

# Probabilistic Latent Network Visualization: Inferring and Embedding Diffusion Networks

Takeshi Kurashima\*, Tomoharu Iwata<sup>+</sup>, Noriko Takaya\* and Hiroshi Sawada\*

\*NTT Service Evolution Labs., NTT Corporation

<sup>+</sup>NTT Communication Science Labs., NTT Corporation

\*1-1 Hikari-no-oka, Yokosuka-Shi, Kanagawa, 239-0847 Japan

<sup>+</sup>2-4 Hidaridai, Seika-Cho, Soraku-gun, Kyoto, 619-0237 Japan

\*+{kurashima.takeshi,iwata.tomoharu,takaya.noriko,sawada.hiroshi}@lab.ntt.co.jp

## ABSTRACT

The diffusion of information, rumors, and diseases are assumed to be probabilistic processes over some network structure. An event starts at one node of the network, and then spreads to the edges of the network. In most cases, the underlying network structure that generates the diffusion process is unobserved, and we only observe the times at which each node is altered/influenced by the process. This paper proposes a probabilistic model for inferring the diffusion network, which we call *Probabilistic Latent Network Visualization* (PLNV); it is based on cascade data, a record of observed times of node influence. An important characteristic of our approach is to infer the network by embedding it into a low-dimensional visualization space. We assume that each node in the network has latent coordinates in the visualization space, and diffusion is more likely to occur between nodes that are placed close together. Our model uses maximum a posteriori estimation to learn the latent coordinates of nodes that best explain the observed cascade data. The latent coordinates of nodes in the visualization space can 1) enable the system to suggest network layouts most suitable for browsing, and 2) lead to high accuracy in inferring the underlying network when analyzing the diffusion process of new or rare information, rumors, and disease.

## Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data Mining*; I.5.1 [Pattern Recognition]: Models

## Keywords

Diffusion network; network visualization; survival analysis

## 1. INTRODUCTION

In a broad range of domains such as viral marketing, information propagation in social network, and disease infection, there has been a great interest in analyzing the diffusion process with respect to the underlying network or, equivalently, a graph. Diffusion starts at one node and then spreads from node to node to the edge of

the network. Understanding the mechanism that causes the diffusion process helps to optimize advertising strategies, predict meme propagation, or counter infection. Some existing methods assume that the network structure (i.e. a set of connectivities) is predefined [5, 6]. However, in most cases, the exact path of the diffusion of information, influence, and disease over networks is unknown, and we only observe when a node mentions information, makes a decision, or becomes infected. Inferring and understanding the underlying network that causes the diffusion process is quite challenging and remains an interesting subject for study [14, 16, 7, 21, 3, 4].

We propose a probabilistic model that processes observed cascade data to infer diffusion networks, which we call *Probabilistic Latent Network Visualization* (PLNV). A cascade is, for one process, a record of when nodes are altered/influenced by the process. We have two criteria for designing a method that can estimate diffusion networks. First, suggesting a network that is embedded into a low-dimensional visualization space is important. In general, visualization techniques are very useful for understanding high-dimensional and complex data such as networks since they enable us to browse intuitively through huge numbers of nodes. The desired property of the diffusion network layout in terms of the visualization space is as follows; each node attempts to place its influential nodes relatively closer than non-influential ones in the visualization space, we call this the *influence preservation principle*. For example, a marketer, who wants to trigger a chain reaction of people into ordering goods for sale, can, from the initial sight of the layout, identify communities wherein the nodes strongly influence each other. A visualization space that satisfies this principle also serves as a vehicle for suggesting the growth process of the cascade since it seems that diffusion spreads over the visualization space gradually. Second, inferring the underlying diffusion network accurately is important, particularly when many cascades remain hidden. When we analyze the diffusion process of new information, disease, and rumors, only a little amount of cascade data might be available. Understanding the underlying diffusion network soon after its occurrence leads to a quick response to a crisis such as the spread of false rumors or disease.

With these criteria in mind, we develop a method for inferring diffusion networks. An important feature of our approach is to infer the network by embedding it into a low-dimensional visualization space. We assume that each node in the network has latent coordinates in the visualization space, and diffusion is more likely to occur between nodes that are placed close together. Our model learns the latent coordinates of nodes that best explain the observed cascade data. By incorporating this assumption into the generative process of diffusion, our model can suggest the diffusion network layout in the visualization space that fulfills the influence preserva-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
KDD'14, August 24–27, 2014, New York, NY, USA.  
Copyright 2014 ACM 978-1-4503-2956-9/14/08 ...\$15.00.  
<http://dx.doi.org/10.1145/2623330.2623646>.

tion principle. Moreover, our proposed model can recognize latent influence relationships between nodes even if we have not observed the common events linking the nodes, since the distances between nodes in the low-dimensional visualization space implies the presence of an influence relation.

There has been great interest in processing cascade data to infer underlying networks. Some approaches infer only the network structure [15, 17], while others infer not only the network structure but also the strength of every edge in the network [13, 14, 16, 7, 21, 3, 4]. The strength of each edge, called transmission rate, represents how quickly or likely information is to spread from one node to another. The network structure estimated by existing diffusion models is complex when the network is large. Visualization methods such as Multi-Dimensional Scaling (MDS) [20], Isomap [19], and the spring method [9], can embed high-dimensional networks into low dimensional spaces, and help in the study of the inferred network. Existing visualization methods can employ a set of distances in the inferred diffusion network as input. The separation distance for each node pair is obtained from the transmission rate of the two nodes.

However, the distances between nodes in the network may not be suitable when they are embedded in the visualization space since the distances are estimated in a space different from the visualization space. Furthermore, the objective function maximized by existing visualization methods is not the likelihood of all cascades, but rather that of input distances. Thus it is impossible for existing visualization methods to determine which distance should be preferentially restored in the visualization space based on the likelihood function of all cascades. Moreover, the accumulation of errors in the network inference process cannot be corrected in the embedding process with existing visualization methods since they are modulated, which may result in poor visualization. Our proposal simultaneously infers and visualizes the diffusion network in one probabilistic framework. Therefore, latent coordinates are estimated so as to be optimal in terms of the likelihood of all cascades when nodes are embedded in the low-dimensional visualization space.

Additionally, when analyzing diffusion networks of new information, disease, and rumors, the network inference performance of existing models is likely to be poor since there are few events that can provide clues as to the influence relations. In existing network inference models, the influence relation of each pair of nodes is recognized only after observing common events that link the pair members. The number of parameters (i.e. transmission rates) to be estimated by existing models is the square of the number of nodes. Thus a lot of cascades are required to estimate all parameters. On the other hand, the proposed model has far fewer parameters. Thus it can capture the essential features of the underlying diffusion network even if the number of cascades is small.

We demonstrate the effectiveness of the proposed model on the *MemeTracker* dataset, which records information diffusion among Web sites. We conduct two quantitative experiments. The first quantitatively shows that the network layout yielded by our proposed model satisfies the influence preservation principle. The second experiment compares its network inference performance against that of the existing network inference model, and shows that ours yields better performance when the number of cascades is small. Our qualitative experiments show that the visualization space learned by our model is not only useful for browsing the underlying network layout intuitively, but also serves as a vehicle for suggesting the growth process of the observed diffusion data.

**Table 1: Notation**

Symbol	Description
$n$	node (e.g. person, news article, and blog post)
$N$	number of nodes
$c$	cascade (event)
$\mathcal{C}$	set of cascades
$\mathbf{t}^c$	observed infection times of cascade, $c \in \mathcal{C}$
$T^c$	length of the observation window of cascade $c$
$\beta$	scale parameter
	for the width of the visualization space
$D$	dimensionality of the visualization space

## 2. PROBABILISTIC LATENT NETWORK VISUALIZATION

### 2.1 Preliminaries

Our model is based on survival analysis [10, 11]. By utilizing time information, survival analysis provides a powerful way to model event occurrence such as infection and information diffusion. We first present the basic concept of basing diffusion models on survival analysis.

