SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2015, Vol. 58 092111:1–092111:15
doi: 10.1007/s11432-015-5288-8

Information diffusion network inferring and pathway tracking

ZHOU DongHao[1,2*], HAN WenBao[2], WANG YongJun[1] & YUAN BaoDi[3]

[1]*College of Computer, National University of Defense Technology, Changsha* 410073*, China;*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou* 450002*, China;*
[3]*College of Computer Science and Technology, Fudan University, Shanghai* 200433*, China*

Received October 9, 2014; accepted December 9, 2014; published online April 30, 2015

Abstract Network diffusion, such as spread of ideas, rumors, contagious disease, or a new type of behaviors, is one of the fundamental processes within networks. Designing effective strategies for influence spread maximization, rumor spread minimization, or epidemic immunization has attracted considerable research attention. However, a key challenge is that many times we can only observe the trace of contagion spreading across network, but the underlying network structure is unknown and the transmission rates between node pairs are unclear to us. In this paper, given the observed information cascades, we aim to address two problems: diffusion network structure inferring and information diffusion pathways tracking. We propose a novel probabilistic model called Network Inferring from Multidimensional Features of Cascades (NIMFC) which takes into account heterogeneous features, including temporal and topological features of cascades, node attributes, and information content, to infer the latent network structure and transmission rates of edges. Also, based on the inferred network structure, we may track diffusion pathways of a cascade in social networks. We use blocked coordinate descent method to learn a sparse estimation of the latent network. Our proposed model NIMFC is evaluated both on large synthetic and real-world data sets, and experimental results show that our method significantly outperforms state-of-the-art models both in terms of recovering the latent network structure and information pathway tracking.

Keywords diffusion network, network inferring, information pathway tracking, temporal feature, node attributes, topological feature, information cascades

Citation Zhou D H, Han W B, Wang Y J, et al. Information diffusion network inferring and pathway tracking. Sci China Inf Sci, 2015, 58: 092111(15), doi: 10.1007/s11432-015-5288-8

1 Introduction

Diffusions over networks, such as the spread of technical innovations, infectious disease, information, rumors, or emotions [1], are a pervasive phenomenon in many networks. Recently, with emergence of Web 2.0, more and more people use social media (e.g., Twitter or Weibo) or social networks (e.g., Facebook) to share and spread ideas, photos, or even rumors. The information diffusion in social networks plays an important role in many real-world events, like Facebook during the 2010 Arab spring [2] or Twitter during the 2008 US presidential elections [3]. In China, Weibo has also proved to be a powerful tool for propagation of news, hot events, and public opinions.

* Corresponding author (email: dhzhou2084@163.com)

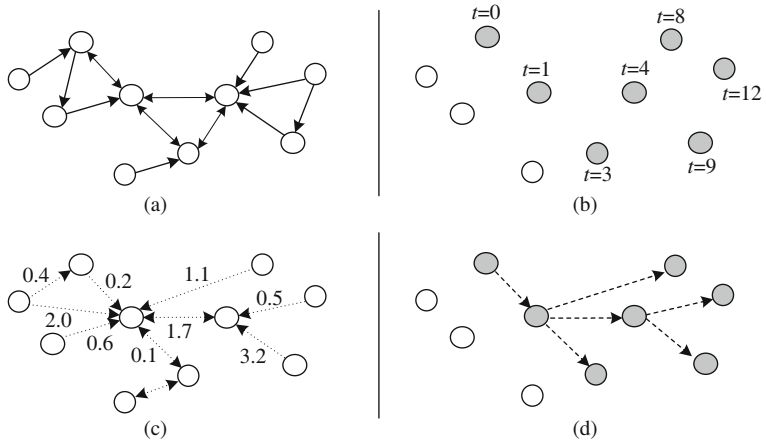


Figure 1 Diffusion network inference and information pathways tracking problems. (a) Underlying diffusion network, which is unknown to us; (b) a piece of information spread across the diffusion network, where the pathway is unknown to us while the infected time of different nodes can be observed; (c) inferred network structure and transmission rates; and (d) inferred diffusion pathways of a contagion.

Understanding the structure of diffusion network and uncovering the dynamics of information diffusion in social networks have attracted much research attention in the past few years. Researchers have been studying extensive problems like influence spread maximization [4,5], rumor spread minimization [6], and information outbreak detection [7]. When doing this, they often take the assumption that the underlying network is known, which in practice is not true in many situations.

Often, we can observe the trace of a contagion (e.g., a piece of information or a popular meme) spreading in a network, which is often named *cascade*, without knowing the latent network structure or the information diffusion pathways, as shown in Figure 1. For example, we see people get infected by an infectious disease at different times, but we don't know who infected them. We observe people talking about a hot event or a fresh idea without knowing from whom did he heard of the information. So, a natural question is how to infer the relationship between these nodes and how to quantify the strength of the relationship in terms of information diffusion. Another interesting question is whether we can recover the way that the information takes to spread across the network.

