be specified in advance. However, the true values of the parameters are not available in practice. Thus, it is an important research issue to develop a method that can efficiently estimate them.

One of the drawbacks of the IC model is that it cannot represent time-delay for information propagation. Consider, as an example, modeling the day-by-day spread of a topic in a blog network in which blog authors are connected to each other as is done in [7]. Here, a topic is a URL or phrase that can be tracked down from blog to blog. Suppose that there are blogroll links from two blog authors v and w to another blog author u. This means that v and w are readers of the blog of u. Suppose that both v and w publish posts on the topic that was addressed in u's post (meaning that v and w are both infected by u). There can be a difference between the dates that v and w publish their posts about the topic. Thus, Gruhl et al. [7] incorporated time-delay into the IC model. We refer to their model as the *ICTD model.*[1] Note that the ICTD model includes the IC model as a special case. The ICTD model is equipped with the parameters that represent time-delay through links as well as the parameters that represent diffusion probabilities through links.

They presented a method for estimating the values of these parameters from observed information diffusion results using an EM-like algorithm, and experimentally showed its effectiveness using sparse Erdös-Renyi networks. Here we note that large real social networks generally include dense subgraphs. For example, Newman and Park [16] observed that social networks represented as undirected graphs have high clustering coefficients, and positive correlations between the degrees of adjacent nodes. In these realistic networks it is important that the diffusion model explicitly addresses the possibility that a node is activated simultaneously by its multiple parent nodes each of which may have become activated at different time in the past. Their method [7], however, ignores this phenomenon and we speculate that their method does not perform well for dense networks, which is experimentally demonstrated in Section 6.2. In addition, it is not clear what they are optimizing in deriving the update formulas of the parameter values. We have already developed a method in [11] for estimating the values of the parameters in case of the IC model. The problem was much simpler because of no time-delay.

In this paper, we extend it to the ICTD model and propose a novel method for estimating the values of the parameters from a set of information diffusion results that are observed as time-sequences of infected (active) nodes. What makes this problem difficult is that incorporating time-delay makes the time-sequence observation data structural. There is no way of knowing from the data which node activated which other node that comes later in the sequence. Further, as the time progresses, the possible activation states increases exponentially, which also makes computation intractable. We introduce an objective function that represents the likelihood of obtaining the observed data sequences under the ICTD model on a given network, and derive an iterative algorithm by which the objective function is maximized. We experimentally show using both one synthetic and three real world networks that the proposed method outperforms the method by Gruhl et al. [7] for finding the correct parameters. We further show that it is crucially important that the parameters are estimated as accurately as possible in order to correctly predict the influence of nodes by which to rank and extract influential nodes.

Our contribution is that 1) we derived an algorithm in a principled way that guarantees convergence, avoids combinatorial explosion and can learn efficiently, from observed data, the parameters for the ICTD model, an information diffusion model that allows asynchronous time-delay and accounts for multiple

[1] It means the "IC with time-delay" model.

activation of a node, and 2) we showed that the algorithm performs satisfactorily for both synthetic and real social networks and outperforms the eGGLT method, a slightly modified version of [7], and a poor performance of the eGGLT method for a dense network is attributed to the ignorance of the multiple activation. We further point out that 3) the ranking method based on the proposed algorithm can be interpreted as a new concept of centrality based on information diffusion.

2. Information diffusion model and learning problem

We first recall the definition of the IC model according to [8], and then define the ICTD model by Gruhl et al. [7]. After that, we formulate our learning problem.

In this paper, we consider mathematical models for the spread of information through a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively.

If there is a link (u, v) from node u to node v, node v is called a *child node* of node u and node u is called a *parent node* of node v. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively, i.e.,

$$F(v) = \{w \in V;\ (v, w) \in E\},$$
$$B(v) = \{u \in V;\ (u, v) \in E\}.$$

We call nodes *active* if they have been influenced with the information. In the information diffusion model, the diffusion process unfolds in a discrete time-step $t \geqslant 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial active node v_0, we assume that the node v_0 has become active at time-step 0, and all the other nodes are inactive at time-step 0.

2.1. IC model

Here we define the IC model. In this model, for each link (u, v), we specify a real value $\kappa_{u,v}$ with $0 < \kappa_{u,v} < 1$ in advance, where $\kappa_{u,v}$ is referred to as the *diffusion probability* through link (u, v). The diffusion process proceeds from a given initial active node v_0 in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2. ICTD model

Gruhl et al. [7] extended the IC model so as to allow time-delay, and presented an information diffusion model with time-delay on network G. Now we call their model, the *ICTD model*.

In the ICTD model, for each link $(u, v) \in E$, we specify real values $r_{u,v}$ and $\kappa_{u,v}$ with

$$0 < r_{u,v},\ \kappa_{u,v} < 1$$

in advance. Gruhl et al. [7] considered modeling the spread of a topic in a blog network, where blog authors are connected by a directed network. The parameter $r_{u,v}$ models how early author v reads the

blog posts of author u, and the parameter $\kappa_{u,v}$ models the probability that author v, after reading the blog post of author u, publishes a blog post about the topic that author u addressed. Thus, $r_{u,v}$ and $\kappa_{u,v}$ were called the reading and the copy probabilities through link (u, v), respectively. In this paper, we refer to $r_{u,v}$ and $\kappa_{u,v}$ as the *time-delay parameter* and the *diffusion parameter* through link (u, v), respectively.

The diffusion process of the model proceeds from a given initial active node v_0 in the following way. Suppose that a node u becomes active at time-step t. Then, node u is given a single chance to activate each currently inactive child node v. We choose a delay-time δ from the geometric distribution with parameter $r_{u,v}$. If node v is not active at time-step $t + \delta$, then node u attempts to activate node v at time-step $t + \delta$, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time-step $t + \delta + 1$. If multiple parent nodes of v attempt to activate v at time-step $t + \delta$, then their activation attempts are sequenced in an arbitrary order. Whether or not u succeeds at time-step $t + \delta$, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.3. *Learning problem*

For the ICTD model on network G, we define the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$ by

$$\boldsymbol{r} = (r_{u,v})_{(u,v)\in E}, \quad \boldsymbol{\kappa} = (\kappa_{u,v})_{(u,v)\in E}.$$

In practice, the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ are not available. Thus, we must estimate them from past information diffusion histories observed as sets of active nodes.

We suppose that

$$\mathcal{D}_M = \{D_m; m = 1, \cdots, M\}$$

is an observed data set of M independent information diffusion results. Here, each D_m is a time-sequence of active nodes for the mth information diffusion result,

$$D_m = \langle D_m(0), D_m(1), \cdots, D_m(T_m)\rangle,$$

where $D_m(t)$ is the set of all the nodes that have first become active at time-step t, and $T_m(\geqslant 1)$ is the observed final time-step. We set

$$C_m(t) = D_m(0) \cup \cdots \cup D_m(t).$$

Note that $C_m(t)$ is the set of active nodes at time-step t for the mth information diffusion result. Note also that $C_m(T_m)$ is the set of all the active nodes for the mth information diffusion result. For any $v \in C_m(T_m)$, let $t_{m,v}$ denote the time-step at which node v becomes active for the mth information diffusion result, that is,

$$v \in D_m(t_{m,v}).$$

In this paper, we consider the problem of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$.

3. Proposed method

Here, we explain how we estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$.

3.1. Likelihood function

For the learning problem described above, we derive the likelihood function $\mathcal{L}(\boldsymbol{r}, \kappa; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and κ for use as our objective function in a rigorous manner.

