Latent Feature Independent Cascade Model for Social Propagation

Yuya Yoshikawa[1], **Tomoharu Iwata**[2], **and Hiroshi Sawada**[3]
[1]Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan
[2]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[3]NTT Service Evolution Laboratories, NTT Corporation, Kanagawa, Japan

Abstract—*People share various types of information including opinions on hot topics, bookmarking activity and rumors via online communities. To make it possible to predict future trends in online communities, it is important that we develop a model of information diffusion through social networks and a method for estimating its parameters. In this paper, we present a latent feature independent cascade model, which can effectively estimate diffusion probabilities by capturing the influences between latent communities. In particular, we incorporate two types of latent features for each node. The first represents the features as a sender and the second represents the features as a receiver. We demonstrate experimentally that the proposed model can estimate the diffusion probabilities more accurately than commonly used methods. We also show the effectiveness of the proposed model for estimating information spread.*

Keywords: information diffusion model, independent cascade, social network, latent feature model

1. Introduction

In online communities, various types of information including opinions on hot topics, bookmarking activity and rumors are shared between individuals by word of mouth. Based on the facts that the user activations in online communities are reflected in the box-office performance of movies [1], market prices [2] and the polling number for elections [3], there is great interest in predicting future trends and discovering instances where information is shared on social networks [4], [5], [6].

Various diffusion models have been proposed for simulating the information diffusion behavior on social networks [7], [8], [9], [10], [11]. The Independent Cascade Model (ICM) proposed by Kempe et al. [7] has been particularly well-studied in recent years, and is used for addressing the influence maximization/minimization problem [7], [12] and finding influential nodes [13], [14], [15]. ICM is a simple probabilistic model that describes processes by which pieces of information spread from node to node on a social network, where the behavior is based on the diffusion probability of each link. Thus, to simulate the real information diffusion behavior, it is important to learn the diffusion probabilities of all links precisely.

Some methods have been developed for estimating the diffusion probability parameters [16], [17], [18], [19]. Saito et al. developed an estimation method based on the EM algorithm under the assumption that continuous time delays occur between the activations [17], while Gruhl et al. assumed discrete time delays [16]. Although their methods provide ways to obtain the diffusion probabilities given the observations of activity, the low generalization performance results when the observations are insufficient. For example, information diffusion does not occur abundantly throughout the network, or often occurs in one portion of the network but not in another. Such cases lead to a poor parameter estimation result.

In realistic social networks, each node has attribute information such as affiliation, age and gender. We can expect the estimation performance regarding diffusion probabilities to improve by using these attributes. However, whether or not the attributes can be observed depends on the target applications.

In this paper, we propose a Latent Feature Independent Cascade Model (LFICM), which is designed to estimate the diffusion probabilities effectively. In the LFICM, we incorporate two types of latent features for each node. The first represents the features as a sender and the second represents the feature as a receiver. The diffusion probabilities are generated based on the latent features between the nodes of each link. By incorporating the latent features, we can estimate the diffusion probabilities with high generalization performance, since the LFICM has a smaller number of parameters than a conventional ICM. For the LFICM, we developed a parameter estimation method based on the EM algorithm. Although a method that estimates the diffusion probabilities based on the observed attribute features of each node has already been developed [18], the proposed model can estimate the ICM parameters without observing additional attribute features by treating the features as latent variables. In our experiments, we show that the proposed model can estimate the diffusion probabilities better than the conventional parameter estimation methods using three real network structures and synthetic activation data. We also show that in a simulation-based influence estimation method, the estimated influence degrees behave in much the same way as the true influence degrees.

2. Proposed Method

In this section, we present our proposed model, the Latent Feature Independent Cascade Model (LFICM), and

Table 1: Notation of LFICM

K	Number of dimensions of latent feature vectors
$\boldsymbol{I}$	Set of pieces of information
$\boldsymbol{x}_u$	Latent feature vector of node u as sender
$\boldsymbol{y}_u$	Latent feature vector of node u as receiver
$\boldsymbol{d}_i$	Diffusion sequence for information i
κ_{uv}	Diffusion probability from node u to node v
r	Time-delay parameter
γ	Bias parameter
σ_X	Standard deviation of $\boldsymbol{x_u}$ (hyperparameter)
σ_Y	Standard deviation of $\boldsymbol{y_u}$ (hyperparameter)

its parameter estimation method.

2.1 Model

Suppose that a set of pieces of information $\boldsymbol{I}$ spreads over a directed social network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes corresponding to individuals and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of links corresponding to relationships between individuals. We define $B(v)$ as a set of parents of node $v \in \mathcal{V}$, $B(v) = \{u|(u,v) \in \mathcal{E}\}$, and $F(v)$ as a set of children of node $v \in \mathcal{V}$, $F(v) = \{w|(v,w) \in \mathcal{E}\}$. Table 1 lists the symbols and the descriptions used in LFICM.

