# Effective Visualization of Information Diffusion Process over Complex Networks

Kazumi Saito[1], Masahiro Kimura[2], and Hiroshi Motoda[3]

[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 Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

**Abstract.** Effective visualization is vital for understanding a complex network, in particular its dynamical aspect such as information diffusion process. Existing node embedding methods are all based solely on the network topology and sometimes produce counter-intuitive visualization. A new node embedding method based on conditional probability is proposed that explicitly addresses diffusion process using either the IC or LT models as a cross-entropy minimization problem, together with two label assignment strategies that can be simultaneously adopted. Numerical experiments were performed on two large real networks, one represented by a directed graph and the other by an undirected graph. The results clearly demonstrate the advantage of the proposed methods over conventional spring model and topology-based cross-entropy methods, especially for the case of directed networks.

## 1 Introduction

Analysis of the structure and function of complex networks, such as social, computer and biochemical networks, has been a hot research subject with considerable attention [10]. A network can play an important role as a medium for the spread of various information. For example, innovation, hot topics and even malicious rumors can propagate through social networks among individuals, and computer viruses can diffuse through email networks. Previous work addressed the problem of tracking the propagation patterns of topics through network spaces [5, 1], and studied effective "vaccination" strategies for preventing the spread of computer viruses through networks [11, 2]. Widely-used fundamental probabilistic models of information diffusion through networks are the *independent cascade (IC) model* and the *linear threshold (LT) model* [8, 5]. Researchers have recently investigated the problem of finding a limited number of influential nodes that are effective for the spread of information through a network under these models [8, 9]. In these studies, understanding the flow of information through networks is an important research issue.

This paper focuses on the problem of visualizing the information diffusion process, which is vital for understanding its characteristic over a complex network. Existing node embedding methods such as spring model method [7] and cross entropy method [14] are solely based on the network topology. They do not take account how information diffuses across the network. Thus, it often happens that the visualized information flow do not match our intuitive understanding, *e.g.*, abrupt information flow gaps, inconsistency between the nodes distance and the reachability of information, irregular pattern of information spread, etc. This sometimes happens when visualizing the diffusion process for a network represented by a directed graph.

Thus, it is important that node embedding explicitly reflects the diffusion process to produce more natural visualization. We have devised a new node embedding method that incorporates conditional probability of information diffusion between two nodes, a target source node where the information is initially issued and a non-target influenced node where the information has been received via intermediate nodes. Our postulation is that good visualization should satisfy the two conditions: path continuity, *i.e.* any information diffusion path is continuous and path separability, *i.e.* each different information diffusion path is clearly separated from each other. To this end, the above node embedding is coupled with two label assignment strategies, one with emphasis on influence of initially activated nodes, and the other on degree of information reachability.

Extensive numerical experiments were performed on two large real networks, one generated from a large connected trackback network of blog data, resulting in a directed graph of $12,047$ nodes and $53,315$ links, and the other, a network of people, generated from a list of people within a Japanese Wikipedia, resulting in an undirected graph of $9,481$ nodes and $245,044$ links. The results clearly indicate that the proposed probabilistic visualization method satisfies the above two conditions and demonstrate its advantage over the well-known conventional methods: spring model and topology-based cross-entropy methods, especially for the case of a directed network. The method appeals well to our intuitive understanding of information diffusion process.

## 2 Information Diffusion Models

We mathematically model the spread of information through a directed network $G = (V, E)$ under the IC or LT model, where $V$ and $E$ ($\subset V \times V$) stands for the sets of all the nodes and links, respectively. We call nodes *active* if they have been influenced with the information. In these models, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set $S$ of active nodes, we assume that the nodes in $S$ have first become active at time-step 0, and all the other nodes are inactive at time-step 0.

### 2.1 Independent Cascade Model

We define the IC model. In this model, for each directed link $(u, v)$, we specify a real value $\beta_{u,v}$ with $0 < \beta_{u,v} < 1$ in advance. Here $\beta_{u,v}$ is referred to as the *propagation probability* through link $(u, v)$. The diffusion process proceeds from a given initial active

set $S$ in the following way. When a node $u$ first becomes active at time-step $t$, it is given a single chance to activate each currently inactive child node $v$, and succeeds with probability $\beta_{u,v}$. If $u$ succeeds, then $v$ will become active at time-step $t+1$. If multiple parent nodes of $v$ first 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.

For an initial active set $S$, let $\varphi(S)$ denote the number of active nodes at the end of the random process for the IC model. Note that $\varphi(S)$ is a random variable. Let $\sigma(S)$ denote the expected value of $\varphi(S)$. We call $\sigma(S)$ the *influence degree* of $S$.

### 2.2 Linear Threshold Model

We define the LT model. In this model, for every node $v \in V$, we specify, in advance, a *weight* $\omega_{u,v}$ ($> 0$) from its parent node $u$ such that

$$\sum_{u \in \Gamma(v)} \omega_{u,v} \leq 1,$$

where $\Gamma(v) = \{u \in V; (u,v) \in E\}$. The diffusion process from a given initial active set $S$ proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* $\theta_v$ is chosen uniformly at random from the interval $[0,1]$. At time-step $t$, an inactive node $v$ is influenced by each of its active parent nodes, $u$, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of $v$ is at least threshold $\theta_v$, that is,

$$\sum_{u \in \Gamma_t(v)} \omega_{u,v} \geq \theta_v,$$

then $v$ will become active at time-step $t+1$. Here, $\Gamma_t(v)$ stands for the set of all the parent nodes of $v$ that are active at time-step $t$. The process terminates if no more activations are possible.