Given observed information cascades, we need to solve the following problems: (1) how to infer the underlying diffusion network structure and (2) how to track the diffusion pathways of the information, as shown in Figure 1.

To achieve these goals, there are some sources of information that can be used, such as temporal and topological properties of cascades, node attributes, and information content. The temporal feature, that is, the infected timestamps of nodes in a cascade, is used with the assumption that infection events closer in time are more likely to be causally related, which is reasonable in practice. For node attributes, if two nodes share more common attributes, such as interested topics, profession, and education background, they are more likely to spread information between each other. So as to topological properties, we refer it to as overlapping structure of different cascades.

Only recently we have approaches for network inference [8–15]. Some of them can only infer latent edges of the diffusion network [9,12], while others like [8,10] can also calculate the transmission rate or weight for each edge in the diffusion network. Most of them [8–10,15] use only temporal features of information cascades as source data, while some of them try to take into account not only the temporal features but also node attributes [13] or information content [14].

We emphasize that it is important to merge all these features together when inferring network structure, because, first, node attributes and other rich features can complement the temporal feature, leading to more precise inference of network structure, and second, if one source of information is noisy, or sparse, others can make up for it. However, combining all these different modalities of information together is also challenging.

In this paper, we propose a uniform probabilistic model which takes into account the temporal and

topological features of cascades, node attributes, and information content features to infer underlying diffusion network structure. We refer to this new model as Network Inferring from Multidimensional Features of Cascades (NIMFC). Also, our model can also be used to track the diffusion pathways of the information, that is, to infer the edges across which a piece of information was transmitted to the target nodes in past time. This can be very helpful in locating the source of rumors (or misinformation), or tracking the information diffusion pathways among criminals.

The rest of the paper is organized as follows. Section 2 briefly surveys related work. In Section 3, we give an introduction of survival analysis theory used in our model. We describe the probabilistic model of NIMFC in detail in Section 4. In Section 5, we discuss how to learn the parameters of our model. We proceed by describing experimental evaluation in Section 6 and conclude in Section 7.

2 Related work

In recent years, a substantial amount of models or algorithms have been proposed for inferring underlying network structure from observed data. Eagle et al. [16] proposed a method that used time series data of collected mobile phone to infer weighted friendship network. Kolar et al. [17] attempted to learn time-varying network structures from multivariate time series. A common feature is that they all treated time as discrete step rather than a continuous random variable.

Some researchers then attempted to model information diffusion process in continuous time. For example, Gomez-Rodriguez et al. [9] proposed a model called NETINF which aimed to infer near-optimal set of k directed edges maximizing the likelihood of observed cascades, and they exploited submodular function optimization to learn parameters of the model.

Meyers and Leskovec [10] inferred not only the connectivity but also a prior probability of infection for every edge using a convex program and some heuristics. However, they considered the transmission rate between all nodes as a fixed constant. Gomez-Rodriguez et al. [8] later proposed a more flexible model called NETRATE, which used survival theory to model the process of information diffusion between node pairs. An obvious advantage of this model is that the instantaneous infection rate can be changed over time with different decaying model, like exponential, power-law, or Rayleigh model. However, all of the aforementioned methods focused on only temporal features of cascades, ignoring other features like node attributes or information content, which we believe are also important for inferring underlying network structure.

Wang et al. [13] proposed a model called MoNET which took into account not only the time differences between events but also node attributes to solve the problem. Du et al. [14] combined temporal feature of information cascades and meme content to propose a probabilistic model TOPICCASCADE, which explicitly modulated the transmission likelihood by the topic distribution of each meme. Yang et al. [18] considered the problem of simultaneous and entangling diffusion process of multiple memes over a hidden network, and proposed a probabilistic mixture model over multivariate Hawkes process for diffusion network inference and meme tracking.

Our work is distinct from those existing works in two aspects: (1) our model provides a uniform framework which can combine heterogeneous features, including temporal and topological properties of cascades, node attributes, and information content, to yield a more precise estimation of the latent network and (2) our model can be extended to track the information pathway. Other models like NETINF, NETRATE, or MoNET were not clearly announced to be used in information pathway tracking.

We summarize the related work in Table 1 and compare them in several dimensions.

3 Preliminaries

In this section, we will give a brief introduction of survival analysis theory [19], which will be used to handle temporal feature in our proposed model. Survival analysis deals with analysis of time duration until one or more events happen, such as death in biological organisms and infection by a infectious disease.