We start with a set of cascades  $\mathcal{C} = \{\mathbf{t}^1, \dots, \mathbf{t}^{|\mathcal{C}|}\}$ . A cascade is represented by  $N$ -dimensional vector  $\mathbf{t}^c = (t_1^c, \dots, t_N^c)$ , where  $N$  is the number of nodes. Cascade  $\mathbf{t}^c$  is the history of event  $c$  within the population during time interval  $[0, T^c]$ . For example, when we analyze the spread of an infection, a certain infectious disease (say influenza) corresponds to an event, and people are represented by nodes. When we analyze information diffusion in Web space, a news topic of a newly-released mobile phone is an event, and a personal blog site that refers to it can be represented by a node.  $t_n^c$  of event  $c$  records the time stamp when event  $c$  occurs at or to node  $n$ , and  $t_n^c \in [0, T^c] \cup \{\infty\}$ . The symbol  $\infty$  labels nodes that are not reached in a cascade within observation window  $[0, T^c]$ . The clock is set to 0 at the start of each cascade (i.e. the time stamp of the node where each event happens first is set to 0). The notations used are summarized in Table 1.

The transmission function  $f(t_i|t_j)$  is the conditional likelihood of transmission time from node  $j$  to node  $i$ ; it represents the likelihood of an event happening to node  $i$  at time  $t_i$  given that the same event has already happened to node  $j$  at time  $t_j$ . If  $f(t_i|t_j)$  is the probability density function of time,  $F(t_i|t_j) = \int_{t_j}^{t_i} f(t) dt$  and is thus its cumulative distribution function. Given that node  $j$  was infected at time  $t_j$ ,  $F(t_i|t_j)$  represents the probability that node  $i$  was infected by node  $j$  within the period of time  $t_j$  to time  $t_i$ . Only if  $t_j < t_i$  can node  $j$ , which was infected at time  $t_j$ , infect node  $i$  at time  $t_i$ .

The *survival function* is defined as the probability that node  $i$  is not infected by node  $j$  before time  $t_i$ ; it is calculated by

$$S(t_i|t_j) = 1 - F(t_i|t_j) = \int_{t_i}^{\infty} f(t) dt. \quad (1)$$

The survival function is a monotonically decreasing function with  $S(\Delta_{ji} = 0) = 1$  and  $S(\Delta_{ji} = \infty) = 0$ , where  $\Delta_{ji} = t_i - t_j$  is the time difference between node  $j$  and node  $i$ .

The *hazard function* is the instantaneous infection rate, the rate at which node  $i$ , which has not been infected yet (up to time  $t_i$ ), will become infected within a small interval after time  $t_i$ . It is calculated by

$$h(t_i|t_j) = \frac{f(t_i|t_j)}{S(t_i|t_j)}. \quad (2)$$

**Table 2: Model**

Model	Transmission function $f(t_i t_j)$	Log survival function $\log S(t_i t_j)$	Hazard function $h(t_i t_j)$
Weibull	$\mu \alpha(\mathbf{x}_i, \mathbf{x}_j) (\Delta_{ji})^{\mu-1} \exp(-\alpha(\mathbf{x}_i, \mathbf{x}_j) (\Delta_{ji})^\mu)$	$-\alpha(\mathbf{x}_i, \mathbf{x}_j) (\Delta_{ji})^\mu$	$\mu \alpha(\mathbf{x}_i, \mathbf{x}_j) (\Delta_{ji})^{\mu-1}$
Exponential	$\alpha(\mathbf{x}_i, \mathbf{x}_j) \exp(-\alpha(\mathbf{x}_i, \mathbf{x}_j) \Delta_{ji})$	$-\alpha(\mathbf{x}_i, \mathbf{x}_j) \Delta_{ji}$	$\alpha(\mathbf{x}_i, \mathbf{x}_j)$
Rayleigh	$\alpha(\mathbf{x}_i, \mathbf{x}_j) \Delta_{ji} \exp\left(-\alpha(\mathbf{x}_i, \mathbf{x}_j) \frac{(\Delta_{ji})^2}{2}\right)$	$-\alpha(\mathbf{x}_i, \mathbf{x}_j) \frac{(\Delta_{ji})^2}{2}$	$\alpha(\mathbf{x}_i, \mathbf{x}_j) \Delta_{ji}$

Here we focus on the shift-invariant transmission function whose value only depends on the time difference  $\Delta_{ji}$ . That is to say,  $f(t_i|t_j) = f(\Delta_{ji})$ ,  $F(t_i|t_j) = F(\Delta_{ji})$ ,  $S(t_i|t_j) = S(\Delta_{ji})$ , and  $h(t_i|t_j) = h(\Delta_{ji})$ .

The likelihood  $\ell(t^c)$  of a cascade induced by event  $c$  is the product of all individual likelihoods  $\ell_n(t^c)$  of event  $c$  occurring to each node  $n$ . If we do not observe event  $c$  happening to node  $n$  in observation window  $[0, T^c]$ , the individual likelihood is calculated as the product of survival functions,

$$\ell_n^{\text{survive}}(t^c) = \prod_{\{j|t_j^c \leq T^c\}} S(\Delta_{jn}^c), \quad (3)$$

since node  $n$  is not infected by nodes that have already been infected up to  $T^c$ .  $\Delta_{ji}^c$  is the time difference between node  $j$  and node  $i$  as recorded in cascade  $c$ ; it is calculated by

$$\Delta_{ji}^c = \begin{cases} T^c - t_j^c & \text{if } t_i^c > T^c \\ t_i^c - t_j^c & \text{otherwise.} \end{cases} \quad (4)$$

The observation of infection plays a much more prominent role in determining the likelihood of transmission. Let us consider that event  $c$  happens to node  $n$  at  $t_n^c$  in observation window  $[0, T^c]$ . In this case, we consider all possible transmission routes. One possibility is that node  $n$  was infected by already-infected node  $j$  (i.e.  $t_j^c < t_n^c$ ), and is not infected by any other infected node. The likelihood is calculated as the sum of all possibilities:

$$\ell_n^{\text{infected}}(t^c) = \sum_{\{j|t_j^c < t_n^c\}} f(\Delta_{jn}^c) \prod_{\{k|k \neq j, t_k^c < t_n^c\}} S(\Delta_{kn}^c). \quad (5)$$

The likelihood  $\ell(t^c)$  of a cascade induced by event  $c$  can be calculated by the product of these two scenarios:

$$\ell(t^c) = \prod_{\{n|t_n^c > T^c\}} \ell_n^{\text{survive}}(t^c) \times \prod_{\{n|t_n^c \leq T^c\}} \ell_n^{\text{infected}}(t^c). \quad (6)$$

Finally, the likelihood of all cascades  $\mathcal{C}$  is the product of individual cascade likelihoods. The negative log likelihood of all cascades can be described using the hazard and the survival functions as follows:

$$L(\mathcal{C}) = - \sum_i \sum_j \sum_{\{c|t_j^c < t_i^c\}} \log S(\Delta_{ji}^c) - \sum_i \sum_{\{c|t_i^c \leq T^c\}} \log \sum_{\{j|t_j^c \leq t_i^c\}} h(\Delta_{ji}^c). \quad (7)$$

## 2.2 Proposed Model

The proposed model infers the diffusion network by embedding it into a low-dimensional visualization space. Our model estimates node coordinates  $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ , where  $\mathbf{x}_n = (x_{n1}, \dots, x_{nD})$  are the coordinates of the  $n$ th node in the visualization space;  $D$  is its dimensionality, usually  $D = 2$  or  $3$ . We assume that information is more likely to be propagated between nodes that are located close together in the visualization space. We estimate coordinates  $\mathbf{X}$  such that we can explain the observed cascade data. By incorporating this assumption into the generative process of diffusion, our

model suggests the diffusion network layout in the visualization space that fulfills the influence preservation principle.