The LT model is also a probabilistic model associated with the uniform distribution on $[0,1]^{|V|}$. Similarly to the IC model, we define a random variable $\varphi(S)$ and its expected value $\sigma(S)$ for the LT model.

### 2.3 Influence Maximization Problem

Let $K$ be a given positive integer with $K < |V|$. We consider the problem of finding a set of $K$ nodes to target for initial activation such that it yields the largest expected spread of information through network $G$ under the IC or LT model. The problem is referred to as the *influence maximization problem*, and mathematically defined as follows: Find a subset $S^*$ of $V$ with $|S^*| = K$ such that $\sigma(S^*) \geq \sigma(S)$ for every $S \subset V$ with $|S| = K$.

For a large network, any straightforward method for exactly solving the influence maximization problem suffers from combinatorial explosion. Therefore, we approximately solve this problem. Here, $U_K = \{u_1, \cdots, u_K\}$ is the set of $K$ nodes to target for initial activation, and represents the approximate solution obtained by this algorithm. We refer to $U_K$ as the *greedy solution*.

Using large collaboration networks, Kempe et al. [8] experimentally demonstrated that the greedy algorithm significantly outperforms node-selection heuristics that rely on the well-studied notions of degree centrality and distance centrality in the sociology literature. Moreover, the quality of $U_K$ is guaranteed:

$$\sigma(U_K) \geq \left(1 - \frac{1}{e}\right) \sigma(S_K^*),$$

where $S_K^*$ stands for the exact solution to this problem.

To implement the greedy algorithm, we need a method for calculating $\{\sigma(U_k \cup \{v\}); v \in V \setminus U_k\}$ for $1 \leq k \leq K$. However, it is an open question to exactly calculate influence degrees by an efficient method for the IC or LT model [8]. Kimura et al. [9] presented the bond percolation method that efficiently estimates influence degrees $\{\sigma(U_k \cup \{v\}); v \in V \setminus U_k\}$. Therefore, we estimate the greedy solution $U_K$ using their method.

## 3 Visualization Method

We especially focus on visualizing the information diffusion process from the target nodes selected to be a solution of the influence maximization problem. To this end, we propose a visualization method that has the following characteristics: 1) utilizing the target nodes as a set of pivot objects for visualization, 2) applying a probabilistic algorithm for embedding all the nodes in the networks into an Euclidean space, and 3) varying appearance of the embedded nodes on the basis of two label assignment strategies. In what follows, we describe some details of the probabilistic embedding algorithm and the label assignment strategies.

### 3.1 Probabilistic Embedding Algorithm

Let $U_K = \{u_k : 1 \leq k \leq K\} \subset V$ be a set of target nodes, which maximizes an expected number of influenced nodes in the network based on an information diffusion model such as IC or LT. Let $v_n \notin U_K$ be a non-target node in the network, then we can consider the conditional probability $p_{k,n} = p(v_n|u_k)$ that a node $v_n$ is influenced when one target node $u_k$ alone is set to an initial information source. Here note that we can regard $p_{k,n}$ as a binomial probability with respect to a pair of nodes $u_k$ and $v_n$. In our visualization approach, we attempt to produce embedding of the nodes so as to preserve the relationships expressed as the conditional probabilities for all pairs of target and non-target nodes in the network. We refer to this visualization strategy as the *conditional probability embedding (CE) algorithm*.

**Objective Function** Let $\{\mathbf{x}_k : 1 \leq k \leq K\}$ and $\{\mathbf{y}_n : 1 \leq n \leq N\}$ be the embedding positions of the corresponding $K$ target nodes and $N = |V| - K$ non-target nodes in an $M$ dimensional Euclidean space. Hereafter, the $\mathbf{x}_k$ and $\mathbf{y}_n$ are called target and non-target vectors, respectively. As usual, we define the Euclidean distance between $\mathbf{x}_k$ and $\mathbf{y}_n$ as follows:

$$d_{k,n} = \|\mathbf{x}_k - \mathbf{y}_n\|^2 = \sum_{m=1}^{M} (x_{k,m} - y_{n,m})^2.$$

Here, we introduce a monotonic decreasing function $\rho(s) \in [0, 1]$ with respect to $s \geq 0$, where $\rho(0) = 1$ and $\rho(\infty) = 0$.

Since $\rho(d_{k,n})$ can also be regarded as a binomial probability with respect to $\mathbf{x}_k$ and $\mathbf{y}_n$, we can introduce a cross-entropy (cost) function between $p_{k,n}$ and $\rho(d_{k,n})$ as follows:

$$\mathcal{E}_{k,n} = -p_{k,n} \ln \rho(d_{k,n}) - (1 - p_{k,n}) \ln(1 - \rho(d_{k,n})).$$

Since $\mathcal{E}_{k,n}$ is minimized when $\rho(d_{k,n}) = p_{k,n}$, this minimization with respect to $\mathbf{x}_k$ and $\mathbf{y}_n$ is consistent with our problem setting. In this paper, we employ a function of the form

$$\rho(s) = \exp\left(-\frac{s}{2}\right)$$

as the monotonic decreasing function, but note that our approach is not restricted to this form. Then, the total cost function (objective function) can be defined as follows:

$$\mathcal{E} = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} p_{k,n} d_{k,n} - \sum_{n=1}^{N} \sum_{k=1}^{K} (1 - p_{k,n}) \ln(1 - \rho(d_{k,n})). \tag{1}$$

Namely, our approach is formalized as a minimization problem of the objective function defined in (1) with respect to $\{\mathbf{x}_k : 1 \leq k \leq K\}$ and $\{\mathbf{y}_n : 1 \leq n \leq N\}$.