Table 1 Comparison of methods for diffusion network inferring

Methods	Goals to achieve			Features used			
	Link inference	Transmission rate inference	Pathway tracking	Temporal feature	Node attributes	Cascades topology	Information content
NETINF	√	×	×	√	×	×	×
NETRATE	√	√	×	√	×	×	×
MoNET	√	√	×	√	√	×	×
TOPICCASCADE	√	√	×	√	×	×	√
NIMFC (our model)	√	√	√	√	√	√	√

The event in this paper is referred to as a piece of information transmitted from one node to another, or in other words, an "infection". Let T be a continuous non-negative random variable representing the waiting time until the occurrence of an event. We assume the probability density function (p.d.f.) $f(t)$ and cumulative distribution function (c.d.f.) can be written as

$$F(t) = P(T < t) = \int_0^t f(x)\mathrm{d}x, \tag{1}$$

which represents the probability that the event has occurred by time t. The survival function is complement of $F(t)$ in the form of

$$S(t) = P(T > t) = \int_t^\infty f(x)\mathrm{d}x = 1 - F(t), \tag{2}$$

which gives the probability that the event of interest has not occurred by time t. The survival function is non-increasing, that is, $S(t_a) \leqslant S(t_b)$ if $t_a > t_b$ and usually one assumes $S(0) = 1$ and $S(t) \to 0$ as $t \to \infty$.

The hazard function is instantaneous rate of occurrence of the event at time t conditional on survival until time t or later, which is defined as

$$H(t) = \lim_{\mathrm{d}t \to 0} \frac{P(t \leqslant T \leqslant t + \mathrm{d}t)}{\mathrm{d}t \cdot S(t)} = \frac{f(t)}{S(t)}. \tag{3}$$

Note that from (2), one can get $-f(t)$ being the derivative of $S(t)$, so $H(t)$ can be rewritten as

$$H(t) = \frac{-S'(t)}{S(t)} = -\frac{\mathrm{d}}{\mathrm{d}t}\log S(t). \tag{4}$$

Then, we can get $S(t) = \exp\{-\int_0^t H(x)\mathrm{d}x\}$.

4 Network structure inferring and information pathway tracking

In this section, we first give a formal definition of the inferring problem, and then build a probabilistic model by adding different features into it step by step. These features include temporal and topological features of cascades, node attributes, and information content. Note that all these features can be extracted from the observed information cascades subtly, without adding any new data set as input. Last, we describe how to extend NIMFC model to track information pathway.

4.1 Problem statement

Let $G(V, E)$ indicate a social network, and V is the node set and E is the edge set. A contagion spreads across the network. Because the latent network structure is unknown, we can only get the trace of the contagion spreading across the network, that is, the "infected " time of different nodes that are infected by the contagion. The diffusion trace can be defined as a **cascade**. It can be represented as a N-dimensional

vector $c := \{t_1^c, t_2^c, \ldots, t_N^c\}$, with t_i^c the ith infection timestamp of information c. Let t^c denote the upper bound of different infection time in cascade c. In a considerable long time window T^c, we may observe many information cascades formed by different messages, which form a cascades set $C = \{c|t^c < T^c\}$.

In addition to the observed information cascades, we may also get some other information of a node, for example, its interested topics, which can be drawn from messages published or forwarded by him in history. In this paper, for a node's attributes, we only focus on its interested topics, and this can be extracted from the observed cascades C. Let's denote the attributes of node j as a K-dimensional vector, $\boldsymbol{X}^{(j)} = \{x_1^{(j)}, x_2^{(j)}, \ldots, x_K^{(j)}\}$, with $x_i^{(j)}$ the ith attribute and $\boldsymbol{X} = \{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(|V|)}\}$. Then, the features of information can also be represented as a L-dimensional vector,$\boldsymbol{Y}^{(c)} = \{y_1^{(c)}, y_2^{(c)}, \ldots, y_L^{(c)}\}$, and then we have $\boldsymbol{Y} = \{\boldsymbol{Y}^{(1)}, \boldsymbol{Y}^{(2)}, \ldots, \boldsymbol{Y}^{(|C|)}\}$. Here, we only focus on topical features of the information content, and specifically, we use topic model like latent Dirichlet allocation (LDA) [20] to analyze it. Therefore, $\boldsymbol{Y}^{(c)}$ can be represented as the topic distribution of the information content of c.

For the underlying network, an adjacent matrix $\boldsymbol{A} = \{\alpha_{i,j}|i, j \in [1, N],\ i \neq j,\ \alpha_{i,j} \geqslant 0\}$ is defined with $\alpha_{i,j}$ being the strength or transmission rate of edge $e_{i,j}$. If $\alpha_{i,j} = 0$, then there is no link between nodes i and j, otherwise there is an edge, and a larger $\alpha_{i,j}$ means a larger transmission rate. Parameter $\alpha_{i,j}$ is unknown and we aim to infer it.