We assume that the transmission function  $f(\Delta_{ji})$  from node  $j$  to  $i$  is determined by the Euclidean distance between them in the visualization space. Here, we use the Weibull model to define the transmission function as follows;

$$f(\Delta_{ji}|\mathbf{x}_i, \mathbf{x}_j) = \mu \alpha(\mathbf{x}_i, \mathbf{x}_j) (\Delta_{ji})^{\mu-1} \exp(-\alpha(\mathbf{x}_i, \mathbf{x}_j) (\Delta_{ji})^\mu), \quad (8)$$

where  $\alpha(\mathbf{x}_i, \mathbf{x}_j) \geq 0$  is the transmission rate (represents how likely node  $j$  with coordinates  $\mathbf{x}_j$  is to infect node  $i$  with  $\mathbf{x}_i$ );  $\mu$  determines the shape of the probability density function. As  $\alpha(\mathbf{x}_i, \mathbf{x}_j)$  approaches zero, the expected diffusion time becomes arbitrarily long (i.e. node  $j$  does not infect node  $i$ ). The transmission rate  $\alpha(\mathbf{x}_i, \mathbf{x}_j)$  is assumed to be determined by the Euclidean distance in the visualization space as follows:

$$\alpha(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\beta}{2} \|\mathbf{x}_j - \mathbf{x}_i\|^2\right), \quad (9)$$

where  $\beta$  is the scale parameter of the visualization space. The assumption underlying our model appeals to our intuition. Consider an outbreak of influenza in the real world. The infection rate is high among those people in close proximity. For example, when a person has been infected, her/his family, friends, and colleagues are also likely to be infected since she/he often meets them in the real world. However, the risk of infection fades over time since the infected person recovers from her/his illness. In this example, the likelihood of infection depends on the individual's activity area like her home, school, and office. Thus the learned visualization space is expected to reflect the geographical closeness between people in the real world.

By plugging (8) and (9) into (1) and (2), the log survival and hazard functions of our model are

$$\log S(\Delta_{ji}|\mathbf{x}_i, \mathbf{x}_j) = - \exp\left(-\frac{\beta}{2} \|\mathbf{x}_j - \mathbf{x}_i\|^2\right) (\Delta_{ji})^\mu, \quad (10)$$

$$h(\Delta_{ji}|\mathbf{x}_i, \mathbf{x}_j) = \mu \exp\left(-\frac{\beta}{2} \|\mathbf{x}_j - \mathbf{x}_i\|^2\right) (\Delta_{ji})^{\mu-1}. \quad (11)$$

Our model has the capacity to deal with a wide variety of cascade data since its definition is based on the Weibull distribution. Some existing network inference methods use exponential or Rayleigh distributions for modeling the process of diffusion [14, 16, 21, 3]. The exponential distribution is a special case of the Weibull distribution where  $\mu = 1$ . Furthermore, the Weibull distribution with  $\mu = 2$  is a good approximation of the Rayleigh distribution. Parameter  $\mu$  is predefined by the user of our system to suit the cascade type. Table 2 summarizes the transmission, survival, and hazard functions of each distribution.

## 2.3 Inference

We use maximum a posteriori (MAP) estimation to determine the unknown parameters  $\mathbf{X}$  in the proposed model that yield best

fits to the observed cascade data. The number of dimensions,  $D$ , is assumed to be specified for visualization.

The negative log likelihood of parameters  $\mathbf{X}$  for the given set of cascades,  $\mathbf{C}$ , with prior is as follows:

$$\begin{aligned} L(\mathbf{X}|\mathbf{C}) = & \sum_i \sum_j \sum_{\{c|t_j^c < t_i^c\}} \exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_i\|^2\right) (\Delta_{ji}^c)^\mu \\ & - \sum_i \sum_{\{c|t_i^c \leq T^c\}} \log \sum_{\{j|t_j^c \leq t_i^c\}} \mu \exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_i\|^2\right) (\Delta_{ji}^c)^{\mu-1} \\ & - \sum_i \log\left(\left(\frac{\gamma}{2\pi}\right)^{\frac{D}{2}} \exp\left(-\frac{\gamma}{2}\|\mathbf{x}_n\|^2\right)\right), \end{aligned} \quad (12)$$

where  $\gamma$  is a hyper-parameter. In order to control the size of the resultant embedding, we use a Gaussian prior with zero mean and spherical covariance for coordinates  $\mathbf{x}_n$ . We use  $\gamma = 0.1$  in all experiments described in Section 4. The objective function can be minimized through the use of a gradient-based numerical optimization method such as the quasi-Newton method [12].

The gradient of  $\mathbf{x}_n$  used in gradient-based numerical optimization can be calculated by considering two cases as follows: If  $i = n$ , the gradient of  $\mathbf{x}_n$  is calculated by

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}_n} = & \sum_j \sum_{\{c|t_j^c < t_n^c\}} \beta(\mathbf{x}_j - \mathbf{x}_n) \exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_n\|^2\right) (\Delta_{jn}^c)^\mu \\ & - \sum_{\{c|t_n^c \leq T^c\}} \frac{\sum_{\{j|t_j^c < t_n^c\}} \beta(\mathbf{x}_j - \mathbf{x}_n) \exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_n\|^2\right) (\Delta_{jn}^c)^{\mu-1}}{\sum_{\{j|t_j^c < t_n^c\}} \exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_n\|^2\right) (\Delta_{jn}^c)^{\mu-1}} \\ & + \gamma \mathbf{x}_n. \end{aligned} \quad (13)$$

If  $j = n$ , the gradient of  $\mathbf{x}_n$  is calculated by

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}_n} = & \sum_i \sum_{\{c|t_n^c < t_i^c\}} \beta(\mathbf{x}_i - \mathbf{x}_n) \exp\left(-\frac{\beta}{2}\|\mathbf{x}_n - \mathbf{x}_i\|^2\right) (\Delta_{ni}^c)^\mu \\ & - \sum_i \sum_{\{c|t_i^c \leq T^c\}} \frac{\beta(\mathbf{x}_i - \mathbf{x}_n) \exp\left(-\frac{\beta}{2}\|\mathbf{x}_n - \mathbf{x}_i\|^2\right) (\Delta_{ni}^c)^{\mu-1}}{\sum_{\{j'|t_{j'}^c < t_i^c\}} \exp\left(-\frac{\beta}{2}\|\mathbf{x}_{j'} - \mathbf{x}_i\|^2\right) (\Delta_{j'i}^c)^{\mu-1}}. \end{aligned} \quad (14)$$

