Learning Diffusion Probability Based on Node Attributes in Social Networks

Kazumi Saito[1], Kouzou Ohara[2], Yuki Yamagishi[1], Masahiro Kimura[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Integrated Information Technology, Aoyama Gakuin University
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[3] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. Information diffusion over a social network is analyzed by modeling the successive interactions of neighboring nodes as probabilistic processes of state changes. We address the problem of estimating parameters (diffusion probability and time-delay parameter) of the probabilistic model as a function of the node attributes from the observed diffusion data by formulating it as the maximum likelihood problem. We show that the parameters are obtained by an iterative updating algorithm which is efficient and is guaranteed to converge. We tested the performance of the learning algorithm on three real world networks assuming the attribute dependency, and confirmed that the dependency can be correctly learned. We further show that the influence degree of each node based on the link-dependent diffusion probabilities is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the true link-dependent diffusion probabilities.

1 Introduction

The growth of Internet has enabled to form various kinds of large-scale social networks, through which a variety of information, e.g. news, ideas, hot topics, malicious rumors, etc. spreads in the form of "word-of-mouth" communications, and it is noticeable to observe how much they affect our daily life style. The spread of information has been studied by many researchers [15,14,4,1,12,7,9]. The information diffusion models widely used are the *independent cascade (IC)* [2,5,7] and the *linear threshold (LT)* [21,22] models. They have been used to solve such problems as the *influence maximization problem* [5,8] and the *contamination minimization problem* [7,20]. These two models focus on different information diffusion aspects. The IC model is sender-centered (push type) and each active node *independently* influences its inactive

M. Kryszkiewicz et al. (Eds.): ISMIS 2011, LNAI 6804, pp. 153–162, 2011.

neighbors with given diffusion probabilities. The LT model is receiver-centered (pull type) and a node is influenced by its active neighbors if their total weight exceeds the threshold for the node.

What is important to note is that both models have parameters that need be specified in advance: diffusion probabilities for the IC model, and weights for the LT model. However, their true values are not known in practice. This poses yet another problem of estimating them from a set of information diffusion results that are observed as time-sequences of influenced (activated) nodes. This falls in a well defined parameter estimation problem in machine learning framework. Given a generative model with some parameters and the observed data, it is possible to calculate the likelihood that the data are generated and the parameters can be estimated by maximizing the likelihood. To the best of our knowledge, we are the first to follow this line of research. We addressed this problem for the IC model [16] and devised the iterative parameter updating algorithm.

The problem with both the IC and LT models is that they treat the information propagation as a series of state changes of nodes and the changes are made in a synchronous way, which is equivalent to assuming a discrete time step. However, the actual propagation takes place in an asynchronous way along the continuous time axis, and the time stamps of the observed data are not equally spaced. Thus, there is a need to extend both models to make the state changes asynchronous. We have, thus, extended both the models to be able to simulate asynchronous time delay (the extended models are called AsIC and AsLT models) and showed that the same maximum likelihood approach works nicely [17,18,19] and recently extended the same approach to opinion propagation problem using the value-weighted voter model with multiple opinions [10]. There are other works which are close to ours that also attempted to solve the similar problem by maximizing the likelihood [3,13], where the focus was on inferring the underlying network. In particular, [13] showed that the problem can effectively be transformed to a convex programming for which a global solution is guaranteed.

In this paper we also address the same problem using the AsIC model, but what is different from all of the above studies is that we try to learn the dependency of the diffusion probability and the time-delay parameter on the node attributes rather than learn it directly from the observed data. In reality the diffusion probability and the time-delay parameter of a link in the network must at least be a function of the attributes of the two connecting nodes, and ignoring this property does not reflect the reality. Another big advantage of explicitly using this relationship is that we can avoid overfitting problem. Since the number of links is much larger than the number of nodes even if the social network is known to be sparse, the number of parameters to learn is huge and we need prohibitively large amount of data to learn each individual diffusion probability separately. Because of this difficulty, many of the studies assumed that the parameter is uniform across different links or it depends only on the topic (not on the link that the topic passes through). Learning a function is much more realistic and does not require such a huge amount of data.