**Learning Algorithm** As the basic structure of our learning algorithms, we adopt a coordinate strategy just like the *EM* (Expectation-Maximization) algorithm. First, we adjust the target vectors, so as to minimize the objective function by freezing the non-target vectors, and then, we adjust the non-target vectors by freezing the target vectors. These two steps are repeated until convergence is obtained.

In the former minimization step for the *CE* algorithm, we need to calculate the derivative of the objective function with respect to $\mathbf{x}_n$ as follows:

$$\mathcal{E}_{\mathbf{x}_k} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_k} = \sum_{n=1}^{N} \frac{p_{k,n} - \rho(d_{k,n})}{1 - \rho(d_{k,n})} (\mathbf{x}_k - \mathbf{y}_n). \tag{2}$$

Since $\mathbf{x}_{k'}$ $(k' \neq k)$ disappears in (2), we can update $\mathbf{x}_k$ without considering the other target vectors. In the latter minimization step for the *CE* algorithm, we need to calculate the following derivative,

$$\mathcal{E}_{\mathbf{y}_n} = \frac{\partial \mathcal{E}}{\partial \mathbf{y}_n} = \sum_{k=1}^{K} \frac{p_{k,n} - \rho(d_{k,n})}{1 - \rho(d_{k,n})} (\mathbf{y}_n - \mathbf{x}_k).$$

In this case, we update $\mathbf{y}_n$ by freezing the other non-target vectors. Overall, our algorithm can be summarized as follows:

**1.** Initialize vectors $\mathbf{x}_1, \cdots, \mathbf{x}_K$ and $\mathbf{y}_1, \cdots, \mathbf{y}_N$.
**2.** Calculate gradient vectors $\mathcal{E}_{\mathbf{x}_1}, \cdots, \mathcal{E}_{\mathbf{x}_K}$.
**3.** Update target vectors $\mathbf{x}_1, \cdots, \mathbf{x}_K$.
**4.** Calculate gradient vectors $\mathcal{E}_{\mathbf{y}_1}, \cdots, \mathcal{E}_{\mathbf{y}_N}$.

**5.** Update non-target vectors $\mathbf{y}_1, \cdots, \mathbf{y}_N$.
**6.** Stop if $\max_{k,n}\{\|\mathcal{E}_{\mathbf{x}_k}\|, \|\mathcal{E}_{\mathbf{y}_n}\|\} < \epsilon$.
**7.** Return to **2**.

Here, a small positive value $\epsilon$ controls the termination condition.

### 3.2 Label Assignment Strategies

In an attempt to effectively understand information diffusion process, we propose two label assignment strategies, on which the appearance of the embedded target and non-target nodes depends. The first strategy assigns labels to non-target nodes according to the standard Bayes decision rule.

$$l_1(v_n) = \arg \max_{1 \leq k \leq K} \{p_{k,n}\}$$

It is obvious that this decision naturally reflects influence of the target nodes. Note that the target node identification number $k$ corresponds to the order determined by the greedy method, *i.e.*, $l_1(u_k) = k$.

In the second strategy, we introduce the following probability quantization by noting $0 \leq \max_{1 \leq k \leq K}\{p_{k,n}\} \leq 1$,

$$l_2(v_n) = \left\lfloor -\log_b \max_{1 \leq k \leq K} \{p_{k,n}\} \right\rfloor + 1,$$

where $\lfloor x \rfloor$ returns the greatest integer not greater than $x$, and $b$ stands for the base of logarithm. To each node belonging to $Z = \{v_n : \max_{1 \leq k \leq K}\{p_{k,n}\} = 0\}$, we assign as the label the maximum number determined by the nodes not belonging to $Z$. We believe that this quantization reasonably reflects the degree of information reachability. Here note that $l_2(u_k) = 1$ because it always becomes active at time step $t = 0$. These labels are further mapped to colors scales according to some monotonic mapping functions.

## 4 Experimental Evaluation

### 4.1 Network Data

In our experiments, we employed two sets of real networks used in [9], which exhibit many of the key features of social networks. We describe the details of these network data.

The first one is a trackback network of blogs. Blogs are personal on-line diaries managed by easy-to-use software packages, and have rapidly spread through the World Wide Web [5]. Bloggers (*i.e.*, blog authors) discuss various topics by using trackbacks. Thus, a piece of information can propagate from one blogger to another blogger through a trackback. We exploited the blog "Theme salon of blogs" in the site "goo" [2], where a blogger can recruit trackbacks of other bloggers by registering an interesting theme. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR

[2] `http://blog.goo.ne.jp/usertheme/`

Fukuchiyama Line Derailment Collision", we collected a large connected trackback network in May, 2005. The resulting network had 12,047 nodes and 53,315 directed links, which features the so-called "power-law" distributions for the out-degree and in-degree that most real large networks exhibit. We refer to this network data as the blog network.

The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages. The undirected graph is represented by an equivalent directed graph by regarding undirected links as bidirectional ones[3]. The resulting network had 9,481 nodes and 245,044 directed links. We refer to this network data as the Wikipedia network.

Newman and Park [12] observed that social networks represented as undirected graphs generally have the following two statistical properties that are different from non-social networks. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* $C$ than the corresponding *configuration models* (*i.e.*, random network models). For the undirected graph of the Wikipedia network, the value of $C$ of the corresponding configuration model was 0.046, while the actual measured value of $C$ was 0.39, and the degrees of adjacent nodes were positively correlated. Therefore, the Wikipedia network has the key features of social networks.