For a piece of information $i \in \boldsymbol{I}$, we observe logs showing when each node transmitted the information i. We refer to a status of a node from which information is transmitted as *active*. The logs form a sequence of length L_i of active node-time pairs as follows,

$$\boldsymbol{d}_i = \{(v_{i1}, t_{v_{i1}}), (v_{i2}, t_{v_{i2}}), \cdots (v_{iL_i}, t_{v_{iL_i}})\}.$$

Note that each active node can be affected by any of its parent nodes, but we cannot observe by whom the active nodes are influenced.

LFICM assumes that the pieces of information spread according to the same mechanism as with the Independent Cascade Model (ICM) [7]. ICM provides a process whereby the information spreads from node to node through the links. ICM has two types of model parameters for each link $(u, v) \in \mathcal{E}$, *diffusion probability* κ_{uv} and *time-delay parameter* r_{uv}, where $0 \leq \kappa_{uv} \leq 1$ and $r_{uv} > 0$. For simplicity, we assume $r_{uv} = r$ although it is easy to run.

The diffusion process of the ICM is as follows. We first fix a set of source nodes $\mathcal{S} \subseteq \mathcal{V}$ from which the information diffusion process starts. Thus, node $v \in \mathcal{S}$ becomes active at time $t_v = 0$. Then the process iteratively executes the following two steps until no more activations are possible:

- When node u becomes active, it attempts to transmit information to each inactive child node $v \in F(u)$. This trial succeeds with the diffusion probability κ_{uv}.
- If node v is activated by node u in the above step, then the activated time of node v is $t_u + \Delta$ where Δ is a random variable following an exponential distribution with parameter r given below:

$$\Delta \sim \text{Exponential}(r) \tag{1}$$

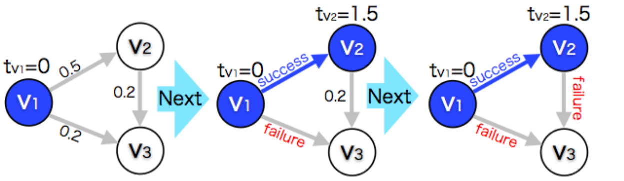

Fig. 1: Step by step procedure of the ICM. **(left)** Initial state. Diffusion probabilities are assigned to each link in advance. Node v_1 is a source node. **(center)** Node v_2 becomes active when affected by node v_1. **(right)** Node v_3 is not affected by node v_1 or v_2.

Figure 1 shows the step-by-step procedure of ICM.

To extend ICM for estimating the diffusion probabilities with high generalization performance, we introduce two types of K-dimension latent feature vectors $\boldsymbol{x}_u \in \mathbb{R}^K$ and $\boldsymbol{y}_u \in \mathbb{R}^K$for each node $u \in \mathcal{V}$. In LFICM, the diffusion probability from node u to node v, κ_{uv}, is calculated as follows:

$$\kappa_{uv} = f(\boldsymbol{x}_u, \boldsymbol{y}_v, \gamma) = \left(1 + \exp(-\boldsymbol{x}_u^\top \boldsymbol{y}_v - \gamma)\right)^{-1}, \tag{2}$$

where γ is a bias parameter that does not depend on the nodes. Function f is a sigmoid function, thus $0 \leq \kappa_{uv} \leq 1$. Here, $\boldsymbol{x}_u$ and $\boldsymbol{y}_u$ represent the features as an information sender and the features as an information receiver, respectively. The diffusion probability κ_{uv} has a high value when $\boldsymbol{x}_u^\top \boldsymbol{y}_v$ is high.

The latent features of node $u \in \mathcal{V}$, $\boldsymbol{x}_u$ and $\boldsymbol{y}_u$ follow a K-dimension normal distribution,

$$\begin{aligned} \boldsymbol{x}_u &\sim \mathcal{N}(\mathbf{x}_u|\mathbf{0}, \sigma_X^2\mathbf{I}), && (3) \\ \boldsymbol{y}_u &\sim \mathcal{N}(\mathbf{y}_u|\mathbf{0}, \sigma_Y^2\mathbf{I}), && (4) \end{aligned}$$

where $\mathbf{0}$ and $\mathbf{I}$ denote K-dimension zero vector and a unit matrix of size $K \times K$, respectively.