We show that the parameter updating algorithm is very efficient and is guaranteed to converge. We tested the performance of the algorithm on three real world networks assuming the attribute dependency of the parameters. The algorithm can correctly estimate both the diffusion probability and the time-delay parameter by way of node

attributes through a learned function, and we can resolve the deficiency of uniform parameter value assumption. We further show that the influence degree of each node based on the link-dependent diffusion probabilities (via learned function) is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the link-dependent diffusion probabilities, indicating that the uniform diffusion probability assumption is not justified if the true diffusion probability is link-dependent.

2 Diffusion Model

2.1 AsIC Model

To mathematically model the information diffusion in a social network, we first recall the AsIC model according to [19], and then extend it to be able to handle node attributes. Let $G = (V, E)$ be a directed network without self-links, where V and E ($\subset V \times V$) stand for the sets of all the nodes and links, respectively. For each node $v \in V$, let $F(v)$ be the set of all the nodes that have links from v, *i.e.*, $F(v) = \{u \in V;\ (v, u) \in E\}$, and $B(v)$ be the set of all the nodes that have links to v, *i.e.*, $B(v) = \{u \in V;\ (u, v) \in E\}$. We say a node is *active* if it has been influenced with the information; otherwise it is inactive. We assume that a node can switch its state only from inactive to active.

The AsIC model has two types of parameter $p_{u,v}$ and $r_{u,v}$ with $0 < p_{u,v} < 1$ and $r_{u,v} > 0$ for each link $(u, v) \in E$, where $p_{u,v}$ and $r_{u,v}$ are referred to as the diffusion probability and the time-delay parameter through link (u, v), respectively. Then, the information diffusion process unfolds in continuous-time t, and proceeds from a given initial active node in the following way. When a node u becomes active at time t, it is given a single chance to activate each currently inactive node $v \in F(u)$: u attempts to activate v if v has not been activated before time $t + \delta$, and succeeds with probability $p_{u,v}$, where δ is a delay-time chosen from the exponential distribution[1] with parameter $r_{u,v}$. The node v will become active at time $t + \delta$ if u succeed. The information diffusion process terminates if no more activations are possible.

2.2 Extension of AsIC Model for Using Node Attributes

In this paper, we extend the AsIC model to explicitly treat the attribute dependency of diffusion parameter through each link. Each node can have multiple attributes, each of which is either nominal or numerical. Let v_j be a value that node v takes for the j-th attribute, and J the total number of the attributes. For each link $(u, v) \in E$, we can consider the J-dimensional vector $\boldsymbol{x}_{u,v}$, each element of which is calculated by some function of u_j and v_j, *i.e.*, $x_{u,v,j} = f_j(u_j, v_j)$. Hereafter, for the sake of convenience, we consider the augmented $(J + 1)$-dimensional vector $\boldsymbol{x}_{u,v}$ by setting $x_{u,v,0} = 1$ as the link attributes. Then we propose to model both the diffusion probability $p_{u,v}$ and the time-delay parameter $r_{u,v}$ for each link $(u, v) \in E$ by the following formulae[2]:

$$p_{u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x}_{u,v})}, \quad r_{u,v} = r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) = \exp(\boldsymbol{\phi}^T \boldsymbol{x}_{u,v}), \qquad (1)$$

[1] We chose a delay-time from the exponential distribution in this paper for the sake of convenience, but other distributions such as power-law and Weibull can be employed.