Combining the two above scenarios, yields the gradient of  $\mathbf{x}_n$  as follows:

$$\begin{aligned} \frac{\partial L}{\partial \mathbf{x}_n} = & \sum_j \sum_{\{c|t_n^c \neq t_j^c\}} \beta(\mathbf{x}_j - \mathbf{x}_n) \exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_n\|^2\right) (\Delta_{jn}^c)^\mu \\ & - \sum_{\{c|t_n^c \leq T^c\}} \sum_{\{j|t_j^c < t_n^c\}} \beta(\mathbf{x}_j - \mathbf{x}_n) P(n|j) \\ & - \sum_j \sum_{\{c|t_j^c \leq T^c\}} \beta(\mathbf{x}_j - \mathbf{x}_n) P(j|n) + \gamma \mathbf{x}_n, \end{aligned} \quad (15)$$

where  $P(n|j)$  is the conditional probability of infection from node  $j$  to node  $n$ , and  $P(j|n)$  is the conditional probability of infection from node  $n$  to node  $j$ . These values are calculated by

$$P(n|j) = \frac{\exp\left(-\frac{\beta}{2}\|\mathbf{x}_j - \mathbf{x}_n\|^2\right) (\Delta_{jn}^c)^{\mu-1}}{\sum_{\{i|t_i^c < t_n^c\}} \exp\left(-\frac{\beta}{2}\|\mathbf{x}_i - \mathbf{x}_n\|^2\right) (\Delta_{in}^c)^{\mu-1}}, \quad (16)$$

$$P(j|n) = \frac{\exp\left(-\frac{\beta}{2}\|\mathbf{x}_n - \mathbf{x}_j\|^2\right) (\Delta_{nj}^c)^{\mu-1}}{\sum_{\{i|t_i^c < t_j^c\}} \exp\left(-\frac{\beta}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\right) (\Delta_{ij}^c)^{\mu-1}}. \quad (17)$$

Scale parameter  $\beta$  can be estimated by using cross validation.

Existing network inference models directly infer transmission parameter  $\alpha_{ji}$ , which represents how likely information is to spread from node  $j$  to node  $i$ . The number of parameters to be estimated is  $N \times N$ , and the number of parameters increases as the square of  $N$ . Thus a lot of cascades are required to learn all parameters. On the other hand, the number of parameters to be estimated by our model is  $D \times N$ . In general,  $D$ , the dimensionality of the visualization space, is considerably smaller than the number of nodes  $N$ . Thus the proposed model has significantly fewer parameters than existing models. Accordingly, our model can capture the essential features of underlying diffusion networks even if the number of cascades is small.

### 3. RELATED WORK

A great deal of effort has been expended to model diffusion processes with respect to the underlying network structure; published techniques are used for viral marketing, information propagation in social network, and following the spread of infectious diseases. Independent cascade models [5] and linear threshold models [6], two representative diffusion models, are mainly used for analyzing activities in online social communities. These models treat time in discrete steps rather than as a continuous variable although real events happen in continuous time. Some prior work treats time as a continuous variable. NETINF uses submodular optimization to estimate network connections [15]. NETINF assumes that the transmission rates between all nodes are predefined and not inferred. On the other hand, [14] proposed a continuous model called NETRATE; it allows different transmission rates across different edges. [14] provides a basic and simple framework for solving the network inference problem, and has been extended as follows; [16] uses stochastic gradients to estimate the time-varying transmission rates of edges and thus capture the temporal dynamics of the network. TOPICCASCADE proposed by [3] estimates topic-dependent transmission rates of edges by using LDA to represent topic structure of memes. MoNET considers additional features of nodes; it uses a predefined similarity measure between meme contents [21]. [7] used shared cascade Poisson processes to model information diffusion in online social communities, and proposed a Bayesian inference procedure. However, in the existing diffusion models, The influence relation of each node pair is recognized only after observing common events linking them. The performance of network inference decreases when the number of cascades is small since there are few events that can provide clues as to the influence relations. Our model infers the network embedded into a low-dimensional visualization space. Since the proposed model has significantly fewer parameters than existing models, ours accurately infers the underlying network even when there are few observed cascades. The distance in the visualization space indicates the strength of the influence relation between nodes even if we have not observed common event(s) that link them. The method proposed by [4] estimates a block matrix of transmission rates by combining ideas from stochastic block modeling and network inference models. The method by [4] is similar to ours in that it reduces the dimensionality of the diffusion network, but differs in that it can not visualize the diffusion network. Ours not only infers the low-dimensional representation of diffusion network but also visualizes it in a Euclidean space, which is a metric space that most closely corresponds to our intuitive understanding of space.

The network structure estimated by existing diffusion models is complex when the network is large. This complexity hinders the scientific discovery of knowledge or principles from the results since we can not understand the results in an intuitive way. In order to embed and visualize high-dimensional data such as diffusion

networks, a number of visualization methods have been proposed [19, 20, 18, 9]. These methods attempt to embed nodes into a low-dimensional Euclidean space in order to restore predefined pairwise distances between nodes. However, the 2-step process frequently results in poor visualization for the reasons noted in Section 1. In order to best explain the observed diffusion process, our model simultaneously estimates the latent coordinates of nodes and visualizes them in the same probabilistic framework.

In the sense of utilizing latent coordinates for modeling generative probabilistic models, the proposed model is related to *Probabilistic Latent Semantic Visualization* (PLSV), which extracts topics by embedding documents in a low-dimensional space [8]. While PLSV is designed for visualizing documents, our proposal, PLNV, is designed for visualizing networks.

## 4. EXPERIMENTS

### 4.1 Data

We evaluated the proposal by analyzing the information diffusion process occurring in Web space. For this we used the MemeTracker dataset of [16]. In this dataset, a cascade is represented by a set of time-stamps. Each time-stamp represents when a news article or blog post referred to a keyword (an event), and is recorded at the time scale of seconds. Each cascade in the dataset is categorized into topics based on what type of keyword it referred to. We chose eight types of cascade data related to eight topics; “iPhone”, “Jobs”, “Baseball”, “Basketball”, “Earthquake”, “Fukushima”, “Sept. 11”, and “Syria”. Each topic-dependent-dataset consisted of the top 500 sites and contained about 1,000 cascades.

### 4.2 Baseline methods

We compared the proposed method against various existing network inference and visualization methods. We used NETRATE as the state-of-the-art network inference method for estimating the weight for each edge in the directed graph based on observed cascade data. NETRATE directly estimates transmission parameter  $\alpha_{ji}$  which represents how likely information is to spread from node  $j$  to node  $i$ . For both our proposed model and NETRATE, we define the transmission function as an exponential model for simplicity. That is to say, our model was defined based on the Weibull distribution with  $\mu = 1$ . Based on an  $N \times N$  matrix of transmission rates estimated by NETRATE, we embedded  $N$  nodes into a  $D$ -dimensional visualization space using the KK spring method [9], MDS [20], or Isomap [19] as existing visualization methods.

Although NETRATE outputs a similarity matrix where each element is the degree of similarity, the visualization method expects a distance matrix as its input. In the distance matrix, as an element approaches infinity, it is much less likely to be infected. Thus, as a pre-processing step,  $\alpha_{ji}$  is converted into its inverse  $\hat{\alpha}_{ji}$ , which represents how far  $j$  is from  $i$ .

$$\hat{\alpha}_{ji} = \frac{1}{\alpha_{ji} + \epsilon}, \quad (18)$$

where  $\epsilon$  is a smoothing factor (this yields a value when  $\alpha_{ji} = 0$ ). We used  $\epsilon = 10^{-3}$  in this experiment.

The KK spring method is a visualization method that uses a pairwise graph-theoretic distance matrix. We define the graph-theoretic distance from node  $j$  to node  $i$  as  $\hat{\alpha}_{ji}$ . The method determines the coordinates of each node in the visualization space such that the following objective function is minimized.

$$L_{KK} = \frac{1}{2} \sum_j \sum_{\{i|i \neq j\}} \frac{(\hat{\alpha}_{ji} - \|\mathbf{x}_j - \mathbf{x}_i\|)^2}{\hat{\alpha}_{ji}^2}. \quad (19)$$

MDS is a linear dimensionality reduction method, and can embed nodes so as to minimize the discrepancy between pairwise distances in the visualization space and those in the original space. We also used Isomap as a non-linear dimensionality reduction method; it first constructs a graph connecting  $h$ -nearest neighbors, and then embeds nodes so as to preserve the shortest path distances in the graph by MDS. Note that MDS and Isomap require the input to be a symmetric matrix (i.e.  $\alpha_{ji} = \alpha_{ij}$ ). We calculated distance  $\bar{\alpha}_{ji}$  ( $= \bar{\alpha}_{ij}$ ), used as input to MDS and Isomap, by two methods listed below.