Under the calculated parameters described above, a diffusion sequence $\boldsymbol{d}_i$ is generated based on the ICM diffusion process shown in Algorithm 1 given source nodes $\mathcal{S}_i \subseteq \mathcal{V}$,

$$\boldsymbol{d}_i \sim \text{ICM}\left(\{\kappa_{uv}\}_{(u,v)\in\mathcal{E}}, r, \mathcal{G}, \mathcal{S}_i\right).$$

2.2 Parameter Estimation

We present a parameter estimation algorithm for the LFICM based on the Expectation-Maximization (EM) algorithm [20].

Given a set of diffusion sequences $\boldsymbol{D} = \{\boldsymbol{d}_i\}_{i\in\boldsymbol{I}}$, we can write the posterior probability for LFICM parameters $\boldsymbol{X} = [\boldsymbol{x}_u]_{u\in\mathcal{V}}, \boldsymbol{Y} = [\boldsymbol{y}_u]_{u\in\mathcal{V}}$ as follows,

$$\begin{aligned} & P(\boldsymbol{X}, \boldsymbol{Y} \mid \gamma, r, \boldsymbol{D}, \sigma_X, \sigma_Y, \mathcal{G}) && (5) \\ & \quad \propto \; P(\boldsymbol{D}|\boldsymbol{X}, \boldsymbol{Y}, \gamma, r, \mathcal{G}) P(\boldsymbol{X}|\sigma_X) P(\boldsymbol{Y}|\sigma_Y). \end{aligned}$$

The first factor on the right hand side of Eq. (5) is the likelihood of LFICM. Let $\Delta_{uv}^{(i)}$ be the difference between the active times of nodes u and v for information i, i.e.,

$\Delta^{(i)}_{uv} = t_{iv} - t_{iu}$. For convenience, let C_i and $C_i(t)$ be a set of active nodes and a set of active nodes by time t for information i, respectively, that is,

$$C_i = \{v \mid (v,t) \in \boldsymbol{d}_i\}, \quad C_i(t) = \{v \mid (v,t') \in \boldsymbol{d}_i,\ t' < t\}.$$

Although we omit the detailed derivation of the likelihood due to space limitations, we can obtain the following likelihood,

$$\begin{aligned} & P(\boldsymbol{D}|\boldsymbol{X},\boldsymbol{Y},\boldsymbol{\gamma},r,\mathcal{G}) \qquad (6) \\ = & \prod_{i\in\boldsymbol{I}} \prod_{v\in C_i} \prod_{u\in B(v)\cap C_i(t_{iv})} p^{(i)}_{u\to v} \sum_{u\in B(v)\cap C_i(t_{iv})} \frac{p^{(i)}_{u\to v}}{p^{(i)}_{u\not\to v}} \\ & \times \prod_{w\in F(v)\setminus C_i} (1-\kappa_{vw}), \end{aligned}$$

where $p^{(i)}_{u\to v}$ represents the probability that node u makes node v active, and $p^{(i)}_{u\not\to v}$ represents the probability that node u fails to affect node v,

$$\begin{aligned} p^{(i)}_{u\to v} &= \kappa_{uv} r \exp(-r\Delta^{(i)}_{uv}), & (7) \\ p^{(i)}_{u\not\to v} &= \kappa_{uv} \exp(-r\Delta^{(i)}_{uv}) + 1 - \kappa_{uv}. & (8) \end{aligned}$$

The second and third factors are prior distributions for $\boldsymbol{X}$ and $\boldsymbol{Y}$, respectively,

$$\begin{aligned} P(\boldsymbol{X}|\sigma_X) &= \prod_{u=1}^{|\mathcal{V}|} \frac{1}{\sqrt{2\pi}\sigma_X^K} \exp\Big(-\frac{\boldsymbol{x}_u^\top \boldsymbol{x}_u}{2\sigma_X^2}\Big), & (9) \\ P(\boldsymbol{Y}|\sigma_Y) &= \prod_{u=1}^{|\mathcal{V}|} \frac{1}{\sqrt{2\pi}\sigma_Y^K} \exp\Big(-\frac{\boldsymbol{y}_u^\top \boldsymbol{y}_u}{2\sigma_Y^2}\Big). & (10) \end{aligned}$$

With the posterior probability Eq. (5), we find parameters $\hat{\boldsymbol{X}}$, $\hat{\boldsymbol{Y}}$, $\hat{\gamma}$ and $\hat{r}$ based on the maximum a posteriori (MAP) principle. The parameters can be estimated with an EM algorithm that alternates between estimating which active nodes are affected by the E-step, and updating the parameters under the E-step result in the M-step. We use $\bar{x}$ as a current estimate for variable x to avoid confusion.