First, we consider any node $v \in C_m(T_m)$ with $t_{m,v} > 0$ for the mth information diffusion result. Let $h_{m,v}$ denote the probability that the node v is activated at time $t_{m,v}$. We need to calculate $h_{m,v}$ in order to derive $\mathcal{L}(\boldsymbol{r}, \kappa; \mathcal{D}_M)$. Here, to calculate $h_{m,v}$, we introduce an indicator vector

$$\boldsymbol{a}_{m,*,v} = (a_{m,u,v})_{u \in B(v) \cap C_m(t_{m,v}-1)},$$

where $a_{m,u,v} = 1$ if node u actually succeeded in activating v at time $t_{m,v}$; $a_{m,u,v} = 0$ otherwise. Note that if there exist multiple active parents for the node v, i.e.,

$$\eta = |B(v) \cap C_m(t_{m,v} - 1)| > 1,$$

it is not possible to know the exact values of $\boldsymbol{a}_{m,*,v}$ from the observed data. For example, in case of $B(v) \cap C_m(t_{m,v} - 1) = \{u, u'\}$, i.e., $\eta = 2$, we need to consider the following three possibilities;

1) $a_{m,u,v} = 1,\ a_{m,u',v} = 0,$

2) $a_{m,u,v} = 0,\ a_{m,u',v} = 1,$

3) $a_{m,u,v} = 1,\ a_{m,u',v} = 1.$

Thus we can regard the indicator vector $\boldsymbol{a}_{m,*,v}$ as a latent vector, the number of hidden states of which amounts to as large as $2^\eta - 1$. This means that a naive calculation algorithm is likely to suffer from computational loads when η becomes larger. To cope with this difficulty, we develop efficient computation formulas (see Eqs (5) and (16) below).

Let $\mathcal{A}_{m,u,v}$ denote the probability that a node $u \in B(v) \cap C_m(t_{m,v} - 1)$ activates the node v at time $t_{m,v}$, that is,

$$\mathcal{A}_{m,u,v} = \kappa_{u,v} r_{u,v} (1 - r_{u,v})^{t_{m,v} - t_{m,u} - 1}. \tag{1}$$

Let $\mathcal{B}_{m,u,v}$ denote the probability that the node v is not activated from a node $u \in B(v) \cap C_m(t_{m,v} - 1)$ within the time-period $[t_{m,u} + 1, t_{m,v}]$, that is,

$$\begin{aligned} \mathcal{B}_{m,u,v} &= 1 - \kappa_{u,v} \sum_{t=t_{m,u}+1}^{t_{m,v}} r_{u,v} (1 - r_{u,v})^{t - t_{m,u} - 1} \\ &= \kappa_{u,v} (1 - r_{u,v})^{t_{m,v} - t_{m,u}} + (1 - \kappa_{u,v}). \end{aligned} \tag{2}$$

By using the indicator vector $\boldsymbol{a}_{m,*,v}$, the probability $h_{m,v}$ can naturally be expressed as

$$h_{m,v} = \sum_{\boldsymbol{a}_{m,*,v} \neq \boldsymbol{0}} f_{m,v}(\boldsymbol{a}_{m,*,v}), \tag{3}$$

where the summation is taken over all non-zero indicator (binary) vectors, and

$$f_{m,v}(\boldsymbol{a}_{m,*,v}) = \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} (\mathcal{A}_{m,u,v})^{a_{m,u,v}} (\mathcal{B}_{m,u,v})^{1 - a_{m,u,v}}. \tag{4}$$

Note that $f_{m,v}(\boldsymbol{a}_{m,*,v})$ is the probability that node v is activated at time $t_{m,v}$ according to the indicator vector $\boldsymbol{a}_{m,*,v}$. In order to efficiently calculate $h_{m,v}$, we consider the following transformation:

$$h_{m,v} = \sum_{\boldsymbol{a}_{m,*,v}} f_{m,v}(\boldsymbol{a}_{m,*,v}) - \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} \mathcal{B}_{m,u,v},$$

where the summation is taken over all indicator (binary) vectors. Thus, by Equation (4), we have

$$h_{m,v} = \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} (\mathcal{A}_{m,u,v} + \mathcal{B}_{m,u,v}) - \prod_{u \in B(v) \cap C_m(t_{m,v}-1)} \mathcal{B}_{m,u,v}. \tag{5}$$

Therefore, by using Eq. (5), we can calculate $h_{m,v}$ efficiently without considering all the possible occurrences of non-zero indicator vectors.

Next, for the mth information diffusion result, we consider any link $(v, w) \in E$ such that $v \in C_m(T_m)$ and $w \notin C_m(T_m)$. Let $g_{m,v,w}$ denote the probability that the node w is not activated by the node v within the observed time-period $[0, T_m]$. We can easily derive the following equation:

$$g_{m,v,w} = \kappa_{v,w}(1 - r_{v,w})^{T_m - t_{m,v}} + (1 - \kappa_{v,w}). \tag{6}$$

Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e.,

$$T_m \gg \max\{t; D_m(t) \neq \emptyset\}.$$

Thus, as $T_m \to \infty$ in Eq. (6), we assume

$$g_{m,v,w} = 1 - \kappa_{v,w}. \tag{7}$$

Therefore, by using Eqs (5) and (7), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{t=1}^{T_m} \prod_{v \in D_m(t)} h_{m,v} \right) \left(\prod_{v \in C_m(T_m)} \prod_{w \in F(v) \setminus C_m(T_m)} g_{m,v,w} \right), \tag{8}$$

where $F(v) \setminus C_m(T_m)$ denotes the set difference of $F(v)$ and $C_m(T_m)$, that is, $F(v) \setminus C_m(T_m) = \{w \in F(v); w \notin C_m(T_m)\}$. In this paper, we focus on the above situation (i.e., Eq. (7)) for simplicity, but we can easily modify our method to cope with the general one (i.e., Eq. (6)). Thus, our problem is to obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, which maximize Eq. (8). For this estimation problem, we derive a method based on an iterative algorithm in order to stably obtain its solution.

3.2. Estimation method

We describe our method for estimating the optimal values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We suppose that

$$\overline{\boldsymbol{r}} = (\overline{r}_{u,v})_{(u,v) \in E}, \quad \overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})_{(u,v) \in E}$$

are the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. We derive update formulas of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ such that

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) \geqslant \mathcal{L}(\overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}; \mathcal{D}_M).$$

For each $v \in C_m(T_m)$ and indicator vector $\boldsymbol{a}_{m,*,v}$, we define $q_{m,v}(\boldsymbol{a}_{m,*,v})$ by

$$q_{m,v}(\boldsymbol{a}_{m,*,v}) = \frac{f_{m,v}(\boldsymbol{a}_{m,*,v})}{h_{m,v}}. \tag{9}$$

Note that $q_{m,v}(\boldsymbol{a}_{m,*,v})$ is the posterior probability that the indicator vector is $\boldsymbol{a}_{m,*,v}$ when v is activated at time $t_{m,v}$ (see Eqs (3), (4)). Let $\overline{\mathcal{A}}_{m,u,v}$, $\overline{\mathcal{B}}_{m,u,v}$, $\overline{h}_{m,v}$, and $\overline{q}_{m,v}(\boldsymbol{a}_{m,*,v})$ denote the values of $\mathcal{A}_{m,u,v}$, $\mathcal{B}_{m,u,v}$, $h_{m,v}$, and $q_{m,v}(\boldsymbol{a}_{m,*,v})$ calculated by using $\overline{\boldsymbol{r}}$ and $\overline{\kappa}$, respectively.