- $\bar{\alpha}_{ji} = \text{Minimum}(\hat{\alpha}_{ji}, \hat{\alpha}_{ij})$
- $\bar{\alpha}_{ji} = \text{Average}(\hat{\alpha}_{ji}, \hat{\alpha}_{ij})$

*Minimum* chooses and returns the minimum value from among the arguments; *Average* calculates and returns the average score of arguments.

Finally, we compared our method with six baselines; the KK spring method, Minimum-MDS, Average-MDS, Minimum-Isomap, and Average-Isomap.

### 4.3 Quantitative Experiments

#### 4.3.1 Performance of Visualization

This experiment quantitatively shows that our method can output a visualization space wherein each node is placed closer to its influential nodes than non-influential ones. We consider that  $j$  is an influential node for node  $i$  if  $i$  is likely to be infected by  $j$ . That is to say, we determine if the visualization space as estimated by our method satisfies the influence preservation principle.

We divided cascades into training and test sets, and evaluated whether or not the visualization space learned from the training data could predict the influence relations present in the test data. We considered that the influence relation between nodes is true (there exists a influence relation between nodes) if test data has an event (cascade) wherein both of them are infected.

We designed an evaluation metric to evaluate embedded diffusion network layouts in terms of how well the influence relations were preserved. We adopted the idea of the *F-measure*, which is widely used in the research field of information retrieval [1]. The method of calculating the F-measure score for the results of the visualization space is as follows: The F-measure combines *precision* and *recall*. For each node  $n$ , we consider a  $D$ -dimensional ball  $B_n(r_n)$  with center  $\mathbf{x}_n$  and radius  $r_n$ . The precision for node  $n$  is the fraction of nodes that are contained in the  $D$ -dimensional ball  $B_n(r_n)$  that have influence relations to node  $n$ ; it is calculated by

$$\text{Precision}_n(r_n) = \frac{\#\{k|\mathbf{x}_k \in B_n(r_n), I(n, k) = 1, k \neq n\}}{\#\{k|\mathbf{x}_k \in B_n(r_n), k \neq n\}}, \quad (20)$$

where  $I(n, k) = 1$  if there is an influence relation and  $I(n, k) = 0$  otherwise. The recall evaluates the sensitivity of the method. The recall for node  $n$  is the fraction of nodes that have influence relations that are successfully contained in  $B_n(r_n)$ ; it is calculated by

$$\text{Recall}_n(r_n) = \frac{\#\{k|\mathbf{x}_k \in B_n(r_n), I(n, k) = 1, k \neq n\}}{\#\{k|I(n, k) = 1, k \neq n\}}. \quad (21)$$

The *F-measure* for node  $n$  is the weighted harmonic average of precision and recall as follows

$$\text{F-measure}_n(r_n) = 1 / \left( \frac{1}{2} \cdot \frac{1}{\text{Precision}_n(r_n)} + \left(1 - \frac{1}{2}\right) \cdot \frac{1}{\text{Recall}_n(r_n)} \right). \quad (22)$$

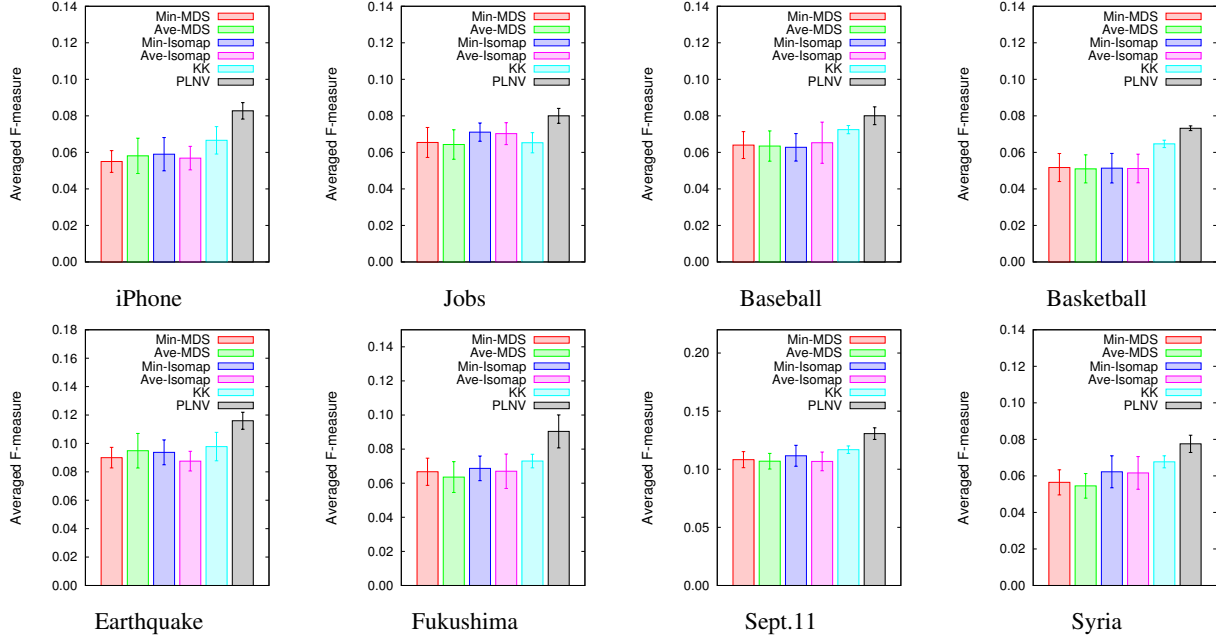


Figure 1: Comparisons of averaged F-measure in terms of the visualization task. The number of dimensions,  $D$ , is 2.

The final score for a visualization space is the averaged F-measure over all  $N$  nodes. In general, a small  $r_n$  leads to high precision and low recall, while a large one leads to low precision and high recall. For each method and node  $n$ , we chose  $r_n$  that yielded the best F-measure.

Figures 1 and 2 show the results of the F-measures so as to compare methods with different dimensions,  $D = 2$  and 3. For more reliable results, 5-fold cross validation was used for testing: We divided the cascades into five subsets. One subset was used for testing, and the others were used for training. This was repeated using each of the subsets as the test set. Thus, the Y-axis in these figures shows the averaged score over all trials. We set 86,400 seconds (a day) as  $T^c$  for our model and baseline methods. For simplicity, we assume  $T^c = 86,400$  for all cascades. Parameter  $\beta$  of our model was set to the best parameter value based on the results of 5-fold cross validation;  $\beta$  was chosen from  $10^1, 10^2, 10^3, 10^4, 10^5$ , and  $10^6$ . As shown, for each dataset (topic), our model yielded better performance than the baseline methods. The results show that the proposed method ensured that node pairs likely to be infected in the same event were placed close to each other. The performance of each baseline method strongly depends on the topic. Based on the type of analyzed data, a user of baseline method may have to find the right combination of distance metric and visualization method which would demand a lot of effort. Our model allows the user to avoid this trial and error process since it simultaneously estimates the coordinates of nodes in one probabilistic framework.