So, the diffusion network structure inferring problem can be formally defined as, given the observed cascades C, node attributes $\boldsymbol{X}$, and information content feature $\boldsymbol{Y}$, finding an estimation of $\boldsymbol{A}$ that can maximize the likelihood of the observed cascades C, that is,

$$\widehat{\boldsymbol{A}} = \underset{\boldsymbol{A}}{\operatorname{argmax}}\, L(C) = \underset{\boldsymbol{A}}{\operatorname{argmax}} \prod_{c \in C} P(c|\boldsymbol{A}, \boldsymbol{X}, \mathbf{Y}). \tag{5}$$

The inferred network with adjacent matrix $\widehat{\boldsymbol{A}}$ is represented as $\widehat{G}$.

4.2 Inferring network structure from multidimensional features of cascades

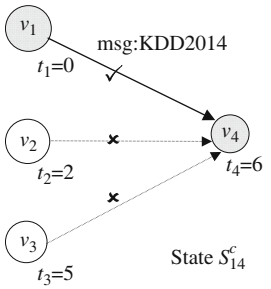
Diffusion state. For cascade $c = \{t_1^c, t_2^c, \ldots, t_N^c\}$ with N infected nodes, if $t_j^c < t_i^c$, then node j has a possibility to infect node i in past time, and we say node j is node i's potential parent. Note that the potential parent–child relationship is different from real parent–child relationship in the underlying network. For an infected node i, all the infected nodes with timestamp smaller than node i are potential parent of node i, and the number is denoted as $M(i)$. For simplicity, we assume all potential infection behaviors are independent and a node can only has one parent in a cascade. That is to say, if node i was infected by node j, then it can't be infected by all other parental nodes, that is, it survived from all other parental nodes. We define such a scenario as a *diffusion state* of node i, which is denoted as S_{ji}^c. Thus, node i has $M(i)$ discrete diffusion states, with each diffusion state corresponding to one parental node. All diffusion states form diffusion state space S_i^c. Figure 2 shows an example of possible diffusion states of an infected node. The cascade has four infected nodes v_1, v_2, v_3, v_4, and their infected timestamps are $t_1 = 0$, $t_2 = 2$, $t_3 = 5$, $t_4 = 6$. Node v_4 has three potential parental nodes corresponding to three diffusion states. Figure 2(a) shows the diffusion state S_{14}^c, where node v_4 gets infected by node v_1 and survives from other nodes and Figure 2(b) shows the diffusion state S_{34}^c where node v_4 gets infected by node v_3 and survives from other nodes.

We define the possibility distribution of diffusion states space S_i^c as

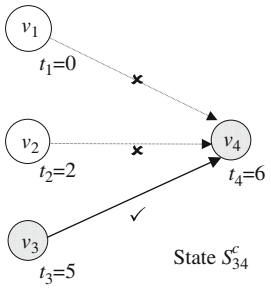
$$\pi_i^c = \{\pi_{1i}^c, \pi_{2i}^c, \ldots, \pi_{M(i)i}^c\}, \quad \text{s.t.} \quad \sum_{j=1}^{M(i)} \pi_{ji}^c = 1, \tag{6}$$

where π_{ji}^c denotes the possibility of diffusion state S_{ji}^c.

Different diffusion states may have different possibilities depending on node attributes and information content, in addition to infection time difference. As shown in Figure 2, although node v_3 is temporally closer to node v_4 than node v_1, but if taken into consideration their interested topics and the message content, node v_1 may be more likely being node v_4's real parent than node v_1, because they are both



(a)



(b)

Node	Topic distribution
v_1	DM: 0.6 HPC: 0.4
v_2	NS: 0.8 DM: 0.2
v_3	CV: 0.7 NS: 0.3
v_3	DM: 0.6 HPC: 0.4

(c)

Figure 2 Diffusion states for an infected node. v_1, v_2, and v_3 are potential parents of v_4. (a) Diffusion state S^c_{14}. v_4 was infected by v_1 and survived from other nodes; (b) diffusion state S^c_{34}. v_4 was infected by v_3 and survived from other nodes; and (c) each node is assigned with a topic distribution. DM, data mining; NS, network security; HPC, high performance computing; CV, computer vision.

interested in topics of "data mining" and the message content "KDD 2014" is exactly about "data mining".

Basically, our model is based on the following intuitions.

(1) Nodes that share common attributes are likely to spread information to each other, as the saying goes, "birds of a feather flock together". Such homophyly effect in social networks has been widely studied by Crandall et al. and McPherson et al. [21,22].

(2) Nodes that have co-occurrence within the same cascades are likely to have larger strength between each other than that have not. This property makes a sense in that one can view a cascade as a group or loose community characterized by the unique information content, and nodes within the same community have stronger ties than nodes out of the community. Such an effect can accumulate for nodes within multiple overlapping cascades, that is, if two nodes both occurred in one or two cascades, it may be a random phenomenon, but if they both occurred in many cascades, then there must be some special relationship between them. We call such an attribute of a node as *cascades affinity* (CA), which we will describe in detail later.