From Eqs (3), (7), and (8), we can transform our objective function $\mathcal{L}(\boldsymbol{r}, \kappa; \mathcal{D}_M)$ as follows:

$$\log \mathcal{L}(\boldsymbol{r}, \kappa; \mathcal{D}_M) = Q(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa}) - H(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa}). \tag{10}$$

Here, $Q(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa})$ is defined by

$$Q(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa}) = \sum_{m=1}^{M} \left(\sum_{t=1}^{T_m} \sum_{v \in D_m(t)} Q_{m,v} + \sum_{v \in C_m(T_m)} \sum_{w \in F(v) \setminus C_m(T_m)} \log(1 - \kappa_{v,w}) \right), \tag{11}$$

where

$$Q_{m,v} = \sum_{\boldsymbol{a}_{m,*,v} \neq \boldsymbol{0}} \overline{q}_{m,v}(\boldsymbol{a}_{m,*,v}) \log f_{m,v}(\boldsymbol{a}_{m,*,v}). \tag{12}$$

Also, $H(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa})$ is defined by

$$H(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa}) = \sum_{m=1}^{M} \sum_{t=1}^{T_m} \sum_{v \in D_m(t)} \sum_{\boldsymbol{a}_{m,*,v} \neq \boldsymbol{0}} J_{m,v}(\boldsymbol{a}_{m,*,v}), \tag{13}$$

where

$$J_{m,v}(\boldsymbol{a}_{m,*,v}) = \overline{q}_{m,v}(\boldsymbol{a}_{m,*,v}) \log q_{m,v}(\boldsymbol{a}_{m,*,v}). \tag{14}$$

Since the function $H(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa})$ of $\boldsymbol{r}$ and κ is maximized at $\boldsymbol{r} = \overline{\boldsymbol{r}}$ and $\kappa = \overline{\kappa}$ from Equations (13) and (14), we can increase the value of $\mathcal{L}(\boldsymbol{r}, \kappa; \mathcal{D}_M)$ by maximizing the function $Q(\boldsymbol{r}, \kappa; \overline{\boldsymbol{r}}, \overline{\kappa})$ of $\boldsymbol{r}$ and κ (see Eq. (10)).

In order to efficiently calculate $Q_{m,v}$, we derive the following formula from Eqs (4), (9) and (12):

$$Q_{m,v} = \sum_{u \in B(v) \cap C_m(t_{m,v}-1)} \left(\overline{\alpha}_{m,u,v} \log \mathcal{A}_{m,u,v} + (1 - \overline{\alpha}_{m,u,v}) \log \mathcal{B}_{m,u,v} \right), \tag{15}$$

where

$$\overline{\alpha}_{m,u,v} = \sum_{\boldsymbol{a}_{m,*,v} \neq \boldsymbol{0}} a_{m,u,v}\, \overline{q}_{m,v}(\boldsymbol{a}_{m,*,v}).$$

Note that $0 < \overline{\alpha}_{m,u,v} < 1$. Then, we can easily show that

$$\overline{\alpha}_{m,u,v} = \frac{\overline{\mathcal{A}}_{m,u,v}}{\overline{h}_{m,v}} \prod_{x \in B(v) \cap C_m(t_{m,v}-1) \setminus \{u\}} \left(\overline{\mathcal{A}}_{m,x,v} + \overline{\mathcal{B}}_{m,x,v} \right). \tag{16}$$

Therefore, by using Eq. (16), we can also calculate $\overline{\alpha}_{m,u,v}$ without computation of exponential order.

Note here that although $\log \mathcal{A}_{m,u,v}$ is a linear combination of $\log r_{u,v}$, $\log(1-r_{u,v})$, and $\log \kappa_{u,v}$, $\log \mathcal{B}_{m,u,v}$ cannot be written by such a linear combination (see Eqs (1) and (2)). In order to cope with this problem of $\log \mathcal{B}_{m,u,v}$, we transform $\log \mathcal{B}_{m,u,v}$ in the same way as Eq. (10):

$$\log \mathcal{B}_{m,u,v} = Q^{\mathcal{B}}_{m,u,v} - H^{\mathcal{B}}_{m,u,v}, \tag{17}$$

where

$$Q^{\mathcal{B}}_{m,u,v} = \overline{\beta}_{m,u,v} \log \left(\kappa_{u,v}(1-r_{u,v})^{t_{m,v}-t_{m,u}}\right) + (1-\overline{\beta}_{m,u,v}) \log(1-\kappa_{u,v}), \tag{18}$$

and

$$H^{\mathcal{B}}_{m,u,v} = \overline{\beta}_{m,u,v} \log \beta_{m,u,v} + (1-\overline{\beta}_{m,u,v}) \log(1-\beta_{m,u,v}). \tag{19}$$

Here, $\beta_{m,u,v}$ is defined by

$$\beta_{m,u,v} = \frac{\kappa_{u,v}(1-r_{u,v})^{t_{m,v}-t_{m,u}}}{\kappa_{u,v}(1-r_{u,v})^{t_{m,v}-t_{m,u}} + (1-\kappa_{u,v})}, \tag{20}$$

and $\overline{\beta}_{m,u,v}$ is the value of $\beta_{m,u,v}$ calculated by using $\overline{\boldsymbol{r}}$ and $\overline{\boldsymbol{\kappa}}$. Note that $0 < \overline{\beta}_{m,u,v} < 1$. We define $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ by

$$Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) = \sum_{m=1}^{M} \left(\sum_{t=1}^{T_m} \sum_{v \in D_m(t)} Q'_{m,v} + \sum_{v \in C_m(T_m)} \sum_{w \in F(v) \setminus C_m(T_m)} \log(1-\kappa_{v,w}) \right), \tag{21}$$

where

$$Q'_{m,v} = \sum_{u \in B(v) \cap C_m(t_{m,v}-1)} \left(\overline{\alpha}_{m,u,v} \log \mathcal{A}_{m,u,v} + (1-\overline{\alpha}_{m,u,v}) Q^{\mathcal{B}}_{m,u,v}\right). \tag{22}$$

Note that by Eqs (19) and (20), the function $H^{\mathcal{B}}_{m,u,v}$ of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ is maximized at $\boldsymbol{r} = \overline{\boldsymbol{r}}$ and $\boldsymbol{\kappa} = \overline{\boldsymbol{\kappa}}$. Thus, we can maximize $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ by maximizing $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ as functions of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, (see Eqs (11), (15), (17), (21) and (22)). Note here that the function $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ is a linear combination of $\{\log r_{u,v}, \log(1-r_{u,v}), \log \kappa_{u,v}, \log(1-\kappa_{u,v}); (u,v) \in E\}$ with positive coefficients.

From Eqs (1), (18), (21) and (22), we can easily obtain the solution which maximizes $Q'(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$. Hence, we have the following theorem.