### 4.2 Experimental Settings

In the IC model, we assigned a uniform probability $\beta$ to the propagation probability $\beta_{u,v}$ for any directed link $(u, v)$ of a network, that is, $\beta_{u,v} = \beta$. We, first, determine the typical value of $\beta$ for the blog network, and use it in the experiments. It is known that the IC model is equivalent to the bond percolation process that independently declares every link of the network to be "occupied" with probability $\beta$ [10]. Let $J$ denote the expected fraction of the maximal strongly connected component (SCC) in the network constructed by the occupied links. Note that $J$ is an increasing function of $\beta$. We focus on the point $\beta_*$ at which the average rate of change of $J$, $dJ/d\beta$, attains the maximum, and regard it as the typical value of $\beta$ for the network. Note that $\beta_*$ is a critical point of $dJ/d\beta$, and defines one of the features intrinsic to the network. Figure 1 plots $J$ as a function of $\beta$. Here, we estimated $J$ using the bond percolation method with the same parameter value as below [9]. From this figure we experimentally estimated $\beta_*$ to be 0.2 for the blog network. In the same way, we experimentally estimated $\beta_*$ to be 0.05 for the Wikipedia network.

In the LT model, we uniformly set weights as follows. For any node $v$ of a network, the weight $\omega_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by $\omega_{u,v} = 1/|\Gamma(v)|$.

Once these parameters were set, we estimated the greedy solution $U_K = \{u_1, \cdots, u_K\}$ of targets and the conditional probabilities $\{p_{k,n};\ 1 \le k \le K,\ 1 \le n \le N\}$ using the bond percolation method with the parameter value 10,000 [9]. Here, the parameter represents

[3] For simplicity, we call a graph with bi-directional links an undirected graph

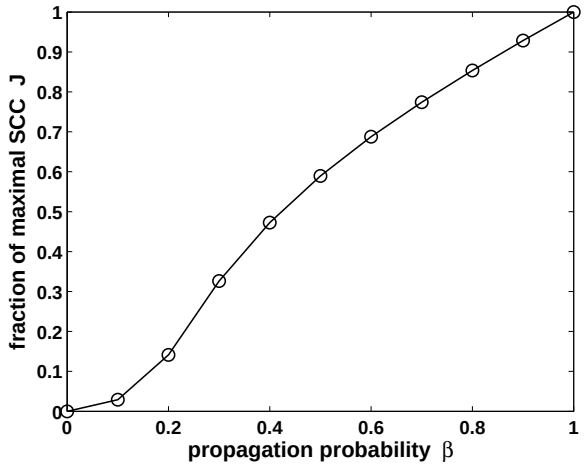



Fig. 1: The fraction $J$ of the maximal SCC as a function of the propagation probability $\beta$

the number of bond percolation processes for estimating the influence degree $\sigma(S)$ of a given initial active set $S$.

### 4.3 Brief Description of Other Visualization Methods used for Comparison

We have compared the proposed method with the two well known methods: spring model method [7] and standard cross-entropy method [14].

Spring model method assumes that there is a hypothetical spring between each connected node pair and locates nodes such that the distance of each node pair is closest to its minimum path length at equilibrium. Mathematically it is formulated as minimizing (3).

$$\mathcal{K}(\mathbf{x}) = \sum_{u=1}^{|V|-1} \sum_{v=u+1}^{|V|} \alpha_{u,v} (g_{u,v} - \|\mathbf{x}_u - \mathbf{x}_v\|)^2, \tag{3}$$

where $g_{u,v}$ is the minimum path length between node $u$ and node $v$, and $\alpha_{u,v}$ is a spring constant which is normally set to $1/(2g_{u,v}^2)$. Standard cross-entropy method first defines a similarity $\rho(\|\mathbf{x}_u - \mathbf{x}_v\|^2) = \exp(-\|\mathbf{x}_u - \mathbf{x}_v\|^2/2)$ between the embedding coordinates $x_u$ and $x_v$ and uses the corresponding element $a_{u,v}$ of the adjacency matrix as a measure of distance between the node pair, and tries to minimize the total cross entropy between these two. Mathematically it is formulated as minimizing (4).

$$C(\mathbf{x}) = \sum_{u=1}^{|V|-1} \sum_{v=u+1}^{|V|} \left\{ -a_{u,v} \log \rho(\|\mathbf{x}_u - \mathbf{x}_v\|^2) - (1 - a_{u,v}) \log(1 - \rho(\|\mathbf{x}_u - \mathbf{x}_v\|^2)) \right\}, \tag{4}$$

Here, note that we used the same function $\rho$ as before.

As is clear from the above formulation, both methods are completely based on graph topology. They are both non-linear optimization problem and easily solved by a standard coordinate descent method. Here note that the applicability of the spring model method and cross-entropy method is basically limitted to undirected networks. Thus, in order to obtain the embedding results by using these methods we neglected the direction in the directed blog network and regarded it as undirected one.

### 4.4 Experimental Results

Two label assignment strategies are independent to each other. They can be used either separately or simultaneously. Here, we used a color mapping to both, and thus, use them separately. The visualization results are shown in four figures, each with six network figures. In each of these four figures, the left three show the results of the first visualization strategy (method 1) and the right three the results of the second visualization strategy (method 2), and the top two show the results of the proposed method (*CE* algorithm), the middle two the results of spring model and the bottom two the results of the topology-based cross entropy method. The first two figures (Figs. 2 and 3) corresponds to the results of blog network and the last two (Figs. 4 and 5) the results of Wikipedia network. For each, the results of the IC model comes first, followed by the results of the LT model.

The most influential top ten nodes are chosen as the target nodes, and the rest are all non-target nodes. In the first visualization strategy, the color of a non-target node indicates which target node is most influential to the node, whereas in the second visualization strategy, it indicates how easily the information diffuses from the most influential target node to reach the node. Note that a non-target node is influenced by multiple target nodes probabilistically, but here the target with the highest conditional probability is chosen. Thus, the most influential target node is determined for each non-target node.