### 4.3.2 Performance of Network Inference

This experiment evaluates our proposed model in terms of network inference performance. When only few cascades are observed, the  $N \times N$  matrix output by existing network inference models is likely to be sparse since there are few events available to provide clues as to the underlying diffusion network. However, inferring underlying the network with few observations is important. As noted before, our method can infer robust models even when

there are few cascades. Our model predicts an unknown element in the sparse matrix (i.e. transmission rate between nodes) based on the distance in the low-dimensional space. More intuitively, even if two nodes have not experienced infection in the same event, the proposed method can predict the existence of connectivity between them from whether the existence of their common influential nodes. This experiment shows that our proposal can accurately predict connections between users even when relatively few cascades have been observed.

We evaluated the diffusion network estimated by the network inference model from the training data in terms of how well it explains the influence relations present in the test data. As in the previous experiment, we used the F-measure as the evaluation metric. Let  $w_{kn}$  be the distance from node  $k$  to node  $n$  as estimated by each network inference method, and  $d_n$  be its threshold which determines whether node  $k$  is an influential node for node  $n$  or not. The precision for node  $n$  represents the probability that the influential node for node  $n$  is located within  $d_n$ ; it is calculated by

$$Precision_n(d_n) = \frac{\#\{k|w_{kn} \leq d_n, I(n, k) = 1, n \neq k\}}{\#\{k|w_{kn} \leq d_n, n \neq k\}}. \quad (23)$$

Recall for node  $n$  is the fraction of its influential nodes that are actually located within  $d_n$ ; it is calculated by

$$Recall_n(d_n) = \frac{\#\{k|w_{kn} \leq d_n, I(n, k) = 1, n \neq k\}}{\#\{k|I(n, k) = 1, n \neq k\}}. \quad (24)$$

The F-measure score is calculated by (22). We compared the averaged F-measures of our model against those of an existing network inference model, NETRATE. We used  $\hat{\alpha}_{kn}$  for  $w_{kn}$  when calculating the F-measure of NETRATE, and  $\|\mathbf{x}_k - \mathbf{x}_n\|$  for  $w_{kn}$  when calculating the F-measure of the proposed model. One parameter that also influences the performance of our method is the number of dimensions  $D$ . Although the number of dimensions  $D$  in our model is set to 2 or 3 for visualization, a larger number of dimensions can be set in our model. This experiment considered four values of  $D$  (2,3,5, and 10).

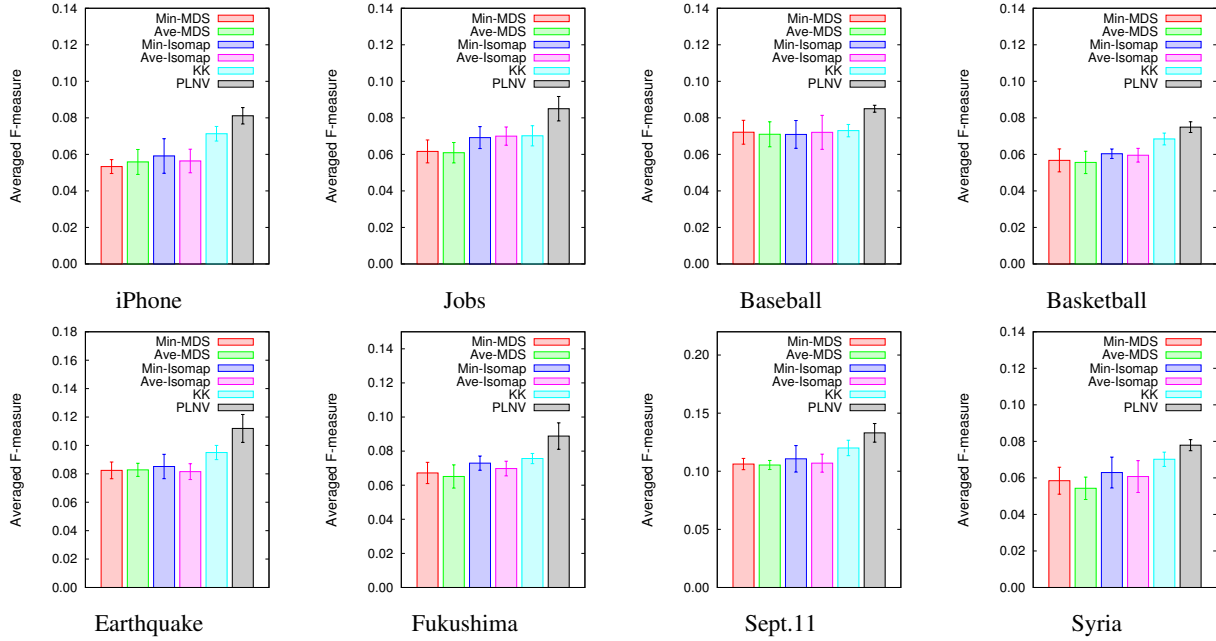


Figure 2: Comparisons of averaged F-measure in terms of the visualization task. The number of dimensions,  $D$ , is 3.

We studied the influence of the number of cascades on the performance of network inference. Figure 3 shows the performance of each model at different cascade sizes. We changed the number of cascades from 100 to 1,000 in steps of 100. For each step, we calculated the averaged F-measure. As shown, the proposed model demonstrated better network inference performance than NETRATE when the number of cascades was less than 300. The results suggest that our model is appropriate for solving network inference problems with sparse cascade data. In some datasets (“Earthquake” and “Sept.11”), even when the number of cascades is relatively large, the proposal inferred connections with the same or better accuracy than NETRATE. In the previous experiment, we tested the performance of visualization on the condition that a lot of cascades were used in training (the number of cascades was set to 1,000). The low accuracy of NETRATE in terms of network inference with the sparse training data implies low accuracy in terms of visualization, since the existing visualization method learns the visualization space by maximizing the likelihood of the given parameters (i.e. transmission rates) from NETRATE. Even if it succeeds in constructing a low-dimensional representation of NETRATE parameters, the existing visualization method yields only inferior visualization performance, particularly when few cascades are observed.

#### 4.4 Qualitative Experiments

This experiment qualitatively discusses the value of our model for two different application scenarios.

**Browsing network layout:** While it remains difficult to evaluate whether a given network layout is suitable for browsing or not, we observed the characteristic features of each method. Figure 4 illustrates the 2- and 3-dimensional layout yielded by each method examined. We used the cascades related to “Baseball”. Based on the results of the quantitative experiments, we adopted the minimum function as the distance measure for MDS and Isomap since it yields better performances than the average function for most datasets. In the layouts created by MDS and Isomap, many nodes

are collapsed to a single point, which is not desirable for browsing. As shown, the layout created by the KK spring method avoids the node collapse problem, but many nodes tend to be placed on the circumferences of circles. This phenomenon is referred to as the *dandelion effect* [2]. Our model is superior to the KK spring method in terms of its support for community-finding. Our visualization space exhibits several densely-populated areas, which we call communities. The nodes in a community are likely to influence each another. An event experienced by a member is likely to be propagated throughout the community. When we use our method for the purpose of viral marketing, the marketer can get key information from the layouts created by the proposed method since the marketer can trigger a chain reaction of people wanting to the goods for sale. Even if the goods are new releases, for which there are few cascades, the proposed method can suggest the network layout that best explains the mechanism of the diffusion process to the marketer.