Theorem 1. *Let $\overline{\boldsymbol{r}} = (\overline{r}_{u,v})$ and $\overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})$ be the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. For each $(u,v) \in E$ and $m \in \{1, \cdots, M\}$, we define $\mathcal{M}^{+}_{u,v}$, $\mathcal{M}^{-}_{u,v}$, and $\overline{\varphi}_{m,u,v}$ by*

$$\mathcal{M}^{+}_{u,v} = \{m \in \{1, \cdots, M\};\ u, v \in C_m(T_m),\ v \in F(u),\ t_{m,u} < t_{m,v}\}, \tag{23}$$

$$\mathcal{M}^{-}_{u,v} = \{m \in \{1, \cdots, M\};\ u \in C_m(T_m),\ v \notin C_m(T_m),\ v \in F(u)\}, \tag{24}$$

and

$$\overline{\varphi}_{m,u,v} = \overline{\alpha}_{m,u,v} + (1-\overline{\alpha}_{m,u,v})\,\overline{\beta}_{m,u,v},$$

respectively. Then, if we update the values of $\boldsymbol{r}$ *and* $\boldsymbol{\kappa}$ *by*

$$r_{u,v} = \frac{\sum_{m \in \mathcal{M}^+_{u,v}} \overline{\alpha}_{m,u,v}}{\sum_{m \in \mathcal{M}^+_{u,v}} (t_{m,v} - t_{m,u})\, \overline{\varphi}_{m,u,v}}, \tag{25}$$

$$\kappa_{u,v} = \frac{\sum_{m \in \mathcal{M}^+_{u,v}} \overline{\varphi}_{m,u,v}}{|\mathcal{M}^+_{u,v}| + |\mathcal{M}^-_{u,v}|} \tag{26}$$

for all $(u, v) \in E$, *we have*

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) \geqslant \mathcal{L}(\overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}; \mathcal{D}_M).$$

Theorem 1 provides the update formulas (25) and (26) of the parameters $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. Since the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ is ensured not to decrease by this updating, this updating mechanism asymptotically converges to at least locally optimal solutions, and we can propose a method similar to EM algorithm in a natural way. Note that each $h_{m,u,v}$ and $\overline{\alpha}_{m,u,v}$ are efficiently calculated by Equations (5) and (16), respectively.

4. Evaluation methods

We first evaluate the estimation accuracy of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the proposed learning method. Next, we exploit the estimated model to predict the influence of nodes and extract the influential nodes, and evaluate their performance.

For an initial active node v, let $\psi(v; \boldsymbol{r}, \boldsymbol{\kappa})$ denote the number of active nodes at the end of the information diffusion process for the ICTD model with parameter values $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. Note that $\psi(v; \boldsymbol{r}, \boldsymbol{\kappa})$ is a random variable since the information diffusion process is a random process. Let $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ denote the expected value of $\psi(v; \boldsymbol{r}, \boldsymbol{\kappa})$. We call $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ the *influence degree* of node v for the ICTD model with parameter values $\boldsymbol{r}$ and $\boldsymbol{\kappa}$.

When we are given the set of information diffusion results $\mathcal{D}_M$, we measure the influence of node v by the influence degree $\sigma(v; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ for the ICTD model which generated $\mathcal{D}_M$, where $\boldsymbol{r}^*$ and $\boldsymbol{\kappa}^*$ are the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. Thus, a node v with high influence degree $\sigma(v; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ is an influential node. We first estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$. Suppose that $\widehat{\boldsymbol{r}}$ and $\widehat{\boldsymbol{\kappa}}$ are the estimated values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. Then, we predict $\sigma(v; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})$ and use it as the estimated value of the influence degree $\sigma(v; \boldsymbol{r}^*, \boldsymbol{\kappa}^*)$ of node v. Moreover, we extract the influential nodes by ranking nodes v based on $\sigma(v; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})$.

We evaluate the proposed method in terms of the capability of both predicting the influence degrees of nodes and ranking the influential nodes. We focus on the performance for high-rank nodes since we are interested in influential nodes. Let $L^*(k)$ denote the set of top k influential nodes for the true ICTD model. Let $\widehat{L}(k)$ be the set of top k influential nodes estimated by a given ranking method. We evaluate the performance of the ranking method by the *ranking similarity* $\mathcal{F}(k)$ within the rank k, where $\mathcal{F}(k)$ is defined by

$$\mathcal{F}(k) = \frac{|L^*(k) \cap \widehat{L}(k)|}{k}. \tag{27}$$

5. Comparison methods

5.1. eGGLT method

Gruhl et al. [7] presented a method for estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ for the ICTD model, from which the underlying link structure E was induced. Thus, their method did not explicitly use the link structure E, and $\boldsymbol{r} = (r_{u,v})$ and $\boldsymbol{\kappa} = (\kappa_{u,v})$ were first regarded as full $|V|(|V|-1)$ dimensional vectors. In practice, by exploiting some heuristics based on the observed data $\mathcal{D}_M$, they restricted the non-zero entries of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ in order to achieve a reduction in computational cost, and inferred the underlying link structure from the estimated values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ under these constraints. However, their method can be straightforwardly extended to the case where the underlying network structure is known. We refer to the extended method as the *eGGLT method*,[2] and compare the proposed method with the eGGLT method.

We begin with describing the original method by Gruhl et al. [7]. Basically, they presented an EM-like algorithm. Let $\overline{\boldsymbol{r}} = (\overline{r}_{u,v})$ and $\overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})$ be the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. The update formulas for $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ are as follows:

$$r_{u,v} = \frac{\sum_{m \in \mathcal{S}^+_{u,v}} \overline{p}_{m,u,v}}{\sum_{m \in \mathcal{S}^+_{u,v}} (t_{m,v} - t_{m,u})\, \overline{p}_{m,u,v}}, \tag{28}$$

$$\kappa_{u,v} = \frac{\sum_{m \in \mathcal{S}^+_{u,v}} \overline{p}_{m,u,v}}{\sum_{m \in \mathcal{S}^+_{u,v}} \overline{\lambda}_{m,u,v} + \sum_{m \in \mathcal{S}^-_{u,v}} \overline{\mu}_{m,u,v}}, \tag{29}$$

for all $u, v \in V$ with $u \neq v$, where

$$\overline{p}_{m,u,v} = \frac{\overline{r}_{u,v}(1-\overline{r}_{u,v})^{t_{m,v}-t_{m,u}-1}\overline{\kappa}_{u,v}}{\sum_{w \in C_m(t_{m,v}-1)} \overline{r}_{w,v}(1-\overline{r}_{w,v})^{t_{m,v}-t_{m,w}-1}\overline{\kappa}_{w,v}}, \tag{30}$$

and

$$\overline{\lambda}_{m,u,v} = 1 - (1-\overline{r}_{u,v})^{t_{m,v}-t_{m,u}},$$

$$\overline{\mu}_{m,u,v} = 1 - (1-\overline{r}_{u,v})^{T_m - t_{m,u}}.$$

Here, $\mathcal{S}^+_{u,v}$ and $\mathcal{S}^-_{u,v}$ are defined by

$$\mathcal{S}^+_{u,v} = \{m \in \{1, \cdots, M\};\ u, v \in C_m(T_m),\ t_{m,u} < t_{m,v}\}, \tag{31}$$

$$\mathcal{S}^-_{u,v} = \{m \in \{1, \cdots, M\};\ u \in C_m(T_m),\ v \notin C_m(T_m)\}. \tag{32}$$

We note that they did not derive the EM-like algorithm in a principled way. No objective function is defined to obtain the updating formulas. Here we should also mention that this method might have an intrinsic limitation because simultaneous activations from multiple parent nodes are not considered in Eq. (30).