(3) The information content also plays an important role in information diffusion process. In particular, people tend to read and spread messages they are interested in.

(4) Two infection events closer in time are more likely to be causally related. This is the base of previous models like NETRATE [8], NETINF [9], and MoNET [13].

Now, we shall proceed by describing how these intuitions are captured by NIMFC in further details. We will add multidimensional features, including topological and temporal features of cascades, nodes attributes, and information content feature, into our model step by step.

Modeling topological features of cascades, node attributes, and information content. To capture how these features influence the transmission behavior, we propose

$$\pi^c_{ji} = P(S^c_{ji}) = \frac{1}{Z_i} g(z^c_{ji}). \tag{7}$$

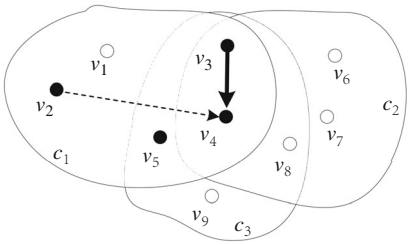
In (7), z^c_{ji} is an index that captures dissimilarity among three entities: node j, node i, and meme content c. In particular, we define z^c_{ji} as

$$z^c_{ji} = d(\boldsymbol{X}^{(j)}, \boldsymbol{X}^{(i)}) + d(\boldsymbol{X}^{(j)}, \boldsymbol{Y}^{(c)}), \tag{8}$$

where $d(\boldsymbol{X}^{(j)}, \boldsymbol{X}^{(i)})$ represents distance between node j and node i, and $d(\boldsymbol{X}^{(j)}, \boldsymbol{Y}^{(c)})$ represents the distance between node j and information content. In (7), $g(x)$ is a monotonic function which satisfies $g(x) : x \in [0, +\infty) \mapsto [0, 1]$, and without loss of generality, here we define

$$g(x) = \mathrm{e}^{-x}. \tag{9}$$

Z_i is a normalizing constant in the form of $Z_i = \sum_{t_j < t_i} g(z^c_{ji})$.



(a)

Cascades affinity
CA(2) = {c_1}
CA(3) = {c_1,c_2,c_3}
CA(4) = {c_1,c_2,c_3}
CA(5) = {c_1,c_3}

(b)

Figure 3 Cascades affinity (CA). (a) Overlapping structure of cascades, including c_1, c_2 and, c_3. Nodes v_2 and v_3 are potential parents of node v_4 in c_1 and (b) cascades affinity of different nodes. Jaccard distance of v_3 and v_4 is smaller than that of v_2 and v_4. So, v_3 is more likely to be v_4's parent.

Note that in (7), as z_{ji}^c shrinks, π_{ji}^c grows and vice versa. In other words, without other information, if node j and node i are more similar in their interest topics and cascades affinity (CA), the possibility node j being node i's parent is higher. The same is for node j and meme content c.

Now, we describe the feature extraction process as follows.

(1) Information content features. We focus on the topical features which can be readily obtained using standard topic model such as LDA [20].

(2) Node attributes. We exploit two kinds of features in particular. The first is topic distribution of a node, denoted as θ_i. For a node, the content of all cascades that occurred to him together reflects his topic distribution, which can be obtained in the way of information content feature extraction. The other is CA of a node, which is extracted from the topology of cascades as below.

(3) Topological features of cascades. The overlapping structure of cascades is shown in Figure 3. Node i's CA is the set of cascades that the node participates in, which can be obtained by examining all cascades that occurred to node i. Figure 3 gives a simple example of a node's CA. Node v_i's CA is denoted as $\mathrm{CA}(i)$. In Figure 3, node v_3's CA is $\mathrm{CA}(3) = \{c_1, c_2, c_3\}$, while node v_5's CA is $\mathrm{CA}(5) = \{c_1, c_3\}$.

Next, as for the choice of distance function $d(\cdot,\cdot)$, we propose that

(1) For topical similarity, a good choice would be Kullback–Leibler divergence $D_{\mathrm{KL}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j)$, which measures the difference between two discrete probability distributions. Assuming there are K topics, then Kullback–Leibler divergence can be represented as

$$D_{\mathrm{KL}}(\boldsymbol{\theta}_i, \boldsymbol{\theta}_j) = \sum_{k=1}^{k=K} \ln\left(\frac{\theta_i(k)}{\theta_j(k)}\right)\theta_i(k), \tag{10}$$

where $\theta_i(k)$ represents the kth element of vector θ_i.