E-step: In the E-step, we need only to consider the likelihood Eq. (6) in the posterior probability Eq. (5). By employing a similar method to that described in [17], we can derive function Q, which is the expectation of the complete-data likelihood, from the likelihood Eq. (6) as follows,

$$\begin{aligned} & Q(\boldsymbol{X},\boldsymbol{Y},\gamma,r;\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}},\bar{\gamma},\bar{r}) \qquad (11) \\ = & \sum_{i\in\boldsymbol{I}} \sum_{v\in C_i} \Bigg[\sum_{u\in B(v)\cap C_i(t_{iv})} \Big(\bar{\xi}^{(i)}_{uv} \log \kappa_{uv} \\ & + (1-\bar{\xi}^{(i)}_{uv}) \log(1-\kappa_{uv}) + \bar{q}^{(i)}_{uv} \log r + \bar{\xi}^{(i)}_{uv} r \Delta^{(i)}_{uv} \Big) \\ & + \sum_{w\in F(v)\setminus C_i} \log(1-\kappa_{vw}) \Bigg] \\ & + \log P(\boldsymbol{X}|\sigma_X) + \log P(\boldsymbol{Y}|\sigma_Y) \end{aligned}$$

Algorithm 1 ICMGENERATOR

```
Require: diffusion probability ${\kappa_{uv}}_{(u,v)\in\mathcal{E}}$, time-delay parameter $r$, network $\mathcal{G}$, source nodes $\mathcal{S}$
$d_i \leftarrow \{(u,0) \mid u \in \mathcal{S}\}$
repeat
    $(u, t_{iu}) \leftarrow \min_{t_{ix}}\{(x, t_{mx}) \mid (x, t_{mx}) \in d_i, x \in \mathcal{S}\}$
    for $v \in$ {inactive nodes in child nodes of $u$ } do
        if $u$ succeeds in propagation to $v$ with $\kappa_{uv}$ then
            $t_{iv} \leftarrow t_{iu} + \Delta$, where $\Delta$ follows Eq. (1).
            $d_i \leftarrow d_i \cup \{(v, t_{iv})\}$
            $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$
        end if
    end for
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{u\}$
until $S = \emptyset$
return $d_i$
```

where

$$\begin{aligned} q^{(i)}_{uv} &= \frac{p^{(i)}_{u\to v}/p^{(i)}_{u\not\to v}}{\sum_{u'\in B(v)\cap C_i(t_{iv})} p^{(i)}_{u\to v}/p^{(i)}_{u\not\to v}}, & (12) \\ \eta^{(i)}_{uv} &= \frac{\kappa_{uv}\exp(-r\Delta^{(i)}_{uv})}{\kappa_{uv}\exp(-r\Delta^{(i)}_{uv}) + (1-\kappa_{uv})}, & (13) \\ \xi^{(i)}_{uv} &= q^{(i)}_{uv} + (1-q^{(i)}_{uv})\eta^{(i)}_{uv}. & (14) \end{aligned}$$

Here, function Q is guaranteed to be a lower bound for the posterior Eq. (5), that is, $P(\boldsymbol{X},\boldsymbol{Y} \mid \boldsymbol{\gamma}, r, \boldsymbol{D}, \sigma_X, \sigma_Y, \mathcal{G}) \geq Q$. $q^{(i)}_{uv}$ can be regarded as the probability that node u is influenced by node v on information i under the current estimates.

In the M-step, we estimate the parameters by maximizing the function Q.

M-step: Using the current estimates, $\bar{\boldsymbol{X}}, \bar{\boldsymbol{Y}}, \bar{r}, \bar{\gamma}, \bar{q}^{(i)}_{uv}, \bar{\eta}^{(i)}_{uv}$ and $\bar{\xi}^{(i)}_{uv}$, we update the parameters, $\boldsymbol{X}, \boldsymbol{Y}, r$ and γ. The closed form solution for Eq. (11) does not exist for the parameters, $\boldsymbol{X}, \boldsymbol{Y}$ and γ owing to the non-linearity of the sigmoid function. Thus, we need to use a kind of optimization method based on a gradient with respect to each parameter. In this study, we use the quasi-Newton method, which only needs first-order derivations with respect to the parameters.