Now, we define the eGGLT method. The eGGLT method straightforwardly incorporates the link structure E into the original method by Gruhl et al. [7]. Namely, the eGGLT method only changes the

[2] It means the extended "Gruhl, Guha, Liben-Nowell, and Tomkins [7]" method.

update formulas (28) and (29) of $r_{u,v}$ and $\kappa_{u,v}$ for any link $(u,v) \in E$ as follows: $\mathcal{S}^+_{u,v}$ and $\mathcal{S}^-_{u,v}$ are replaced with $\mathcal{M}^+_{u,v}$ and $\mathcal{M}^-_{u,v}$, respectively (see Equations (31), (32), (23) and (24)). Note here that if the underlying network $G = (V, E)$ is a complete graph, we have $\mathcal{S}^+_{u,v} = \mathcal{M}^+_{u,v}$ and $\mathcal{S}^-_{u,v} = \mathcal{M}^-_{u,v}$ for all $(u,v) \in E$. The eGGLT method can be applied to the problem of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$ when the underlying network structure is known. Therefore, the eGGLT method can be also applied to the problems of predicting the influence degrees of nodes for the true ICTD model and ranking the influential nodes.

5.2. *Conventional methods in social network analysis*

As for the problem of extracting the high ranked influential nodes for the true ICTD model, we also compare the proposed method with four heuristics from social network analysis, which are the degree, the betweenness, the closeness, and the PageRank methods.

First, "degree centrality", "betweenness centrality", and "closeness centrality" are commonly used as influence measure for a bidirectional network in sociology [20], where the degree of node v is defined as the number of links attached to v, the betweenness of node v is defined as the total number of shortest paths between pairs of nodes that pass through v, and the closeness of node v is defined as the reciprocal of the average distance between v and other nodes in the network.

We also consider measuring the influence of each node by its "authoritativeness" obtained by the "PageRank" method [3], since this is a well known method for identifying authoritative or influential pages in a hyperlink network of web pages. This method has a parameter ε; when we view it as a model of a random web surfer, ε corresponds to the probability with which a surfer jumps to a page picked uniformly at random [17]. In our experiments, we used a typical setting of $\varepsilon = 0.15$.

6. Experimental evaluation

We first compared the proposed method with the eGGLT method for the capability of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ in the case of a complete graph. Next, using three large real social networks, we evaluated the effectiveness of the proposed method for 1) estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, 2) predicting the influence degrees of the high-ranked nodes under the true ICTD model, and 3) ranking the influential nodes.

6.1. *Experimental settings*

According to [7], we generated the training data $\mathcal{D}_M$ by simulating the true ICTD model M_0 times from every single node as being an initial active node, where $M = M_0|V|$. When we estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$, we always used the same initial guess and the same iteration number for the proposed and the eGGLT methods. Strictly speaking, we always set the initial values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ as $r_{u,v} = 1/2$ and $\kappa_{u,v} = 1/2$ for any $(u,v) \in E$, and performed 100 iterations. We confirmed that the relative difference of the parameter values in the successive iteration is in the order of 10^{-5}, and thus, the solutions are judged to be converged.

We note that the influence degree $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ of a node v is invariant with respect to the values of the delay-parameters $\boldsymbol{r}$. Thus, we can calculate the $\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa})$ of the ICTD model by the influence degree of v for the corresponding IC model. Hence, we evaluated the influence degrees $\{\sigma(v; \boldsymbol{r}, \boldsymbol{\kappa}); v \in V\}$ by applying the method of [9] with the parameter value 10,000 to the corresponding IC model, where the parameter represents the number of bond percolation processes (see [9] for more details).

Table 1
Learning results for the complete graph dataset. Correct values: $r_{u,v} = 0.667$, $\kappa_{u,v} = 0.1$, for $\forall(u,v) \in E$

Proposed method				
M_0	mean(r)	std(r)	mean(κ)	std(κ)
20	0.641	0.254	0.103	0.058
40	0.688	0.180	0.101	0.041
60	0.677	0.166	0.101	0.037
eGGLT method				
M_0	mean(r)	std(r)	mean(κ)	std(κ)
20	0.994	0.028	0.054	0.026
40	0.993	0.027	0.054	0.018
60	0.993	0.025	0.054	0.016

6.2. Synthetic network

We recall that the proposed method considers the possibility that a node v can be activated simultaneously by multiple parent nodes $\{u\}$ that has become activated at different times, whereas the eGGLT method does not assume this possibility. Thus, we first consider a network with the highest clustering coefficients (see [15]), where there is a large possibility that the above situations happen. Here, we exploited the complete directed graph of 50 nodes as the network $G = (V, E)$. Note that the eGGLT method coincides with the original method by Gruhl et al. [7] for a complete graph. According to [7], we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/10$ for any $(u,v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We refer to this dataset as the complete graph dataset.

We compared the capability of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ for the proposed and the eGGLT methods. Table 1 shows the results for different number of simulations M_0 for all nodes. Here, mean($\boldsymbol{r}$) and mean($\boldsymbol{\kappa}$) denote the means of the estimated values of $\boldsymbol{r} = (r_{u,v})$ and $\boldsymbol{\kappa} = (\kappa_{u,v})$, respectively, and std($\boldsymbol{r}$) and std($\boldsymbol{\kappa}$) denote their standard deviations, respectively. The results demonstrate that the proposed method outperforms the eGGLT method. Our algorithm can converge to the true values efficiently when there is a reasonable amount of training data. The results demonstrate the effectiveness of the proposed method.

6.3. Real networks

6.3.1. Network dataset

As a large real social network $G = (V, E)$, we first employed the blog network used in [10]. This was a bidirectional network with 12,047 nodes and 79,920 directed links. Again, we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/10$ for any $(u,v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We refer to this dataset as the blog network dataset.

Second, we employed a network derived from the Enron Email Dataset [12]. We first extracted the email addresses that appeared in the Enron Email Dataset as senders and recipients. We regarded each email address as a node, and constructed an undirected network obtained by linking two email addresses u and v if u sent an email to v and received an email from v. Next, we extracted its maximal strongly connected component, and constructed a directed network by regarding those undirected links as bidirectional ones. We refer to this strongly connected bidirectional network as the Enron network. This network had 4,254 nodes and 44,314 directed links. Again, we used $r_{u,v} = 2/3$ and $\kappa_{u,v} = 1/10$ for any $(u,v) \in E$ as the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$. We refer to this dataset as the Enron network dataset.

Table 2
Learning results for the blog network dataset. Correct values: $\tau_{u,v} = 0.667$, $\kappa_{u,v} = 0.1$, for $\forall(u,v) \in E$

Proposed method				
M_0	mean(τ)	std(τ)	mean(κ)	std(κ)
20	0.686	0.120	0.100	0.027
40	0.679	0.092	0.100	0.019
60	0.674	0.075	0.100	0.016
eGGLT method				
M_0	mean(τ)	std(τ)	mean(κ)	std(κ)
20	0.756	0.135	0.096	0.029
40	0.750	0.119	0.096	0.022
60	0.746	0.112	0.096	0.019

Table 3
Learning results for the Enron network dataset. Correct values: $\tau_{u,v} = 0.667$, $\kappa_{u,v} = 0.1$, for $\forall(u,v) \in E$

Proposed method				
M_0	mean(τ)	std(τ)	mean(κ)	std(κ)
20	0.657	0.101	0.102	0.020
40	0.657	0.083	0.102	0.016
60	0.657	0.073	0.102	0.013
eGGLT method				
M_0	mean(τ)	std(τ)	mean(κ)	std(κ)
20	0.856	0.137	0.082	0.030
40	0.855	0.134	0.082	0.028
60	0.854	0.133	0.082	0.028

Third, we employed the co-authorship network used in [18]. This was a bidirectional network with 12,357 nodes and 38,896 directed links. For the co-authorship network, we used $\tau_{u,v} = 2/3$ and $\kappa_{u,v} = 1/5$ for any $(u,v) \in E$ as the true values of τ and κ, since the mean out-degree of the co-authorship network is much smaller than the blog and the Enron networks. In fact, when we used $\kappa_{u,v} = 1/10$ for any link $(u,v) \in E$, the influence degrees of nodes became very low (they were less than 15). We refer to this dataset as the co-authorship network dataset.