(2) For node's CA, Jaccard distance is suitable here, that is,

$$D_J(i,j) = 1 - J(\mathrm{CA}(i), \mathrm{CA}(j)) = \frac{|\mathrm{CA}(i) \cup \mathrm{CA}(j)| - |\mathrm{CA}(i) \cap \mathrm{CA}(j)|}{|\mathrm{CA}(i) \cup \mathrm{CA}(j)|}, \tag{11}$$

where $J(\mathrm{CA}(i), \mathrm{CA}(j))$ is Jaccard similarity coefficient of $\mathrm{CA}(i)$ and $\mathrm{CA}(j)$.

The above model captures the influence of cascades topological features, node attributes, and information content on transmission probability. Next we will incorporate temporal features of cascade into our model.

Modeling temporal features of cascades. We define transmission function from node j to node i as $f(t_i|t_j;\alpha_{j,i})$, which is the conditional 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. The value depends on time difference $\Delta_{j,i} = t_i - t_j$ and transmission rate $\alpha_{j,i}$. It captures the temporal distance between two successive events from node j to i. If $\alpha_{j,i} = 0$, then it means that directed edge $j \to i$ doesn't exist in underlying network, and $f(t_i|t_j;\alpha_{j,i}) = 0, h(t_i|t_j;\alpha_{j,i}) = 0$. Three models are often used for function $f(t_i|t_j;\alpha_{j,i})$, that is, exponential model, power–law model, and Rayleigh model. In this paper, we use Rayleigh distribution to model transmission likelihood, where the infection possibility increases fast to a peak and then drops slowly. This Rayleigh distribution is widely used to model epidemiology [23] and information

diffusion [13,14]. The likelihood function for Rayleigh model is

$$\alpha_{j,i}(t_i - t_j)\exp\Big\{-\alpha_{j,i}\frac{(t_i-t_j)^2}{2}\Big\}, \tag{12}$$

where the survival function and hazard function are $\exp\big\{-\alpha_{j,i}\frac{(t_i-t_j)^2}{2}\big\}$ and $\frac{\alpha_{j,i}}{(t_i-t_j)}$, respectively.

First, let us consider the infected nodes in cascade c. In the diffusion state S_{ji}^c, node i was infected by node j, and it survived from all other parental nodes. Let $L_{ji}^{c,+}$ represent the likelihood of S_{ji}^c, when we only consider temporal feature of cascade c, that is,

$$L_{ji}^{c,+} = P(S_{ji}^c) = f(t_i^c|t_j^c;\alpha_{j,i}) \prod_{t_k^c<t_i^c, k\neq j} S(t_i^c|t_k^c;\alpha_{k,i}). \tag{13}$$

Merging all features together. The likelihood of node i being infected at time t_i, given other infected timestamps, can be obtained by summing over all diffusion states S_{ji}^c of node i and incorporating factor of π_{ji}^c, that is,

$$\begin{aligned} L_i^{c,+} &= \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) L_{ji}^{c,+} = \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) f(t_i^c|t_j^c;\alpha_{j,i}) \prod_{t_k^c<t_i^c, k\neq j} S(t_i^c|t_k^c;\alpha_{k,i}) \\ &= \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) H(t_i^c|t_j^c;\alpha_{j,i}) \prod_{t_k^c<t_i^c} S(t_i^c|t_k^c;\alpha_{k,i}). \end{aligned} \tag{14}$$

The term π_{ji}^c in (14) serves as a bias for different diffusion state S_{ji}^c, and it can be viewed as a weight of $L_{ji}^{c,+}$ in final likelihood $L_i^{c,+}$.

Note that, besides the infected nodes within cascade, there are also nodes not infected by the meme until time T^c. Next we consider the uninfected nodes outside cascade c. If node i is not infected by any nodes within cascade c, we say it survives from all the infected nodes within cascade c until time window T^c. Let $L_i^{c,-}$ denote the likelihood of such situation for node i, then we get

$$L_i^{c,-} = \prod_{t_j^c<t^c} S(T^c|t_j^c;\alpha_{j,i})). \tag{15}$$

Then, by combining the above two components together, we can obtain the overall likelihood of a cascade, that is,

$$\begin{aligned} L(c) &= \prod_{t_i^c<t^c} L_i^{c,+} \prod_{t_i^c>t^c} L_i^{c,-} \\ &= \prod_{t_i^c<t^c} \Big\{ \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) H(t_i^c|t_j^c;\alpha_{j,i}) \prod_{t_k^c<t_i^c} S(t_i^c|t_k^c;\alpha_{k,i}) \Big\} \prod_{t_i^c>t^c} \prod_{t_j^c<t^c} S(T^c|t_j^c;\alpha_{j,i})). \end{aligned} \tag{16}$$