Observation of the results of the proposed method (Figs. 2a, 2b, 3a, 3b, 4a, 4b, 5a, and 5b) indicates that the proposed method has the following desirable characteristics: 1) the target nodes tend to be allocated separately from each other, and from each target node, 2) the non-target nodes that are most affected by the same target node are laid out forming a band and 3) the reachability changes continuously from the highest at the target node to the lowest at the other end of the band. From this observation, it is confirmed that the two conditions we postulated are satisfied for the both diffusion models. Observation 2) above, however, needs further clarification. Note that our visualization does not necessarily cause the information diffusion to neighboring nodes to be in the form of a line in the embedded space. For example, if there is only one source (K=1), the information would diffuse concentrically. A node in general receives information from multiple sources. The fact that the embedding result forms a line, on the contrary, reveals an important characteristic that little information is coming from the other sources for the networks we analyzed.

In the proposed method, non-target nodes that are readily influenced are easily identified, whereas those that are rarely influenced are placed together. Overlapping of the color well explains the relationship between each target and a non-target node. For example, in Figures 3a and 3b it is easily observed that the effect of the target nodes

5, 2 on non-target nodes interferes with the three bands that are spread from the target nodes 8, 3, 10, and non-target nodes overlap as they move away from the target nodes, demonstrating that a simple two-dimensional visualization facilitates how different node groups overlap and how the information flows from different target nodes interfere each other. The same observation applies for the target nodes 6, 1, 9, 7. On the contrary, the target node 4 has its own effect separately. A similar argument is possible for relationship within target nodes. For example, in Figures 2a target nodes 4, 5, 6, 8 are located in relatively near positions compared with the other target nodes. It is crucial to abstract and visualize the essence of information diffusion by deleting the unnecessary details (node to node diffusion). A good explanation for the overlap like the above is not possible by other visualization methods. Further, the visualization results of both IC and LT models are spatially well balanced. In addition, there are no significant differences on the results of visualization between the directed network and undirected network. Both are equally good.

Observation of the results of the spring model (Figs. 2c, 2d, 3c, 3d, 4c, 4d, 5c, and 5d) and the topology-based cross entropy method (Figs. 2e, 2f, 3e, 3f, 4e, 4f, 5e, and 5f) reveals the followings. The clear difference of these from the proposed method is that it is not that easy to locate the target nodes. This is true, in particular, for the spring model. It is slightly easier for the standard cross-entropy method because the target nodes are placed in the cluster centers, but clusters often overlap, which makes visualization less understandable. It is also noted that those nodes with high reachability, *i.e.*, nodes with red, which should be placed separately due to the influence of different target nodes are placed in mixture. Further, unlike the proposed method, there is clear difference between the IC model and the LT model. In the IC model, we can easily recognize non-target nodes with high reachability, which cover a large portion of the network, whereas in the LT model, such nodes covering only a small portion are almost invisible in the network. In contrast, we can easily pick up such non-target nodes with high reachability even for the LT model in the proposed method.

We observe that the standard cross-entropy method is in general better than the spring model method in terms of the clarity of separability. The standard cross-entropy method does better for the IC model than for the LT model, and is comparable to the proposed method in terms of the clarity of reachability. However, the results of the standard cross-entropy method (e.g., Fig. 2f) are unintuitive, where the high reachability non-target nodes are placed away from the target nodes, and some target node forms several isolated clusters. We believe that this point is an intrinsic limitation of the standard cross-entropy method.

The concept of our visualization is based on the notion that how the information diffuses should primarily determine how the visualization is made, irrespective of the graph topology. We observe that the visualization which is based solely on the topology has intrinsic limitation when we deal with a huge network from the point of both computational complexity (*e.g.*, the spring model does not work for a network with millions nodes) and understandability. Overall, we can conclude that the proposed method provides better visualization which is more intuitive and easily understandable.

## 5 Related Work and Discussion

As defined earlier, let $K$ and $N$ be the numbers of target and non-target nodes in a network. Then the computational complexity of our embedding method amounts to $O(NK)$, where we assume the number of learning iterations and the embedding dimension to be constants. This reduced complexity greatly expands the applicability of our method over the other representative network embedding methods, *e.g.*, the spring model method [7] and the standard cross-entropy method [14], both of which require the computational complexity of $O(N^2)$ under the setting that $K \ll N$.

In view of computational complexity, our visualization method is closely related to those conventional methods, such as FastMap or Landmark Multidimensional Scaling (LMDS), which are based on the Nyström approximation [13]. Typically, these methods randomly select a set of pivot (or landmark) objects, then produce the embedding results so as to preserve relationships between all pairs of pivot and non-pivot objects. In contrast, our method selects target (pivot) nodes based on the information diffusion models.

Our method adopts the basic idea of the probabilistic embedding algorithms including Parametric Embedding (PE) [6] and Neural Gas Cross-Entropy (NG-CE) [4]. The PE method attempts to uncover classification structures by use of posterior probabilities, while the NG-CE method is restricted to visualize the codebooks of the neural gas model. Our purpose, on the other hand, is to effectively visualize information diffusion process. The two visualization strategies we proposed match this aim.

We are not the first to try to visualize the information diffusion process. Adar and Adamic [1] presented a visualization system that tracks the flow of URL through blogs. However, same as above, their visualization method did not incorporate an information diffusion model. Further, they laid out only a small number of nodes in a tree structure, and it is unlikely that their approach scales up to a large network.

Finally we should emphasize that unlike most representative embedding methods for networks [3], our visualization method is applicable to large-scale directed graphs while incorporating the effect of information diffusion models. In this paper, however, we also performed our experiments using the undirected (bi-directional) Wikipedia network. This is because we wanted to include favorable evaluation for the comparison methods. As noted earlier, we cannot directly apply the conventional embedding methods to directed graphs without some topology modification such as link addition or deletion.