6.3.2. *Experimental results*

First, we evaluated the performance for estimating the values of τ and κ. Tables 2, 3 and 4 show the results for different number of simulations M_0 for all nodes in the blog, the Enron and the co-authorship network datasets, respectively. Here, the meanings of mean(τ), std(τ), mean(κ), and std(κ) are the same as in Table 1. We observe that the proposed method outperforms the eGGLT method for all of these three datasets in estimating the values of τ and κ, and can converge to the true values efficiently when there is a reasonable amount of training data.

Next, we investigated the performance for predicting the influence degrees of the top 200 influential nodes for the true ICTD model. As described in Section 4, we predicted $\sigma(v_k^*; \widehat{\tau}, \widehat{\kappa})$ and used it as the influence degree $\sigma(v_k^*; \tau^*, \kappa^*)$ of node v_k^* ($k = 1, \cdots, 200$), where v_k^* denotes the node of rank k for the true ICTD model. Figures 1, 2 and 3 show the results in the case of $M_0 = 60$ for the blog, the Enron and the co-authorship network datasets, respectively. In each figure, the thick dashed line displays the true influence degrees, and circles and squares indicate the influence degrees predicted by the proposed

Table 4
Learning results for the co-authorship network dataset. Correct values: $\tau_{u,v} = 0.667$, $\kappa_{u,v} = 0.2$, for $\forall(u,v) \in E$

Proposed method				
M_0	mean(τ)	std(τ)	mean(κ)	std(κ)
20	0.682	0.100	0.200	0.044
40	0.675	0.070	0.200	0.031
60	0.672	0.057	0.200	0.025
eGGLT method				
M_0	mean(τ)	std(τ)	mean(κ)	std(κ)
20	0.690	0.100	0.209	0.048
40	0.682	0.070	0.209	0.034
60	0.680	0.057	0.209	0.028

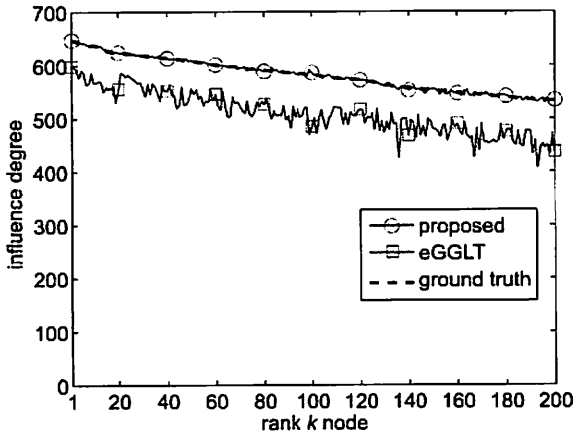

Fig. 1. Performance comparison for predicting the influence degrees of nodes in the case of $M_0 = 60$ for the blog network dataset. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-2011-0486)

and the eGGLT methods, respectively. We observe from these figures that unlike the eGGLT method, the proposed method makes good predictions of the true influence degrees of the high-ranked nodes for all of these three datasets. Here, we evaluated the *mean influence-prediction error* $\mathcal{E}(k)$ within the top rank k, which is defined by

$$\mathcal{E}(k) = \frac{1}{k}\sum_{i=1}^{k} |\sigma(v_i^*; \boldsymbol{r}^*, \boldsymbol{\kappa}^*) - \sigma(v_i^*; \widehat{\boldsymbol{r}}, \widehat{\boldsymbol{\kappa}})|,$$

and also evaluated the mean $\mathcal{H}(k)$ of the true influence degrees $\{\sigma(v_i^*; \boldsymbol{r}^*, \boldsymbol{\kappa}^*);\ i = 1, \cdots, k\}$ for the top k nodes. Table 5 compares the errors by the two methods for the three datasets in case of $k = 20$ and $k = 200$. We see that the mean influence-prediction error $\mathcal{E}(k)$ of the proposed method is much smaller than 1% of the mean of the true influence degrees $\mathcal{H}(k)$ regardless of the value of k for every dataset, whereas, in most cases, $\mathcal{E}(k)$ of the eGGLT method is more than 50 times greater than $\mathcal{E}(k)$ of the proposed method and more than 10% of $\mathcal{H}(k)$. Even in the best case where the percentage error of the eGGLT method is minimum, i.e., 6.6% of $\mathcal{H}(k)$ for $k = 20$ for the Enron network dataset, $\mathcal{E}(k)$ of the eGGLT method is about 500 times greater than $\mathcal{E}(k)$ of the proposed method. Therefore, we see

Table 5
Mean influence-prediction errors $\mathcal{E}(k)$ of the proposed and eGGLT methods and the means of the true influence degrees $\mathcal{H}(k)$ for the three network datasets ($M_0 = 60$)

dataset	k	$\mathcal{E}(k)$ of proposed method	$\mathcal{E}(k)$ of eGGLT method	$\mathcal{H}(k)$
blog network	20	1.2	65.0	635.5
	200	1.8	71.0	581.9
Enron network	20	0.2	99.4	1500.7
	200	3.2	183.1	1485.1
co-authorship network	20	1.5	129.4	573.7
	200	2.1	105.1	415.1

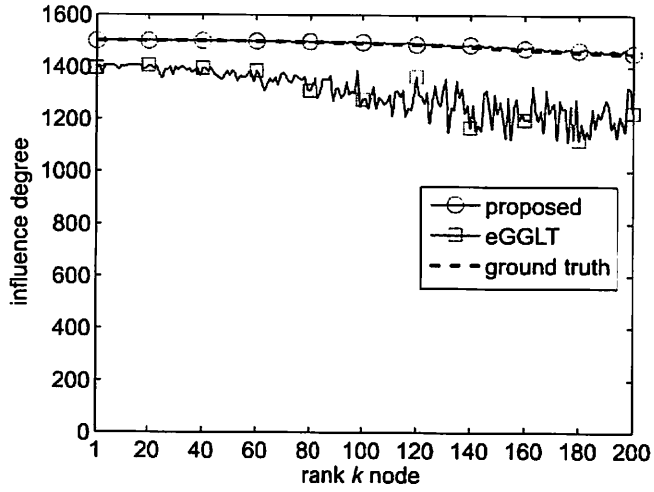

Fig. 2. Performance comparison for predicting the influence degrees of nodes in the case of $M_0 = 60$ for the Enron network dataset. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-2011-0486)

that for all of these three datasets the proposed method works well and gives far better results than the eGGLT method.