Assuming all cascades are independent, the likelihood of cascades set C can be written as a product of these individual cascade likelihoods (in the form of logarithm), that is,

$$L(C) = \log \prod_{c\in C} L(c) = \sum_{c\in C} \log L(c). \tag{17}$$

Specially, we can write

$$L(C) = \Psi_1 + \Psi_2 + \Psi_3, \tag{18}$$

with

$$\begin{aligned} \Psi_1 &= \sum_{c\in C} \sum_{t_i^c<t^c} \log \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) H(t_i^c|t_j^c;\alpha_{j,i}), \\ \Psi_2 &= \sum_{c\in C} \sum_{t_i^c<t^c} \sum_{t_k^c<t_i^c} \log S(t_i^c|t_k^c;\alpha_{k,i}), \\ \Psi_3 &= \sum_{c\in C} \sum_{t_i^c>t^c} \sum_{t_j^c<t^c} \log S(T^c|t_j^c;\alpha_{j,i})). \end{aligned} \tag{19}$$

4.3 Information pathway tracking

The proposed model NIMFC can be further extended to track the diffusion pathway of contagion spatially across the network. The main idea for diffusion pathway tracking is, given a cascade c and infected nodes within c, to infer the most likely parent for each infected node. In real world, a node within c may be infected by nodes outside c or even nodes outside the network G, but here we take the assumption that an infected node can only be infected by nodes within cascade c. Let $\text{parent}_c(i)$ denote node i's real parent. Based on the value of transmission rate $\alpha_{j,i}, i, j \in [1, N]$, we propose

$$\begin{aligned}\text{parent}_c(i) &= \underset{j,t_j^c<t_i^c,\alpha_{j,i}\neq 0}{\text{argmax}} \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) L_{ji}^{c,+} = \underset{j,t_j^c<t_i^c,\alpha_{j,i}\neq 0}{\text{argmax}} \pi_{ji}^c f(t_i^c|t_j^c;\alpha_{j,i}) \prod_{t_k^c<t_i^c,k\neq j} S(t_i^c|t_k^c;\alpha_{k,i}) \\ &= \underset{j,t_j^c<t_i^c,\alpha_{j,i}\neq 0}{\text{argmax}} \pi_{ji}^c H(t_i^c|t_j^c;\alpha_{j,i}) \prod_{t_k^c<t_i^c} S(t_i^c|t_k^c;\alpha_{k,i}). \end{aligned} \tag{20}$$

The edge from $\text{parent}_c(i)$ to node i is the path which the contagion takes to spread to node i. By linking all these selected edges within cascade c, we can recover the whole picture of information spreading history across the underlying network, as shown in Figure 1(d).

5 Optimization algorithm

The maximization of $L(C)$ in (18) can be transformed and decomposed into N independent subproblems, with each corresponding to one specific node in network G. The subproblem with respect to node i can be defined as

$$\underset{\{\alpha_{j,i}\}_{j=1}^N}{\text{maximize}}\, L_i(\{\alpha_{j,i}\}_{j=1}^N), \quad \text{s.t.} \quad \alpha_{j,i} \geqslant 0, \tag{21}$$

where

$$\begin{aligned} L_i(\{\alpha_{j,i}\}_{j=1}^N) = &\sum_{\{c|t_i^c<t^c\}} \log \sum_{t_j^c<t_i^c} \pi_{ji}^c M(i) H(t_i^c|t_j^c;\alpha_{j,i}) + \sum_{\{c|t_i^c<t^c\}} \sum_{t_j^c<t_i^c} \log S(t_i^c|t_j^c;\alpha_{j,i}) \\ &+ \sum_{\{c|t_i^c>t^c\}} \sum_{t_j^c<t^c} \log S(T^c|t_j^c;\alpha_{j,i})). \end{aligned} \tag{22}$$

Note that in real world the diffusion networks are often sparse. To guarantee such a property and to avoid overfitting, we add an l_1-norm (Lasso) [24] regularization to (21). Then, we can rewrite the optimization subproblem as

$$\underset{\{\alpha_{j,i}\}_{j=1}^N}{\text{maximize}}\, L_i(\{\alpha_{j,i}\}_{j=1}^N) - \lambda \sum_{j=1}^N |\alpha_{j,i}|, \quad \text{s.t.} \quad \alpha_{j,i} \geqslant 0, \tag{23}$$

where λ is a regularization parameter.

By Using l_1-norm regularization, the algorithm will yield a sparse network structure with some of the parameter estimations being exactly zero. If $\alpha_{j,i} = 0$, then there is no edge between node i and node j. The larger the value of λ, the further some parameter estimations will shrink toward zero. Note that $L_i(\{\alpha_{j,i}\}_{j=1}^N)$ is a convex differentiable function and $\sum_{j=1}^N |\alpha_{j,i}|$ is separable. For such optimization problem, one can use blocked coordinate descent to obtain the global maximum, which is proved by Tseng and Mangasarian [24].