[2] Note that both are simple and smooth functions of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ that guarantee $0 < p < 1$ and $r > 0$.

where $\boldsymbol{\theta}^T = (\theta_0, \cdots, \theta_J)$ and $\boldsymbol{\phi}^T = (\phi_0, \cdots, \phi_J)$ are the $(J+1)$-dimensional parameter vectors for diffusion probability and time-delay parameter, respectively. Note here that θ_0 and ϕ_0 correspond to the constant terms, and $\boldsymbol{\theta}^T$ stands for a transposed vector of $\boldsymbol{\theta}$.

Although our modeling framework does not depend on a specific form of function f_j, we limit the form to be the following: $x_{u,v,j} = \exp(-|u_j - v_j|)$ if the j-th node attribute is numerical; $x_{u,v,j} = \delta(u_j, v_j)$ if the j-th node attribute is nominal, where $\delta(u_j, v_j)$ is a delta function defined by $\delta(u_j, v_j) = 1$ if $u_j = v_j$; $\delta(u_j, v_j) = 0$ otherwise. Intuitively, the more similar u_j and v_j are, that is, the closer their attribute values are to each other, the larger the diffusion probability $p_{u,v}$ is if the corresponding parameter value θ_j is positive, and the smaller if it is negative. We can see the similar observation for the time-delay parameter $r_{u,v}$.

3 Learning Problem and Method

We consider an observed data set of M independent information diffusion results, $\mathcal{D}_M = \{D_m;\ m = 1, \cdots, M\}$. Here, each D_m represents a sequence of obserbation. It is given by a set of pairs of active node and its activation time, $D_m = \{(u, t_{m,u}),\ (v, t_{m,v}),\ \cdots\}$, and called the mth diffusion result. These sequences may partially overlap, *i.e.*, a node may appear in more than one sequence, but are treated separately according to the AsIC model. We denote by $t_{m,v}$ the activation time of node v for the mth diffusion result. Let T_m be the observed final time for the mth diffusion result. Then, for any $t \leq T_m$, we set $C_m(t) = \{v \in V;\ (v, t_{m,v}) \in D_m,\ t_{m,v} < t\}$. Namely, $C_m(t)$ is the set of active nodes before time t in the mth diffusion result. For convenience sake, we use C_m as referring to the set of all the active nodes in the mth diffusion result. For each node $v \in C_m$, we define the following subset of parent nodes, each of which had a chance to activate v, i.e., $\mathcal{B}_{m,v} = B(v) \cap C_m(t_{m,v})$.

3.1 Learning Problem

According to Saito et al. [17], we define the probability density $\mathcal{X}_{m,u,v}$ that a node $u \in \mathcal{B}_{m,v}$ activates the node v at time $t_{m,v}$, and the probability $\mathcal{Y}_{m,u,v}$ that the node v is not activated by a node $u \in \mathcal{B}_{m,v}$ within the time-period $[t_{m,u}, t_{m,v}]$.

$$\mathcal{X}_{m,u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) \exp(-r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})). \tag{2}$$

$$\mathcal{Y}_{m,u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) \exp(-r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})) + (1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})). \tag{3}$$

Then, we can consider the following probability density $h_{m,v}$ that the node v is activated at time $t_{m,v}$:

$$h_{m,v} = \sum_{u \in \mathcal{B}_{m,v}} \mathcal{X}_{m,u,v} \left(\prod_{z \in \mathcal{B}_{m,v} \setminus \{u\}} \mathcal{Y}_{m,z,v} \right) = \prod_{z \in \mathcal{B}_{m,v}} \mathcal{Y}_{m,z,v} \sum_{u \in \mathcal{B}_{m,v}} \mathcal{X}_{m,u,v} (\mathcal{Y}_{m,u,v})^{-1}. \tag{4}$$

Next, we consider the following probability $g_{m,v,w}$ that the node w is not activated by the node v before the observed final time T_m.

$$g_{m,v,w} = p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta}) \exp(-r(\boldsymbol{x}_{v,w}, \boldsymbol{\phi})(T_m - t_{m,v})) + (1 - p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta})). \tag{5}$$

Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e., $T_m \gg \max\{t_{m,v}; (v, t_{m,v}) \in D_m\}$. Thus, as $T_m \to \infty$ in Equation (5), we can assume

$$g_{m,v,w} = 1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}). \tag{6}$$

By using Equations (4) and (6), and the independence properties, we can define the likelihood function $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ by

$$\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi}) = \log \prod_{m=1}^{M} \prod_{v \in C_m} \left(h_{m,v} \prod_{w \in F(v) \setminus C_m} g_{m,v,w} \right). \tag{7}$$

In this paper, we focus on Equation (6) for simplicity, but we can easily modify our method to cope with the general one (i.e., Equation (5)). Thus, our problem is to obtain the values of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, which maximize Equation (7). For this estimation problem, we derive a method based on an iterative algorithm in order to stably obtain its solution.

3.2 Learning Method

Again, according to Saito et al. [17], we introduce the following variables to derive an EM like iterative algorithm.

$$\begin{aligned}
\mu_{m,u,v} &= \mathcal{X}_{m,u,v} (\mathcal{Y}_{m,u,v})^{-1} \Big/ \sum_{z \in \mathcal{B}_{m,v}} \mathcal{X}_{m,z,v} (\mathcal{Y}_{m,z,v})^{-1}. \\
\eta_{m,u,v} &= p_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) \,/\, \mathcal{Y}_{m,u,v}. \\
\xi_{m,u,v} &= \mu_{m,u,v} + (1 - \mu_{m,u,v}) \eta_{m,u,v}.
\end{aligned}$$

Let $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}$ be the current estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Similarly, let $\bar{\mathcal{X}}_{m,u,v}$, $\bar{\mathcal{Y}}_{m,u,v}$, $\bar{\mu}_{m,u,v}$, $\bar{\eta}_{m,u,v}$, and $\bar{\xi}_{m,u,v}$ denote the values of $\mathcal{X}_{m,u,v}$, $\mathcal{Y}_{m,u,v}$, $\mu_{m,u,v}$, $\eta_{m,u,v}$, and $\xi_{m,u,v}$ calculated by using $\bar{\boldsymbol{\theta}}$ and $\bar{\boldsymbol{\phi}}$, respectively.

From Equations (4), (6) and (7), we can transform our objective function $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ as follows:

$$\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi}) = Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) - \mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}), \tag{8}$$

where $Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is defined by

$$\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) &= Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) + Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) \\
Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) &= \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} (\bar{\xi}_{m,u,v} \log p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}) + (1 - \bar{\xi}_{m,u,v}) \log(1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})) \right. \\
&\quad \left. + \sum_{w \in F(v) \setminus C_m} \log(1 - p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta})) \right), \qquad (9) \\
Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) &= \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} \log r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi}) - \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})), \qquad (10)
\end{aligned}$$

and $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is defined by

$$\begin{aligned}
\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}}) = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} & (\bar{\mu}_{m,u,v} \log \mu_{m,u,v} \\
& + (1 - \bar{\mu}_{m,u,v})(\bar{\eta}_{m,u,v} \log \eta_{m,u,v} + (1 - \bar{\eta}_{m,u,v}) \log(1 - \eta_{m,u,v})) . \qquad (11)
\end{aligned}$$

Since $\mathcal{H}(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ is maximized at $\boldsymbol{\theta} = \bar{\boldsymbol{\theta}}$ and $\boldsymbol{\phi} = \bar{\boldsymbol{\phi}}$ from Equation (11), we can increase the value of $\mathcal{L}(\mathcal{D}_M; \boldsymbol{\theta}, \boldsymbol{\phi})$ by maximizing $Q(\boldsymbol{\theta}, \boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})$ (see Equation (8)).