## 6 Conclusion

We proposed an innovative probabilistic visualization method to help understand complex network. The node embedding scheme in the method, formulated as a model-based cross-entropy minimization problem, explicitly take account of information diffusion process, and therfore, the resulting visualization is more intuitive and easier to understand than the state-of-art approaches such as the spring model method and the standard cross-entropy method. Our method is efficient enough to be applied to large networks. The experiments performed on a large blog network (directed) and a large Wikipedia

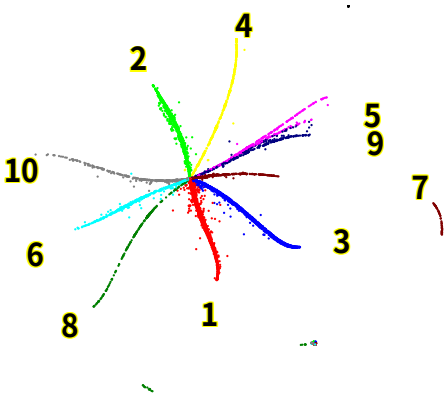
network (undirected) clearly demonstrate the advantage of the proposed method. The proposed method is confirmed to satisfy both path continuity and path separability conditions which are the important requirement for the visualization to be understandable. Our future work includes the extension of the proposed approach to the visualization of growing networks.

## Acknowledgments

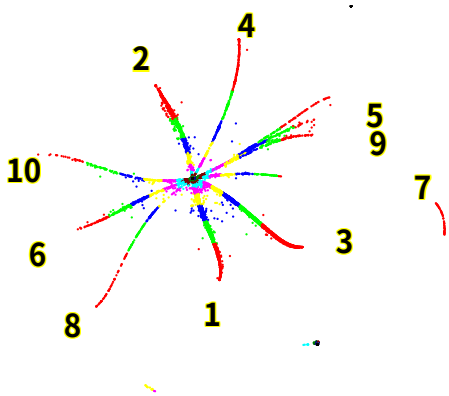
This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

## References

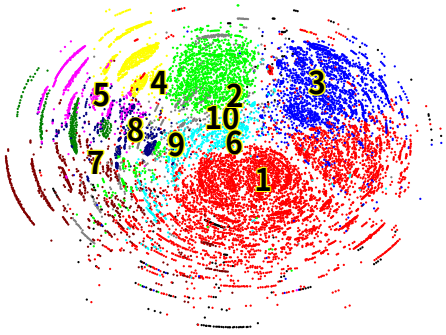
1. Adar, E., & Adamic, L. (2005). Tracking information epidemics in blogspace. *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 207–214).
2. Balthrop, J., Forrest, S., Newman, M. E. J., & Williampson, M. W. (2004). Technological networks and the spread of computer viruses. *Science*, *304*, 527–529.
3. Battista, G., Eades, P., Tamassia, R., & Tollis, I. (1999). *Graph drawing: An annotated bibliography*. Prentice-Hall, New Jersy.
4. Estévez, P. A., Figueroa, C. J., & Saito, K. (2005). Cross-entropy embedding of high-dimensional data using the neural gas model. *Neural Networks*, *18*, 727–737.
5. Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004). Information diffusion through blogspace. *Proceedings of the 13th International World Wide Web Conference* (pp. 107–117).
6. Iwata, T., Saito, K., Ueda, N., Stromsten, S., Griffiths, T. L., & Tenenbaum, J. B. (2007). Parametric embedding for class visualization. *Neural Computation*, *19*, 2536–2556.
7. Kamada, K., & Kawai, S. (1989). An algorithm for drawing general undirected graph. *Information Processing Letters*, *31*, 7–15.
8. Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 137–146).
9. Kimura, M., Saito, K., & Nakano, R. (2007). Extracting influential nodes for information diffusion on a social network. *Proceedings of the 22nd AAAI Conference on Artificial Intelligence* (pp. 1371–1376).
10. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*, 167–256.
11. Newman, M. E. J., Forrest, S., & Balthrop, J. (2002). Email networks and the spread of computer viruses. *Physical Review E*, *66*, 035101.
12. Newman, M. E. J. & Park, J. (2003). Why social networks are different from other types of networks. *Physical Review E*, *68*, 036122.
13. Platt, J. C. (2005). Fastmap, metricmap, and landmark mds are all nyström algorithms. *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 261–268).
14. Yamada, T., Saito, K., & Ueda, N. (2003). Cross-entropy directed embedding of network data. *Proceedings of the 20th International Conference on Machine Learning* (pp. 832–839).