The first-order derivations of the parameters, $\boldsymbol{x}_u, \boldsymbol{y}_u$ for each $u \in \mathcal{V}$ and γ are derived as follows:

$$\begin{aligned} & \frac{\partial Q(\boldsymbol{X},\boldsymbol{Y},\gamma,r;\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}},\bar{\gamma},\bar{r})}{\partial \boldsymbol{x}_u} \qquad (15) \\ & = \sum_{i\in\boldsymbol{I}} \Big(\sum_{v\in F(u)\cap C_i} \big(\bar{\xi}^{(i)}_{uv} - f(\boldsymbol{x}_u, \bar{\boldsymbol{y}}_v, \bar{\gamma})\big) \bar{\boldsymbol{y}}_v \\ & \quad - \sum_{w\in F(u)\setminus C_i} f(\boldsymbol{x}_u, \bar{\boldsymbol{y}}_w, \bar{\gamma}) \bar{\boldsymbol{y}}_w \Big) - \frac{1}{\sigma_X^2} \boldsymbol{x}_u \end{aligned}$$

$$\frac{\partial Q(\boldsymbol{X},\boldsymbol{Y},\gamma,r;\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}},\bar{\gamma},\bar{r})}{\partial \boldsymbol{y}_v} \tag{16}$$
$$= \sum_{i\in\boldsymbol{I}}\Big(\sum_{u\in B(v)\cap C_i(t_{iv})}\big(\bar{\xi}^{(i)}_{uv} - f(\bar{\boldsymbol{x}}_u,\boldsymbol{y}_v,\bar{\gamma})\big)\bar{\boldsymbol{x}}_u - \sum_{s\in B(v)\cap C_i, v\notin C_i} f(\bar{\boldsymbol{x}}_s,\boldsymbol{y}_v,\bar{\gamma})\bar{\boldsymbol{x}}_s\Big) - \frac{1}{\sigma_Y^2}\boldsymbol{y}_v$$

$$\frac{\partial Q(\boldsymbol{X},\boldsymbol{Y},\gamma,r;\bar{\boldsymbol{X}},\bar{\boldsymbol{Y}},\bar{\gamma},\bar{r})}{\partial \gamma} \tag{17}$$
$$= \sum_{i\in\boldsymbol{I}}\Big(\sum_{v\in C_i}\sum_{u\in B(v)\cap C_i(t_{iv})}\big(\bar{\xi}^{(i)}_{uv} - f(\bar{\boldsymbol{x}}_u,\bar{\boldsymbol{y}}_v,\gamma)\big) - \sum_{v\in C_i}\sum_{s\in F(v)\setminus C_i} f(\bar{\boldsymbol{x}}_u,\bar{\boldsymbol{y}}_s,\gamma)\Big)$$

The time-delay parameter r can be calculated using the following closed form,

$$r = \frac{\sum_{i\in\boldsymbol{I}}\sum_{v\in C_i}\sum_{u\in B(v)\cap C_i(t_{iv})}\bar{q}^{(i)}_{uv}}{\sum_{i\in\boldsymbol{I}}\sum_{v\in C_i}\sum_{u\in B(v)\cap C_i(t_{iv})}\bar{\xi}^{(i)}_{uv}\Delta^{(i)}_{uv}}. \tag{18}$$

In the EM algorithm, the parameters are estimated by alternating E-step and M-step and continuing the procedure until the improvement of the log-likelihood converges. In summary, the parameter estimation procedure is given by Algorithm 2.

Algorithm 2 LFICMESTIMATOR

```
Require: network $\mathcal{G}$, diffusion sequences $\boldsymbol{D}$, dimension of
    features $K$, hyper-parameters $\sigma_X$ and $\sigma_Y$
Initialize $\boldsymbol{X}, \boldsymbol{Y}, \gamma, r$
repeat
    E-step: update $q_{muv}, \eta_{muv}, \xi_{muv}$ based on Eqs. (12),(13) and (14)
    M-step: update $\boldsymbol{X}, \boldsymbol{Y}, \gamma$ using quasi-Newton method
        based on Eqs. (15) - (17), and $r$ based on Eq. (18)
until convergence of improving the log-likelihood Eq. (5)
return $\boldsymbol{X}, \boldsymbol{Y}, \gamma, r$
```

3. Experiments

To evaluate the proposed model with respect to the precision and effectiveness of the estimated diffusion probabilities of the ICM, we ran experiments using real networks and synthetic diffusion sequences.

3.1 Experimental Data and Settings

Three real networks data. In this study, we used three kinds of real network structure datasets. The first is BLOG data, which we obtained by tracing the track-back of posts in the *goo* blog in May 2005 [17]. The network is constructed by putting a link from blog (node) u to blog v if a post on blog u refers to one on blog v through the track-back function. The second dataset is referred to as ENRON data and consists of exchanges of e-mail in Enron Corp. The network regards each sender and receiver as a node and puts a link if node u sends an e-mail to node v. The last dataset is referred to as MIXI data, which is bidirectional (co-link) friendship network data obtained from a well-known social networking service in Japan[1].