We can maximize Q by independently maximizing Q_1 and Q_2 with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively. Here, by noting the definition of $p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})$ described in Equation (1), we can derive the gradient vector and the Hessian matrix of Q_1 as follows:

$$\frac{\partial Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\theta}} = \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} (\bar{\xi}_{m,u,v} - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})) \boldsymbol{x}_{u,v} - \sum_{w \in F(v) \setminus C_m} p(\boldsymbol{x}_{v,w}, \boldsymbol{\theta}) \boldsymbol{x}_{v,w} \right), \quad (12)$$

$$\frac{\partial^2 Q_1(\boldsymbol{\theta}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}^T} = - \sum_{m=1}^{M} \sum_{v \in C_m} \left(\sum_{u \in \mathcal{B}_{m,v}} \zeta_{u,v} \boldsymbol{x}_{u,v} \boldsymbol{x}_{u,v}^T + \sum_{w \in F(v) \setminus C_m} \zeta_{v,w} \boldsymbol{x}_{v,w} \boldsymbol{x}_{v,w}^T \right), \quad (13)$$

where $\zeta_{u,v} = p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta})(1 - p(\boldsymbol{x}_{u,v}, \boldsymbol{\theta}))$. We see that the Hessian matrix of Q_1 is non-positive definite, and thus, we can obtain the optimal solution of Q_1 by using the Newton method. Similarly, we can derive the gradient vector and the Hessian matrix of Q_2 as follows:

$$\frac{\partial Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi}} = \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} (\bar{\mu}_{m,u,v} - \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u})) \boldsymbol{x}_{u,v}, \quad (14)$$

$$\frac{\partial^2 Q_2(\boldsymbol{\phi}; \bar{\boldsymbol{\theta}}, \bar{\boldsymbol{\phi}})}{\partial \boldsymbol{\phi} \partial \boldsymbol{\phi}^T} = - \sum_{m=1}^{M} \sum_{v \in C_m} \sum_{u \in \mathcal{B}_{m,v}} \bar{\xi}_{m,u,v} r(\boldsymbol{x}_{u,v}, \boldsymbol{\phi})(t_{m,v} - t_{m,u}) \boldsymbol{x}_{u,v} \boldsymbol{x}_{u,v}^T. \quad (15)$$

The Hessian matrix of Q_2 is also non-positive definite, and we can obtain the optimal solution by Q_2. Note that we can regard our estimation method as a variant of the EM algorithm. We want to emphasize here that each time iteration proceeds the value of the likelihood function never decreases and the iterative algorithm is guaranteed to converge due to the convexity of Q.

4 Experimental Evaluation

We experimentally evaluated our learning algorithm by using synthetic information diffusion results generated from three large real world networks. Due to the page limitation, here we show only the results for the parameter vector $\boldsymbol{\theta}$, but we observed the similar results for the parameter vector $\boldsymbol{\phi}$. Note that $\boldsymbol{\phi}$ does not affect the influence degree used in our evaluation described later.

4.1 Dataset

We adopted three datasets of large real networks, which are all bidirectionally connected networks. The first one is a trackback network of Japanese blogs used in [7], and has 12,047 nodes and 79,920 directed links (the blog network). The second one is a network derived from the Enron Email Dataset [11] by extracting the senders and the recipients and linking those that had bidirectional communications and there were 4,254 nodes and 44,314 directed links (the Enron network). The last one is a network

Table 1. Absolute errors of estimated parameter values for each network. Values in parentheses are the assumed true values.

network	θ_0	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	θ_7	θ_8	θ_9	θ_{10}
Blog	0.0380	0.0587	0.1121	0.0941	0.0874	0.0873	0.0419	0.0723	0.0398	0.0400	0.0378
	(-2.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)
Enron	0.0371	0.0465	0.1152	0.0637	0.0758	0.0692	0.0382	0.0831	0.0400	0.0370	0.0385
	(-3.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)
Wikipedia	0.0485	0.0455	0.1505	0.0945	0.0710	0.0897	0.0444	0.1079	0.0438	0.0458	0.0434
	(-4.0)	(2.0)	(-1.0)	(0.0)	(0.0)	(0.0)	(1.0)	(-2.0)	(0.0)	(0.0)	(0.0)