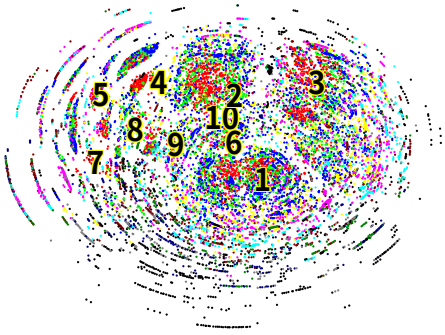

(a) Proposed method 1



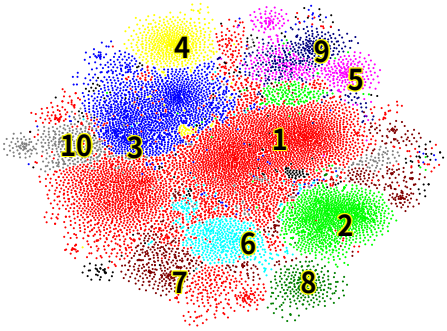

(b) Proposed method 2



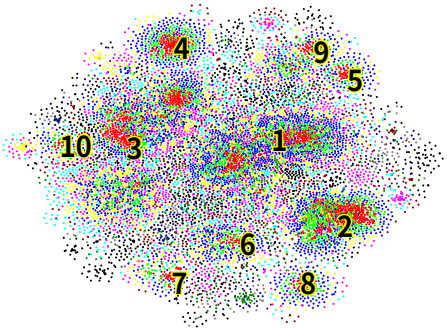

(c) Spring model method 1




(d) Spring model method 2




(e) Toplogy-based cross-entropy method 1

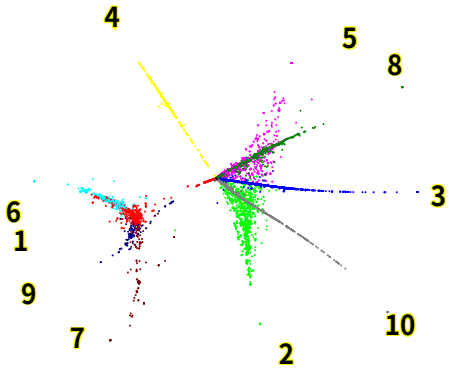



(f) Toplogy-based cross-entropy method 2

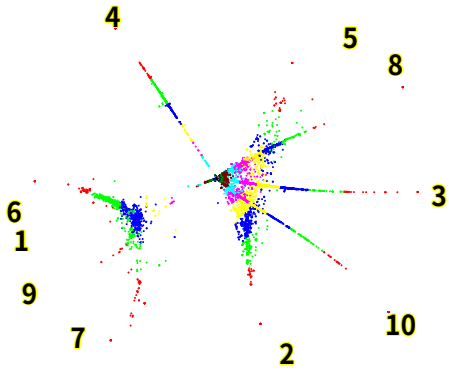

(g) Class-label assignment


(h) Color-map assignment

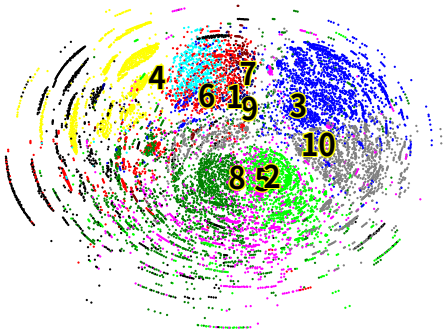
Fig. 2: Visualization of IC model for blog network



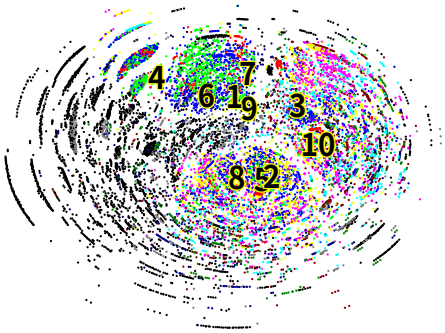

(a) Proposed method 1



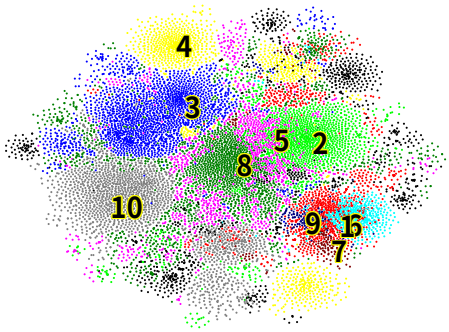

(b) Proposed method 2



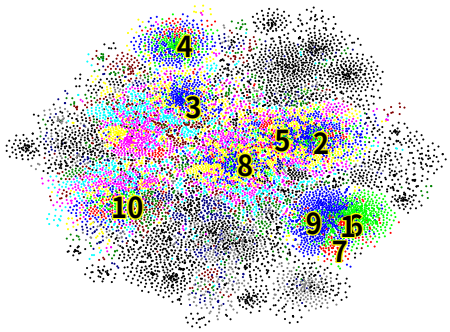

(c) Spring model method 1




(d) Spring model method 2




(e) Toplogy-based cross-entropy method 1

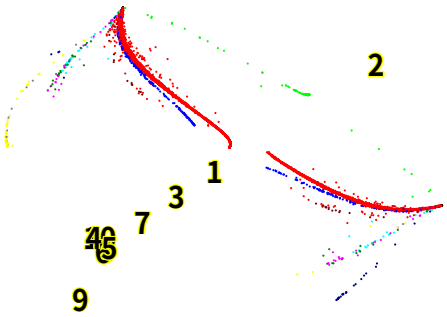



(f) Toplogy-based cross-entropy method 2

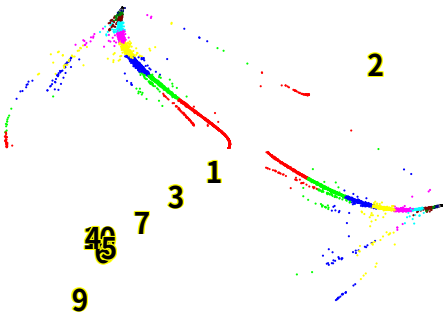

(g) Class-label assignment


(h) Color-map assignment

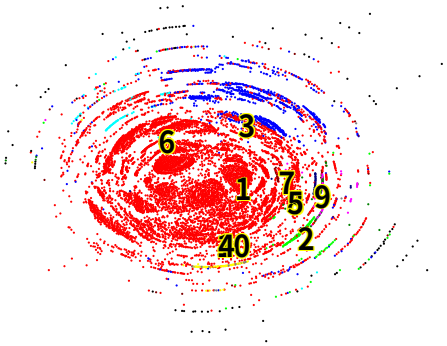
Fig. 3: Visualization of LT model for blog network