**Browsing growth process of cascade:** The visualization space output by our model serves as a vehicle for suggesting the growth process of a cascade since it satisfies the influence preservation principle. An event that occurs at one node in a community is expected to spread like an epidemic from node to node. Figure 5 shows examples of cascade visualizations yielded by the network layout. We used the cascade datasets related to three topics; “iPhone”, “Basketball”, and “Sept. 11”. For each topic, we chose the most widely propagated event (cascade), and show the infected and survival nodes at four different time points. In these figures, colored and uncolored nodes indicate infected and survival nodes, respectively. For example, the node that was the first to experience an event related to iPhone is plotted on (a). The nodes which became infected within 1200, 2400, and 3600 seconds of the start of the event are marked by the colored nodes in (b), (c), and (d), respectively. We compared our proposed method against the combination of NETRATE and the KK spring model. As shown, in the layout created by the KK spring method, each event seems to prop-

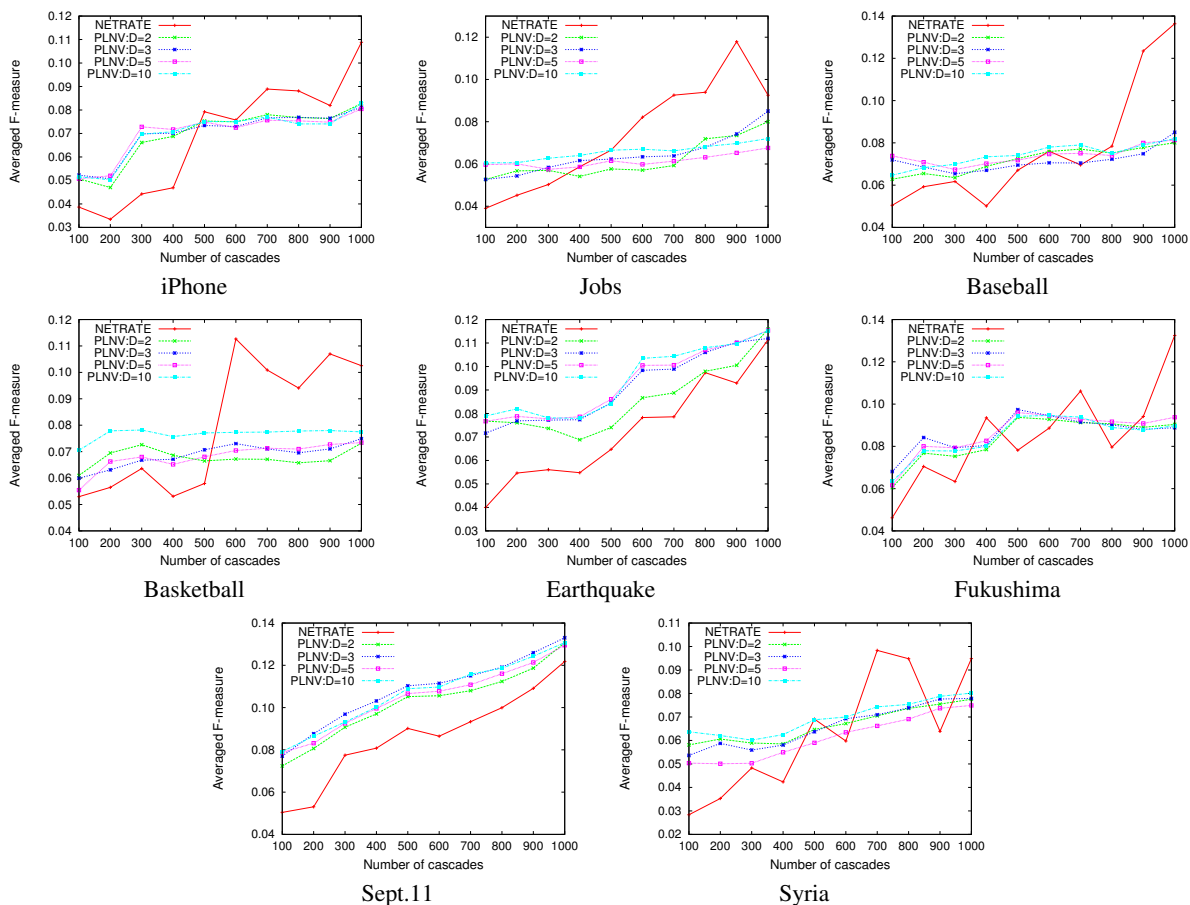


Figure 3: Comparisons of averaged F-measure in terms of the network inference task.

agate along the circumference of a circle since many nodes tend to be arranged on circles. On the other hand, in the layout created by the proposed model, each event propagates within a narrow space at an early stage, but diffusion eventually covers a much wider area. The structure enables us to grasp an appropriate understanding of the present situation since we can characterize each diffusion event by its speed, coverage, and density. Thus we can gain an understanding of the big picture through the visualization. Furthermore, an analysis of a diffusion process is expected to establish a reasonable understanding of future infection patterns since diffusion gradually spreads over the visualization space. For example, to prevent the initial carrier from spreading his infectious disease or spreading a rumor, the people in his community can be put on their guard.

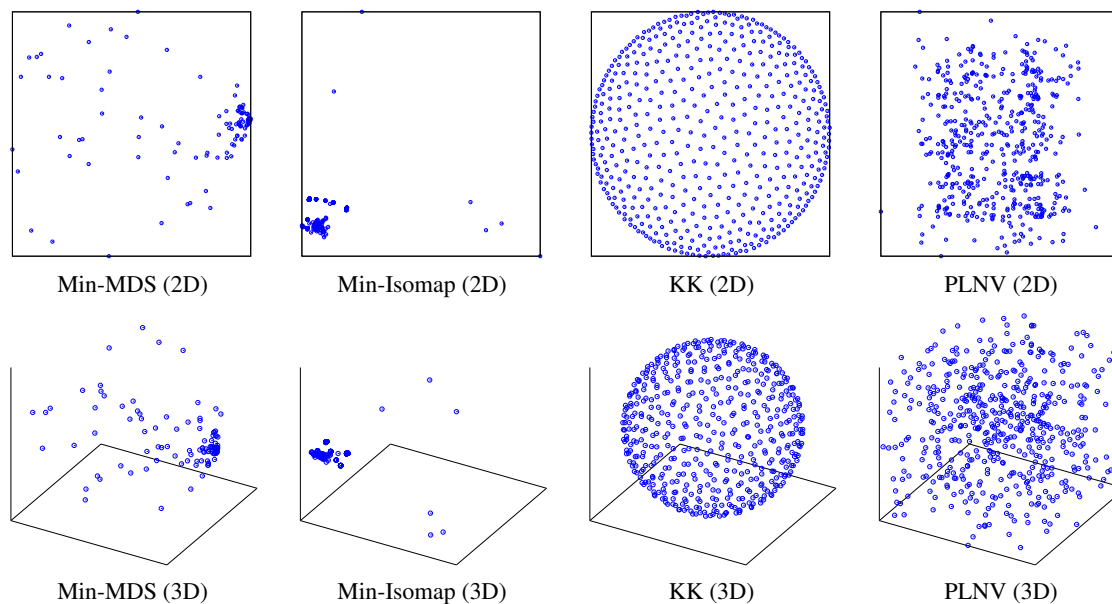
## 5. CONCLUSION

This paper proposed a probabilistic model for estimating diffusion networks and embedding them into low-dimensional visualization spaces. We consider that the desired property of the layout of the diffusion network is that each node should be more closely co-sited with its influential nodes than with non-influential ones in the visualization space; we call this the influence preservation principle. This property is realized by the proposed model through the following assumption; each node in a network has latent coordinates in the visualization space, and propagation is more likely to occur between nodes that are placed closer together. Our model learns the latent coordinates in the visualization space that best ex-

plain the observed cascade data; our model is fitted to the given cascade data by using maximum a posteriori estimation. Different from previous works, our model simultaneously estimates the latent coordinates of nodes in the visualization space in one probabilistic framework. As a result, the coordinates of nodes are optimal in terms of the likelihood of all cascades. Another important feature of our model is its robustness. The parameters in our model can be learned accurately even when there are few cascades since the number of parameters to be estimated is significantly smaller than that in existing network inference models. Our model can accurately predict unknown influence-relations between nodes based on the distance in the learned space, even if we have not yet observed a common event which both nodes are infected.