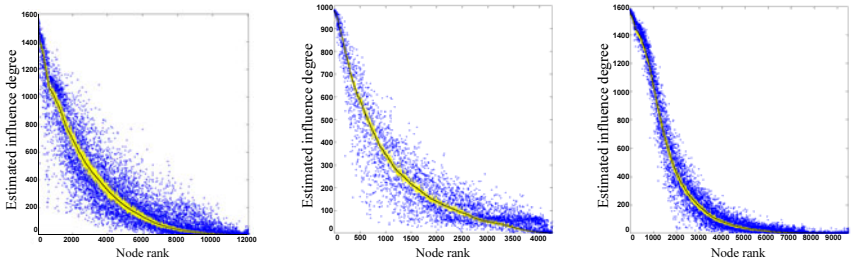
of people that was derived from the "list of people" within Japanese Wikipedia, used in [6], which has 9,481 nodes and 245,044 directed links (the Wikipedia network).

For each network, we generated synthetic information diffusion results in the following way: 1) artificially generate node attributes and determine their values in a random manner; 2) determine a parameter vector $\boldsymbol{\theta}$ which is assumed to be true; and then 3) generate 5 distinct information diffusion results, $\mathcal{D}_5 = \{D_1, \cdots, D_5\}$, each of which starts from a randomly selected initial active node, and contains at least 10 active nodes by the AsIC model mentioned in section 2.2. We generated a total of 10 attributes for every node in each network: 5 ordered attributes, each with a non-negative integer less than 20, and 5 nominal attributes, each with either 0, 1, or 2. The true parameter vector $\boldsymbol{\theta}$ was determined so that, according to [5], the average diffusion probability derived from the generated attribute values and $\boldsymbol{\theta}$ becomes smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. We refer to thus determined values as base values. The resulting average diffusion probability was 0.142 for the blog network, 0.062 for the Enron network, and 0.026 for the Wikipedia network, respectively.

4.2 Results

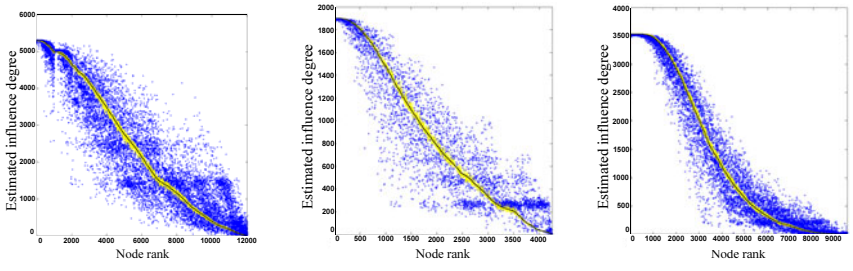
First, we examined the accuracy of parameter values $\hat{\boldsymbol{\theta}}$ estimated by our learning algorithm. Table 1 shows the absolute error $|\theta_i - \hat{\theta}_i|$ for each network which is the average over 100 trials, each obtained from a different $\mathcal{D}_5$ (we generated $\mathcal{D}_5$ 100 times.) where the values in the parentheses are true parameter values. On average, the absolute error of each parameter is 0.0645, 0.0586, and 0.0714 for the blog, Enron, and Wikipedia network, and their standard deviations are 0.0260, 0.0243, and 0.0338, respectively. This result shows that our learning method can estimate parameter values with very high accuracy regardless of networks. Note that $\theta_3, \theta_4, \theta_5, \theta_8, \theta_9$, and θ_{10} are set to 0. This is different from limiting the number of attributes to 4. The average computation time that our learning algorithm spent to estimate the parameter values was 2.96, 6.01, and 28.24 seconds for the blog, Enron, and Wikipedia network, respectively, which means that our learning method is very efficient (machine used is Intel(R) Xeon(R) CPU W5590 @3.33GHz with 32GB memory). Note that, from the derivation in Section 3.2, the computation time depends on the density of the network, i.e. the number of parents of a node.