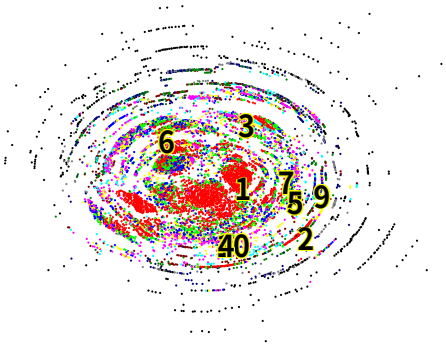

(a) Proposed method 1



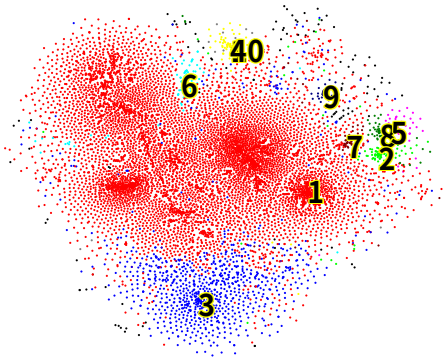

(b) Proposed method 2



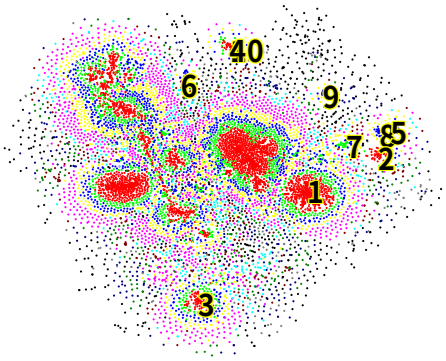

(c) Spring model method 1




(d) Spring model method 2




(e) Toplogy-based cross-entropy method 1

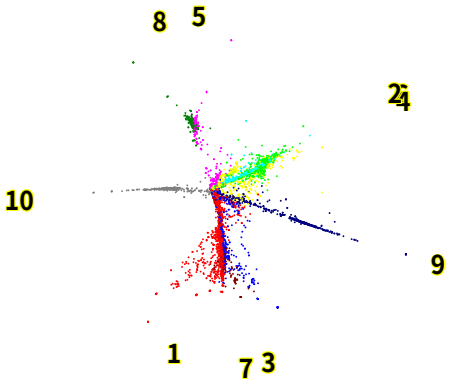



(f) Toplogy-based cross-entropy method 2

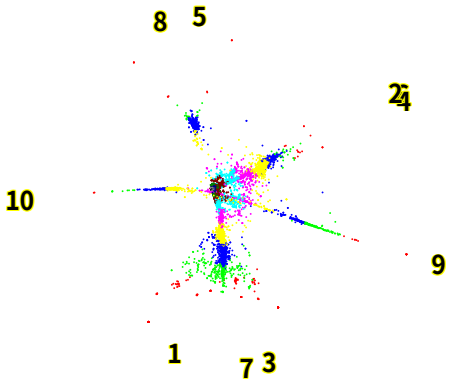

(g) Class-label assignment


(h) Color-map assignment

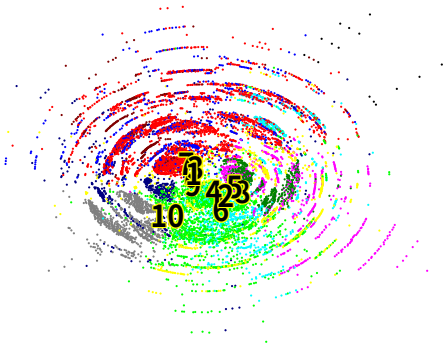
Fig. 4: Visualization of IC model for Wikipedia network



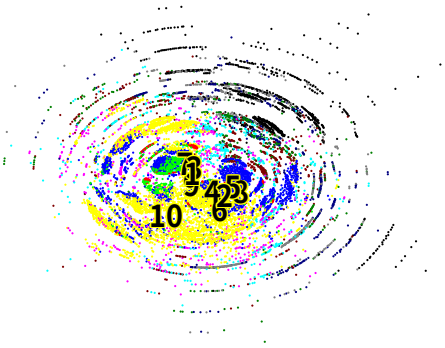

(a) Proposed method 1



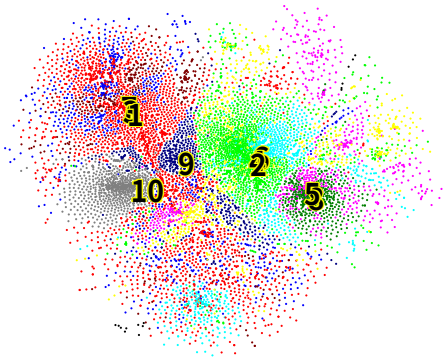

(b) Proposed method 2



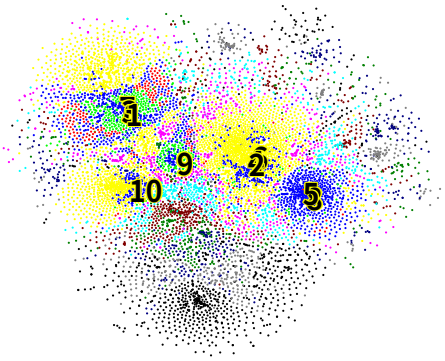

(c) Spring model method 1




(d) Spring model method 2




(e) Toplogy-based cross-entropy method 1




(f) Toplogy-based cross-entropy method 2


(g) Class-label assignment


(h) Color-map assignment

Fig. 5: Visualization of LT model for Wikipedia network