Table 2: Parameter settings for generating synthetic diffusion sequences

	K	γ	r	$\mathcal{S}$	σ_X	σ_Y
BLOG	5	-3.5	10.0	Random	1.0	1.0
ENRON	5	-6.5	1.0	Random	1.0	1.0
MIXI	7	-6.5	1.0	Random	1.0	1.0

Table 3: Statistics of network data for evaluation of LFICM

	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	$\lvert\boldsymbol{I}\rvert$	avg. $\lvert\boldsymbol{d}_i\rvert$
BLOG	12,047	79,920	200	207.1
ENRON	36,692	367,662	200	203.4
MIXI	80,608	571,136	1,500	8.6

Synthetic diffusion sequence generation. In our experiments, diffusion sequences are generated artificially based on the proposed model. The procedure for generating the diffusion sequences is as follows. First we generate the latent feature vectors under the parameter settings for each dataset shown in Table 2, and calculate the diffusion probability for each link. Then we generate diffusion sequences $\boldsymbol{d}_i$ based on Algorithm 1. Table 3 shows the statistics of generating diffusion sequences. Note that there are too few diffusion sequences for complete estimation, as suggested by a comparison of the number of links with the volume of the sequences.

Baseline methods. We use two parameter estimation methods as baselines for comparison, which are adopted in [17], [19]. The first method attempts to estimate the parameter directly by the maximum likelihood method [17]. This method is the basis of our model. We refer to it as `SaitoICM`. The second method is identical to `SaitoICM` but estimates a uniform diffusion probability throughout the network, that is, $\kappa_{uv} = \kappa$. We refer to it as `SimpleICM`. This method makes it possible to estimate the parameters robustly, but it is not a flexible model.

3.2 Precision of Parameter Estimation

In the first experiment, we evaluate the proposed method and the baselines by estimating the diffusion probabilities from synthetic diffusion sequences for each dataset. Since the proposed model controls the complexity of the model

[1] https://mixi.jp/

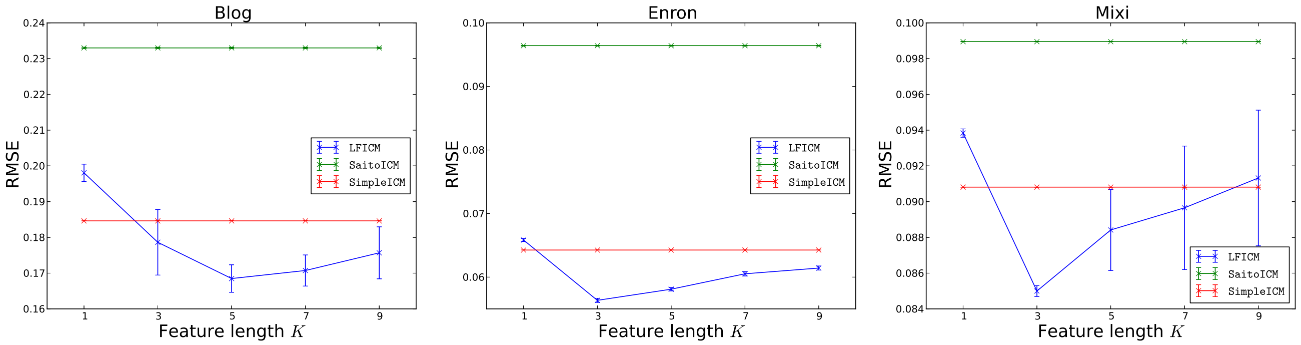

Fig. 2: Average RMSE between true and estimated diffusion probabilities in each dataset

by changing the value of K, we adopt K value of 1, 3, 5, 7 and 9. Note that the true K values, which are used for generating synthetic diffusion sequences for BLOG, ENRON and MIXI, are five, five and seven, respectively.

Figure 2 shows comparison results of the estimation error between the proposed method and each baseline for each dataset. The vertical axis denotes the root mean squared error (RMSE) between the true and estimated diffusion probabilities, while the horizontal axis denotes the dimension of the feature vector, K. In the results using BLOG and ENRON data, respectively, the proposed model outperformed the baselines except for when $K = 1$. The result obtained using MIXI data is characterized by a large standard deviation shown as a vertical bar on the proposed model. Nevertheless, `LFICM` outperformed the baselines when $K = 3, 5, 7$.

To summarize these results, `SaitoICM` seems to be a poor estimator compared with `LFICM` and `SimpleICM`, even though it is the most flexible model. This is because `SaitoICM` poses an over-fitting problem due to a lack of observed data.