Next, we evaluated the performance for ranking the top 200 influential nodes under the true ICTD model. Figures 4, 5 and 6 show the ranking similarity $\mathcal{F}(k)$ (see Eq. (27)) as a function of the top rank k in the case of $M_0 = 60$ for the blog, the Enron and the co-authorship network datasets, respectively. Here, circles, squares, triangles, diamonds, crosses, and asterisks indicate the results for the proposed, the eGGLT, the degree, the betweenness, the closeness, and the PageRank methods, respectively. For the blog network dataset, the proposed method performed best, and the eGGLT method followed. The other methods (the conventional methods in social network analysis) were much worse than these two methods. For the Enron network dataset, the proposed method performed best, and the eGGLT and the out-degree methods followed. The performance difference between the eGGLT and the out-degree methods was small. For example, the eGGLT method was worse than the out-degree method for the ranking similarity within rank $k = 200$. For the co-authorship network dataset, the proposed method performed best, and the eGGLT method followed. The other methods (the conventional methods in social network analysis) were much worse than these two methods. Therefore, we see for these datasets that the proposed method works effectively.

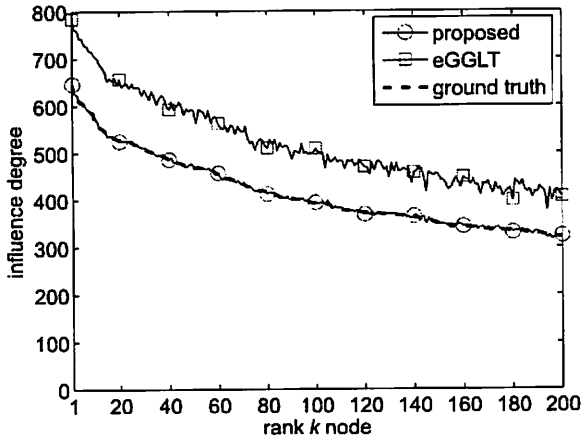

Fig. 3. Performance comparison for predicting the influence degrees of nodes in the case of $M_0 = 60$ for the co-authorship network dataset. (Colours are visible in the online version of the article; http://dx.doi.org/10.3233/IDA-2011-0486)

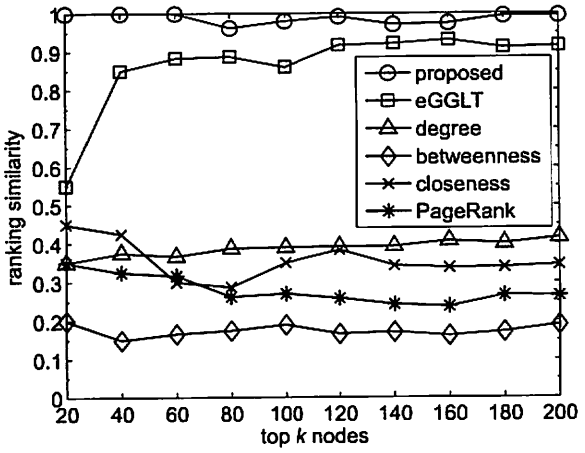

Fig. 4. Performance comparison for ranking the influential nodes in the case of $M_0 = 60$ for the blog network dataset.

7. Discussion

We note that the results of the eGGLT method were not good in Table 1, contrary to the results in [7]. We attribute this to the inadequacy of the parameter estimation methods and the different setting of network parameters. First, as stated earlier, note that the proposed method considers the possibility that a node v can be activated simultaneously by multiple parents $\{u\}$ that has become activated at different times (although v is activated only once), whereas the eGGLT method does not assume this possibility. Second note that the networks used in [7] are modified Erdös-Renyi random graphs with $|V| = 1000$ and d (the degree of node) $= 3$. This gives a sparse graph with almost 0 clustering coefficients (see [15]). Further, the diffusion parameter $\kappa_{u,v}$ they used for any link (u, v) is $1/10$, which implies that the above multiple activation possibility is essentially 0. However, the network we used in our experiment in Section 6.2 is a complete graph with very high clustering coefficients (i.e., clustering coefficient 1), and there is a large possibility that the above situations happen. In this setting, ignoring the possibility of a node being activated simultaneously from more than one parent node would most

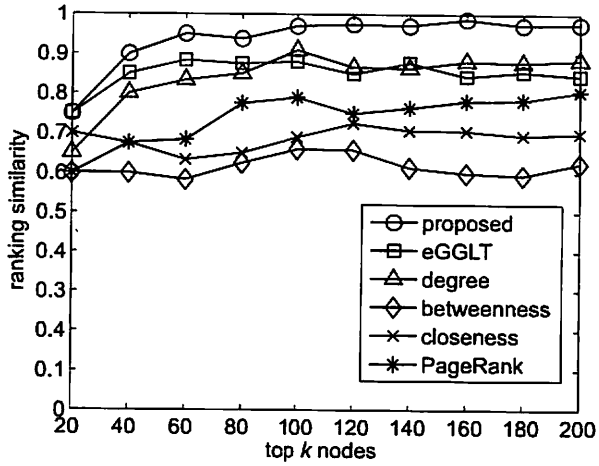

Fig. 5. Performance comparison for ranking the influential nodes in the case of $M_0 = 60$ for the Enron network dataset.

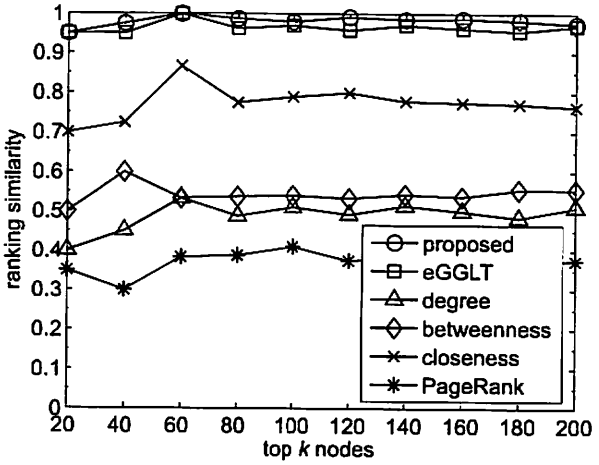

Fig. 6. Performance comparison for ranking the influential nodes in the case of $M_0 = 60$ for the co-authorship network dataset.

probably give inaccurate estimates of the parameters. The results in Section 6.2 are consistent with this observation. For the large real social networks used in our experiments, the mean clustering coefficients of the blog, the Enron and the co-authorship networks were 0.262, 0.370 and 0.218, respectively. This means that the Enron network has a larger possibility that the above situations happen than the blog and the co-authorship networks. Tables 2, 3 and 4 show that the parameter estimation results of the eGGLT method for the Enron network were worst. This is consistent with the discussion above.

Compared with the eGGLT method, our method derives the learning algorithm in a principled way. It has the objective function which has a clear meaning of the likelihood of obtaining the observed data, and the parameter updating algorithm is derived such that it iteratively increases the likelihood with the convergence guaranteed. Therefore, for large real social networks, the proposed method far outperformed the eGGLT method for predicting influence degrees of true high-ranked nodes (see Figs 1, 2 and 3), and always gave better results than the eGGLT method for ranking the influential nodes (see Figs 4, 5 and 6). These results also imply that estimating the parameters as accurately as possible is

very important. Further note that, in deriving the proposed algorithm, tactics are employed to avoid computational explosion.

We consider that our ranking method presents a novel concept of centrality based on the information diffusion model, i.e., the ICTD model. Actually, Figs 4, 5 and 6 show that nodes identified as higher ranked by our method are substantially different from those by each of the conventional methods in social network analysis. This means that our method enables a new type of social network analysis if past information diffusion data are available. Of course, it is beyond controversy that each conventional method has its own merit and usage, and our method is an addition to them which has a different merit in terms of information diffusion.