Next, we evaluated our learning algorithm in terms of the influence degree of each node v which is defined as the expected number of active nodes after the information diffusion is over when v is chosen to be the initial active node. In this experiment,



(a) Blog (b) Enron (c) Wikipedia

Fig. 1. Comparison of three influence degrees σ (black solid line), $\hat{\sigma}$ (yellow marker) and $\bar{\sigma}$ (blue marker) for one particular run, randomly selected from the 100 independent trials in case that the diffusion probabilities are the base values



(a) Blog (b) Enron (c) Wikipedia

Fig. 2. Comparison of three influence degrees σ (black solid line), $\hat{\sigma}$ (yellow marker) and $\bar{\sigma}$ (blue marker) for one particular run, randomly selected from the 100 independent trials in case that the diffusion probabilities are larger than the base values

we derived the influence degree of each node by computing the empirical mean of the number of active nodes obtained from 1,000 independent runs which are based on the bond percolation technique described in [9]. Here, we compared the influence degree $\hat{\sigma}(v)$ of a node v which was derived using the parameter values estimated by our learning algorithm with the influence degree $\bar{\sigma}(v)$ which was derived by a naive way that uses the uniform diffusion probability approximated by averaging the true link-dependent diffusion probabilities.

Figure 1 presents three influence degrees σ, $\hat{\sigma}$, and $\bar{\sigma}$ for each node v for one particular run, randomly chosen from the 100 independent trials, where σ denotes the influence degree derived using the true link-dependent diffusion probability. The nodes are ordered according to the estimated true rank of influential degree. From these figures, we can observe that the difference between σ (solid line) and $\hat{\sigma}$ (yellow) is quite small, while the difference between σ and $\bar{\sigma}$ (blue) is very large and widely fluctuating. In fact, for $\hat{\sigma}$, the average of the absolute error defined as $|\hat{\sigma}(v) - \sigma(v)|$ over all nodes and all trials is 13.91, 6.80, and 8.32 for the blog, Enron, and Wikipedia network, and their standard deviations are 16.41, 7.08, and 11.31, respectively. Whereas, for $\bar{\sigma}$, the corresponding average of $|\bar{\sigma}(v) - \sigma(v)|$ is 77.75, 54.51, and 35.02, and their standard

deviations are 96.12, 57.80, and 51.84, respectively. Even in the best case for $\bar{\sigma}$ (the Wikipedia network), the average error for $\bar{\sigma}$ is about 4 times larger than that for $\hat{\sigma}$.

We further investigated how the error changes with the diffusion probabilities. Figure 2 is the results where the diffusion probabilities are increased, *i.e.*, larger influence degrees expected. To realize this, θ_0 is increased by 1 for each network, i.e. $\theta_0 = -1, -2$, and -3 for the blog, Enron, and Wikipedia network, respectively, which resulted in the corresponding average diffusion probability of 0.28, 0.14, and 0.063, respectively. It is clear that the difference between σ and $\hat{\sigma}$ remains very small, but the difference between σ and $\bar{\sigma}$ becomes larger than before (Fig. 1). Actually, for $\hat{\sigma}$, the average (standard deviation) of the absolute error over all nodes and all trials is 47.95 (28.03), 13.27 (12.30), and 15.11 (16.25) for the blog, Enron, and Wikipedia network, respectively, while, for $\bar{\sigma}$, the corresponding average (standard deviation) is 518.94 (502.05), 162.56 (159.40), and 163.51 (205.17), respectively. These results confirm that $\hat{\sigma}$ remains close to the true influence degree regardless of the diffusion probability p, while $\bar{\sigma}$ is very sensitive to p.