3.3 Estimation of degree of influence

Estimation method for degree of influence. Let us define the *degree of influence* of node u as the average number of active nodes affected by source node u. Thus, it consists of the number of nodes that receive information transmitted from node u directly and indirectly. We have studied a method for estimating the degree using the ICM [15]. This method is largely dependent on the performance of a selected parameter estimation method. We simplify the algorithm of [15] into Algorithm 3, and evaluate the estimated degree of influence using the parameters with the proposed model and the baseline values. In this experiment, we fix $T = 50$ in Algorithm 3.

Evaluation method. We evaluate our model according to the following procedure:

Algorithm 3 INFLUENCEPREDICTOR

```
Require: network $\mathcal{G}$, source nodes $\mathcal{S}$, time-delay parameter $r$, diffusion probability $\{\kappa_{uv}\}_{(u,v)\in\mathcal{E}}$, trials $T$.
influence $\leftarrow 0$
for $i \leftarrow 1, 2, \cdots, T$ do
    $\boldsymbol{d}_m \leftarrow$ ICMGENERATOR$(\{\kappa_{uv}\}_{(u,v)\in\mathcal{E}}, r, \mathcal{G}, \mathcal{S})$
    $influence \leftarrow influence + \frac{1}{T}|\boldsymbol{d}_i|$
end for
return $influence$
```

1) For each node, we calculate the degree of influence using Algorithm 3 under the true parameter settings shown in Table 2, and we define it as the true degree of influence for each node.
2) We calculate the degree of influence under the parameters estimated in Section 3.2, by `LFICM`, `SaitoICM` and `SimpleICM`, respectively, and we define them as the estimated degree of influence of each method for each node.
3) We calculate the *Pearson correlation coefficient* and *Kendall's tau coefficient* between the true and estimated degrees of influence for each method.

where, the Pearson correlation coefficient is calculated simply using the degree of influence, while Kendall's tau coefficient uses the rank of degree of influence for each method. The values of these coefficients range from -1 to $+1$, and if the value is close to $+1$, then we can consider that the two compared values behave in the same way.

Experimental results. Tables 4 and 5 show results of an evaluation of the Pearson correlation coefficient and Kendall's tau coefficient between the true and estimated degree of influence, respectively. They are significantly correlated with significant probability $p < 0.01$. As shown in these tables, with any of the proposed methods, `LFICM`, is better than the baseline methods for each dataset.

However, we find that the K value of at which highest correlation is reached does not necessarily coincide with that one of the lowest RMSE. With the evaluation by RMSE of the estimated diffusion probabilities, all the diffusion

Table 4: Pearson correlation coefficient between the true and estimated degree influence .

	LFICM ($K = 1$)	LFICM ($K = 3$)	LFICM ($K = 5$)	LFICM ($K = 7$)	LFICM ($K = 9$)	SaitoICM	SimpleICM
Blog	0.360	0.734	**0.848**	0.828	0.769	0.753	0.827
Enron	0.442	**0.730**	0.703	0.669	0.657	0.539	0.656
Mixi	0.174	0.542	0.545	**0.556**	0.484	0.468	0.382

Note: numbers in bold indicate the best method for each set of data.

Table 5: Kendall's tau between the true and estimated degree of influence.

	LFICM ($K = 1$)	LFICM ($K = 3$)	LFICM ($K = 5$)	LFICM ($K = 7$)	LFICM ($K = 9$)	SaitoICM	SimpleICM
Blog	0.424	0.534	**0.599**	0.585	0.553	0.532	0.591
Enron	0.293	**0.379**	0.371	0.370	0.370	0.372	0.369
Mixi	0.517	0.443	0.441	0.445	**0.445**	0.252	0.441

Note: numbers in bold indicate the best method for each set of data.

probabilities are used only once. Estimating each diffusion probability itself with high generalization performance or without outliers leads to a good result. On the other hand, this experiment might evaluate the diffusion probability of a link many times because the information passes through nodes with a lot of links, i.e, *authorities*, many times in our simulations. This fact makes it especially important in this experiment to learn the probabilities of links extending from the authority nodes to estimate the degrees of influence of all the nodes. The proposed model is useful in complex situations such as that represented by this experiment.

4. Conclusion and Future Work

In this paper, we proposed the Latent Feature Independent Cascade Model (LFICM) for modeling information diffusion phenomena and the predicting future trends to estimate the diffusion probabilities of each link. In particular, we newly incorporated two latent features for each node, which represent the sensitivity to incoming information and the power of influence. We then assumed that each diffusion probability is calculated from the features of both terminating nodes. To estimate the parameters of the LFICM from observations we formulated the posterior probability of the model and derived the parameter estimation method based on the EM algorithm. In the experiments, the proposed model outperformed the conventional estimation methods with respect to the precision of the diffusion probability estimation for four kinds of dataset. Moreover, we showed that estimating the parameters of the ICM with the proposed model allows us to understand precisely the degrees of influence of nodes.