We used the histories of information diffusion in a Web space to evaluate the performance of network visualization, and quantitatively showed that the visualization results of the proposed model satisfy the influence preserving principle, unlike the existing methodology; it combines existing network inference and visualization methods. The visualization space learned by our model is not only useful for browsing the underlying network layout intuitively, but also serves as a vehicle for suggesting the growth process of the observed cascade. The second experiment showed that our model addresses the sparsity problem better than existing network inference models. We compared its performance in the network inference task to that of an existing network inference method, and showed that ours yields better performance when there are few cascades.





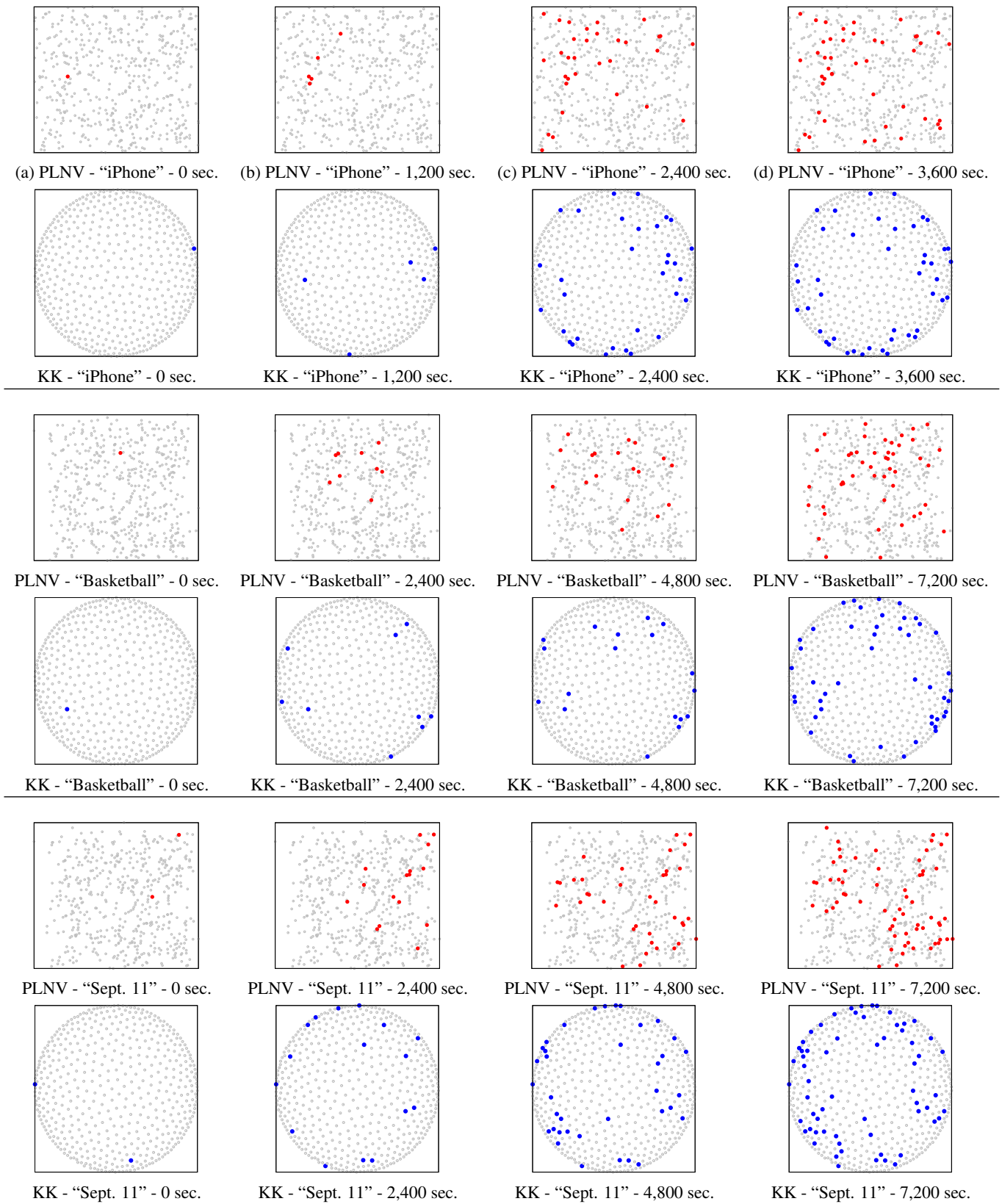
**Figure 4: Comparisons of 2-, and 3- dimensional visualizations of each method. Each visualization space was learned from a set of cascades related to “Baseball”.**

This is because our model allows us to associate nodes based on the distance in the visualization space, even in the absence of any event that involves both nodes.

Our experiments, offline evaluations using observed cascade data, are an important starting point for exploring the validity of our model, but are only one approach to evaluating visualization methods. We will let real users in realistic settings evaluate the real strengths of our approach. Another future direction is to use other types of cascade data for evaluating the effectiveness of our model. We will analyze diffusion process with respect to information propagation in social networks, and the spread of infectious diseases.

## 6. REFERENCES

- [1] R. Baeza-Yates, B. Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
- [2] A. Buja, D. F. Swayne, M. L. Littman, N. Dean, H. Hofmann, and L. Chen. Data visualization with multidimensional scaling. *Journal of computational and graphical statistics*, 17(2):444–472, 2008.
- [3] N. Du, L. Song, H. Woo, and H. Zha. Uncover topic-sensitive information diffusion networks. In *AISTATS’13*, pages 229–237, 2013.
- [4] C. DuBois, C. Butts, and P. Smyth. Stochastic blockmodeling of relational event dynamics. In *AISTATS’13*, pages 238–246, 2013.
- [5] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.
- [6] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420, 1978.
- [7] T. Iwata, A. Shah, and Z. Ghahramani. Discovering latent influence in online social activities via shared cascade poisson processes. In *KDD’13*, pages 266–274, 2013.
- [8] T. Iwata, T. Yamada, and N. Ueda. Probabilistic latent semantic visualization: Topic model for visualizing documents. In *KDD’08*, pages 363–371, 2008.
- [9] T. Kamada and S. Kawai. An algorithm for drawing general undirected graphs. *Information processing letters*, 31(1):7–15, 1989.
- [10] J. F. Lawless. *Statistical models and methods for lifetime data*, volume 362. John Wiley & Sons, 2011.
- [11] E. T. Lee and J. W. Wang. *Statistical methods for survival data analysis*. John Wiley & Sons, 2013.
- [12] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.
- [13] S. A. Myers and J. Leskovec. On the convexity of latent social network inference. In *NIPS’10*, pages 1741–1749, 2010.
- [14] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML’11*, pages 561–568, 2011.
- [15] M. G. Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD’10*, pages 1019–1028, 2010.
- [16] M. G. Rodriguez, J. Leskovec, and B. Schölkopf. Structure and dynamics of information pathways in online media. In *WSDM’13*, pages 23–32, 2013.
- [17] M. G. Rodriguez and B. Schölkopf. Submodular inference of diffusion networks from multiple trees. In *ICML’12*, pages 1019–1028, 2012.
- [18] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.
- [19] J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- [20] W. S. Torgerson. *Theory and methods of scaling*. 1958.
- [21] L. Wang, S. Ermon, and J. E. Hopcroft. Feature-enhanced probabilistic models for diffusion network inference. In *ECML/PKDD’12*, pages 499–514, 2012.



**Figure 5: Comparisons of 2-dimensional cascade visualization.** We compared the proposed method against the combination of NETRATE and the KK spring model. The infected and survival nodes at four different time points are shown.