Overall, we can say that our learning algorithm is useful for estimating the influence degrees of nodes in a network, provided that we have some knowledge of dependency of diffusion probability on the selected attributes. It can accurately estimate them from a small amount of information diffusion results and avoid the overfitting problem.

5 Conclusion

Information diffusion over a social network is analyzed by modeling the cascade of interactions of neighboring nodes as probabilistic processes of state changes. The number of the parameters in the model is in general as many as the number of nodes and links, and amounts to several tens of thousands for a network of node size about ten thousands. In this paper, we addressed the problem of estimating link-dependent parameters of probabilistic information diffusion model from a small amount of observed diffusion data. The key idea is not to estimate them directly from the data as has been done in the past studies, but to learn the functional dependency of the parameters on the small number of node attributes. The task is formulated as the maximum likelihood estimation problem, and an efficient parameter update algorithm that guarantees the convergence is derived. We tested the performance of the learning algorithm on three real world networks assuming a particular class of attribute dependency, and confirmed that the dependency can be correctly learned even if the number of parameters (information diffusion probability of each link in this paper) is several tens of thousands. We further showed that the influence degree of each node based on the link-dependent diffusion probabilities is substantially different from that obtained assuming a uniform diffusion probability which is approximated by the average of the true link-dependent diffusion probabilities. This indicates that use of uniform diffusion probability is not justified if the true distribution is non-uniform, and affects the influential nodes and their ranking considerably.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research under Grant No. AOARD-10-4053.

References

1. Domingos, P.: Mining social networks for viral marketing. IEEE Intell. Syst. 20, 80–82 (2005)
2. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters 12, 211–223 (2001)
3. Gomez-Rodriguez, M., Leskovec, J., Krause, A.: Inferring networks of diffusion and influence. In: KDD, pp. 1019–1028 (2010)
4. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations 6, 43–52 (2004)
5. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: KDD, pp. 137–146 (2003)
6. Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: AAAI 2008, pp. 1175–1180 (2008)
7. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Trans. Knowl. Discov. Data 3, 9:1–9:23 (2009)
8. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: AAAI 2007, pp. 1371–1376 (2007)
9. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Min. and Knowl. Disc. 20, 70–97 (2010)
10. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Learning to predict opinion share in social networks. In: AAAI 2010, pp. 1364–1370 (2010)
11. Klimt, B., Yang, Y.: The enron corpus: A new dataset for email classification research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
12. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: EC 2006, pp. 228–237 (2006)
13. Myers, S.A., Leskovec, J.: On the convexity of latent social network inference. In: Proceedings of Neural Information Processing Systems (NIPS)
14. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. 45, 167–256 (2003)
15. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Phys. Rev. E 66, 035101 (2002)
16. Saito, K., Kimura, M., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: SBP 2009, pp. 138–145 (2009)
17. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Learning continuous-time information diffusion model for social behavioral data analysis. In: Zhou, Z.-H., Washio, T. (eds.) ACML 2009. LNCS, vol. 5828, pp. 322–337. Springer, Heidelberg (2009)
18. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Behavioral analyses of information diffusion models by observed data of social network. In: Chai, S.-K., Salerno, J.J., Mabry, P.L. (eds.) SBP 2010. LNCS, vol. 6007, pp. 149–158. Springer, Heidelberg (2010)
19. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Selecting information diffusion models over social networks for behavioral analysis. In: ECML PKDD 2010, pp. 180–195 (2010)
20. Tong, H., Prakash, B.A., Tsoourakakis, C., Eliassi-Rad, T., Faloutsos, C., Chau, D.H.: On the vulnerability of large graphs. In: Perner, P. (ed.) ICDM 2010. LNCS, vol. 6171, pp. 1091–1096. Springer, Heidelberg (2010)
21. Watts, D.J.: A simple model of global cascades on random networks. PNAS 99, 5766–5771 (2002)
22. Watts, D.J., Dodds, P.S.: Influence, networks, and public opinion formation. J. Cons. Res. 34, 441–458 (2007)