In realistic settings, the influence power and the sensitivity of each individual vary widely and these variations are unknown. To consider this fact, we will attempt to extend the proposed model to a Bayesian model using proper prior distributions so as to estimate standard deviations σ_X and σ_Y along with the other parameters. For K estimation, we can use the cross-validation method and model selection methods such as AIC and BIC.

References

[1] A. S. S. Reddy, A. Siva, P. Kasat, and A. Jain, "Box-Office Opening Prediction of Movies based on Hype Analysis through Data Mining," *International Journal of Computer Applications*, vol. 56, no. 1, pp. 1–5, 2012.

[2] X. Zhang, H. Fuehres, and P. A. Gloor, "Predicting Stock Market Indicators Through Twitter " I hope It Is Not as Bad as I Fear "," *The 2nd Collaborative Innovation Networks Conference*, vol. 26, pp. 55–62, 2011.

[3] F. Franch, "2010 UK Election Prediction with Social Media," *Journal of Information Technology & Politics*, vol. 10, no. 1, pp. 57–71, Jan. 2013.

[4] J. Iribarren and E. Moro, "Impact of Human Activity Patterns on the Dynamics of Information Diffusion," *Physical Review Letters*, vol. 103, no. 3, pp. 8–11, July 2009.

[5] W. Galuba and K. Aberer, "Outtweeting the Twitterers-Predicting Information Cascades in Microblogs," *WOSN'10 Proceedings of the 3rd Conference on Online Social Networks*, 2010.

[6] T.-T. Kuo, S.-C. Hung, W.-S. Lin, S.-D. Lin, T.-C. Peng, and C.-C. Shih, "Assessing the Quality of Diffusion Models Using Real-World Social Network Data," *2011 International Conference on Technologies and Applications of Artificial Intelligence*, pp. 200–205, Nov. 2011.

[7] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the Spread of Influence through a Social Network," *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, p. 137, 2003.

[8] V. Sood and S. Redner, "Voter Model on Heterogeneous Graphs," *Physical Review Letters*, vol. 94, no. 17, pp. 178 701–, May 2005.

[9] D. Trpevski and L. Kocarev, "Model for Rumor Spreading over Networks," *Physical Review E*, vol. 81, no. 5, pp. 1–11, May 2010.

[10] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the Spread of Misinformation in Social Networks," in *Proceedings of the 20th International Conference on World Wide Web*. New York, New York, USA: ACM Press, Mar. 2011, p. 665.

[11] W. Chen, A. Collins, and R. Cummings, "Influence Maximization in Social Networks When Negative Opinions May Emerge and Propagate," *SIAM International Conference on Data Mining*, 2011.

[12] M. Kimura, K. Saito, and H. Motoda, "Blocking Links to Minimize Contamination Spread in a Social Network," *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no. 2, pp. 1–23, Apr. 2009.

[13] D. Kempe, J. Kleinberg, and E. Tardos, "Influential Nodes in a Diffusion Model for Social Networks," *Automata, Languages and Programming*, vol. 3580, pp. 1127–1138, 2005.

[14] M. Kimura, K. Saito, and R. Nakano, "Extracting Influential Nodes for Information Diffusion on a Social Network," *Proceedings of the National Conference on Artificial Intelligence*, vol. 22, no. 2, p. 1371, 2007.

[15] Y. Yoshikawa, K. Saito, H. Motoda, K. Ohara, and M. Kimura, "Acquiring Expected Influence Curve from Single Diffusion Sequence," *Knowledge Management and Acquisition for Smart Systems and Services*, pp. 273–287, 2010.

[16] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins, "Information Diffusion through Blogspace," *Proceedings of the 13th International Conference on World Wide Web*, pp. 491–501, 2004.

[17] K. Saito, M. Kimura, K. Ohara, and H. Motoda, "Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis," *First Asian Conference on Machine Learning*, pp. 322–337, 2009.

[18] K. Saito, K. Ohara, Y. Yamagishi, and M. Kimura, "Learning Diffusion Probability Based on Node Attributes in Social Networks," *19th International Symposium, ISMIS 2011*, pp. 153–162, 2011.

[19] L. Dickens, I. Molloy, and J. Lobo, "Learning Stochastic Models of Information Flow," *2012 IEEE 28th International Conference on Data Engineering (ICDE)*, 2012.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977.