The formulation we showed in Section 3.2 dealt with the case where each of $r_{u,v}$ and $\kappa_{u,v}$ can take a different value for each link $(u, v) \in E$. However, this would cause a serious problem of overfitting as well as unacceptably high computation cost if we are to analyze large real networks with high clustering coefficients. Parameter sharing helps improve generalization capability. Further, it is more realistic that some of the parameters share the same values across different links. In our framework, placing constraints, e.g., assigning uniform values to parameters across all links or grouping the parameters $(\kappa_{u,v})$ and $(r_{u,v})$ into several categories, etc. is straightforward. For example, we can divide the link set E into subsets $\{E_1, E_2, \cdots, E_N\}$ and assign unique parameter values r_n and κ_n for all the links in each category E_n. If the overfitting is the real problem even after grouping links, we can follow the standard approach of introducing prior distributions over the model parameters. For placing constraints in a more realistic setting, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. If there is some background knowledge about the node grouping, our method can make the best use of it, one of the characteristics of the artificial intelligence approach. Obtaining such background knowledge is also an important research topic in the knowledge discovery from social networks.

The final objective of Gruhl et al.'s work [7] was to estimate network structure. We did not focus on this aspect in this paper, however. This does not mean that our proposed method is unable to estimate the structure. Just like Gruhl et al. assumed a fully connected complete graph, we could have taken the same approach, i.e., initially assuming the complete network and deleting links for which no parameter values are obtained or the values are very small under the assumption that the observed sequence data is sufficiently large enough to cover all the possible information propagation paths. However, scalability becomes an issue with this naive approach. It is too computationally expensive to be applied for a real network, e.g. for a network with 10,000 nodes, the number of links, thus the number of each parameter, to be considered is 100,000,000. With such a huge number of parameters to search, both methods become infeasible. In addition, the amount of observation data that is required to estimate the parameter values is tremendously large and it is almost unrealistic to collect such data. Better approaches including parameter sharing mentioned above must be yet explored so as to pose strong constraints on network structure and reasonably and effectively restrict parameters to be considered. This is our future work.

Information diffusion with time-delay we picked in this paper poses an interesting machine learning problem. Each piece of training data is a sequence of observed data (thus, relational), but it has some hidden structure and it is not straightforward to map the data to node-to-node information diffusion. In theory we have to consider all the possible paths to each activated node from unknown source nodes with different time-delays. We managed this by introducing indicator variables. How to avoid computational explosion then became crucial and we introduced neat tactics.

Our method utilizes sequential data of information diffusion. In this aspect, it has some commonality with re-enforcement learning, but the main difference is that a reward for each sequence is not used in our

learning framework. Our model is meant to be useful for analyzing information diffusion via a human network, e.g., via words-of-mouth. It is not clear at the moment whether the similar approach can be used for analyzing a diffusion process in other domains, e.g., biological networks. If a similar model is confirmed to be usable, our method can also be an important technique to analyze general diffusion process. We plan to apply the proposed method to some specific tasks in a more practical setting, in which case the evaluation must be based on a task-specific performance measure for each task.

8. Conclusion

We addressed the problem of estimating the parameters for an information diffusion model (i.e., the ICTD model) in a complex social network, given the network topology and the observed time-sequence data. The model allows time-delay in information diffusion under the framework of independent cascade (IC) model, and has two kinds of parameters: the time-delay parameter and the diffusion parameter. We formulated the likelihood of obtaining the observed sequence data, and proposed an EM-like iterative method to obtain the parameter values by maximizing this likelihood. We first confirmed by using a complete graph that the proposed method outperforms a slightly modified existing method eGGLT in estimating correct parameters. Next, we showed by using three real world networks that the proposed method can much more accurately predict the influence degrees of the high-ranked nodes for the true ICTD model than the eGGLT method. Moreover, we demonstrated that it outperforms the eGGLT method and the conventional methods in social network analysis for ranking the influential nodes.

In conclusion, we blazed the path to learn a probabilistic information diffusion model over a network in a principled way. The IC model we used is the most basic, and there are other diffusion models [8]. A similar approach can be extended to these models, e.g., *linear threshold model* with time-delay, which will also be our future work.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

[1] N. Agarwal and H. Liu, Blogosphere: Research issues, tools, and application, *SIGKDD Explorations* **10** (2008), 18–31.
[2] R. Albert, H. Jeong and A.L. Barabási, Error and attack tolerance of complex networks, *Nature* **406** (2000), 378–382.
[3] S. Brin and L. Page, The anatomy of a large-scale hypertextual web search engine, *Computer Networks and ISDN Systems* **30** (1998), 107–117.
[4] P. Domingos, Mining social networks for viral marketing, *IEEE Intelligent Systems* **20** (2005), 80–82.
[5] S.N. Dorogovtsev, A.V. Goltsev and J.F.F. Mendes, Critical phenomena in complex networks, *Reviews of Modern Physics* **80** (2008), 1275–1335.
[6] J. Goldenberg, B. Libai and E. Muller, Talk of the network: A complex systems look at the underlying process of word-of-mouth, *Marketing Letters* **12** (2001), 211–223.
[7] D. Gruhl, R. Guha, D. Liben-Nowell and A. Tomkins, Information diffusion through blogspace, in: *Proceedings of the 17th International World Wide Web Conference (WWW 2004)*, 2004, pp. 107–117.
[8] D. Kempe, J. Kleinberg and E. Tardos, Maximizing the spread of influence through a social network, in: *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, 2003, pp. 137–146.

[9] M. Kimura, K. Saito and R. Nakano, Extracting influential nodes for information diffusion on a social network, in: *Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07)*, 2007, pp. 1371–1376.
[10] M. Kimura, K. Saito and H. Motoda, Blocking links to minimize contamination spread in a social network, *ACM Transactions on Knowledge Discovery from Data* **3** (2009), Article 9.
[11] M. Kimura, K. Saito, R. Nakano and H. Motoda, Finding influential nodes in a social network from information diffusion data, in: *Proceedings of the 2nd International Workshop on Social Computing, Behavioral Modeling and Prediction (SBP09)*, 2009, pp. 139–145.
[12] B. Klimt and Y. Yang, The Enron corpus: A new dataset for email classification research, in: *Proceedings of the 15th European Conference on Machine Learning (ECML 2004)*, 2004, pp. 217–226.
[13] J. Leskovec, L. Adamic and B.A. Huberman, The dynamics of viral marketing, *ACM Transactions on the Web* **1** (2007), Article 5.
[14] M.E.J. Newman, S. Forrest and J. Balthrop, Email networks and the spread of computer viruses, *Physical Review E* **66** (2002), Article 035101.
[15] M.E.J. Newman, The structure and function of complex networks, *SIAM Review* **45** (2003), 167–256.
[16] M.E.J. Newman and J. Park, Why social networks are different from other types of networks, *Physical Review E* **68** (2003), Article 036122.
[17] A.Y. Ng, A.X. Zheng and M.I. Jordan, Link analysis, eigenvectors and stability, in: *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001, pp. 903–910.
[18] G. Palla, I. Derényi, I. Farkas and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, *Nature* **435** (2005), 814–818.
[19] K. Saito, M. Kimura and H. Motoda, Effective visualization of information diffusion process over complex networks, in: *Proceedings of the 2008 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2008)*, 2008, pp. 326–341.
[20] S. Wasserman and K. Faust, *Social network analysis*, Cambridge University Press, 1994.
[21] D.J. Watts and P.S. Dodds, Influence, networks, and public opinion formation, *Journal of Consumer Research* **34** (2007), 441–458.