The coordinate descent procedure starts with some initial guess like $\{\alpha_{j,i}^{(0)} = 0\}_{j=1}^N$ and repeat for $k = 1, 2, 3, \ldots, k_{\max}$. In every loop, we iteratively update each element of $\{\alpha_{j,i}\}_{j=1}^N$ while keeping all

other parameters fixed, that is,

$$\begin{cases} \alpha_{1,i}^{(k)} = \underset{\alpha_{1,i}>0}{\operatorname{argmax}} L_i(\alpha_{1,i}, \alpha_{2,i}^{(k-1)}, \alpha_{3,i}^{(k-1)}, \ldots, \alpha_{N,i}^{(k-1)}) - \lambda|\alpha_{1,i}|, \\ \alpha_{2,i}^{(k)} = \underset{\alpha_{2,i}>0}{\operatorname{argmax}} L_i(\alpha_{1,i}^{(k)}, \alpha_{2,i}, \alpha_{3,i}^{(k-1)}, \ldots, \alpha_{N,i}^{(k-1)}) - \lambda|\alpha_{2,i}|, \\ \ldots \\ \alpha_{N,i}^{(k)} = \underset{\alpha_{N,i}>0}{\operatorname{argmax}} L_i(\alpha_{1,i}^{(k)}, \alpha_{2,i}^{(k)}, \ldots, \alpha_{N-1,i}^{(k)}, \alpha_{N,i}) - \lambda|\alpha_{N,i}|. \end{cases} \tag{24}$$

Note that after we update $\alpha_{j,i}^{(k)}$, we use its new value from then on. We stop iterating once the likelihood does not increase (by at least 0.01%) after a full iteration over all $\alpha_{j,i}$. In the following presentation, for the sake of notational convenience, we let $Q_{j,i}^{(k)}$ denote $L_i(\alpha_{1,i}^{(k)}, \ldots, \alpha_{j-1,i}^{(k)}, \alpha_{j,i}, \alpha_{j+1,i}^{(k-1)}, \ldots, \alpha_{N,i}^{(k-1)}) - \lambda|\alpha_{j,i}|$. For each $\alpha_{j,i}^{(k)}$ in (24), because $L_i(\{\alpha_{j,i}\}_{j=1}^N)$ is convex, we use projected gradient ascent method. The gradient can be computed straightforwardly as

$$\frac{\partial Q_{j,i}^{(k)}}{\partial \alpha_{j,i}} = \frac{\partial L_i(\alpha_{1,i}^{(k)}, \ldots, \alpha_{j-1,i}^{(k)}, \alpha_{j,i}, \alpha_{j+1,i}^{(k-1)}, \ldots, \alpha_{N,i}^{(k-1)})}{\partial \alpha_{j,i}} - \lambda. \tag{25}$$

Then, we update $\alpha_{j,i}^{(k)}$ by a step toward the direction of gradient ascent. Because of the constrain of $\alpha_{j,i}^{(k)} \geqslant 0$, we project $\alpha_{j,i}^{(k)}$ to a non-negative value in $[0, +\infty)$, that is,

$$\alpha_{j,i}^{(k)} := \max\left(0, \alpha_{j,i}^{(k)} + \delta \frac{\partial Q_{j,i}^{(k)}}{\partial \alpha_{j,i}}\right), \tag{26}$$

where δ is a learning rate.

Tuning of parameter. For l_1-norm penalty, large value of λ tends to shrink some $\alpha_{j,i}$ to be exactly zero, and thus imposes a sparse structure on network G. To tune parameter λ, we use cross-validation approach to test what value of λ can yield the best result on test data.

The whole algorithm is summarized in Algorithm 1.

Algorithm 1 NIMFC: Network Inferring from Multidimensional Features of Cascades

INPUT: $C = \{c|t^c < T^c\}$, $\boldsymbol{X} = \{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \ldots, \boldsymbol{X}^{(|V|)}\}$, $\boldsymbol{Y} = \{Y^{(1)}, \boldsymbol{Y}^{(2)}, \ldots, \boldsymbol{Y}^{(|C|)}\}$
OUTPUT: $\boldsymbol{A} = \{\alpha_{i,j}|i, j \in [1, N],\ \alpha_{i,j} \geqslant 0\}$
computation of $\{\pi_{ji}^c\}_{i,j\in[1,N],c\in C}$
for $i = 1$ to N **do**
  initialize: $\{\alpha_{j,i}^{(0)} = 1\}_{j=1}^N$
  while no convergence **do**
    for $j = 1$ to N **do**
      while no convergence **do**
        computation of $\frac{\partial Q_{j,i}^{(k)}}{\partial \alpha_{j,i}}$ according to (24) and (25)
        $\alpha_{j,i}^{(k)} := \max\left(0, \alpha_{j,i}^{(k)} + \delta \frac{\partial Q_{j,i}^{(k)}}{\partial \alpha_{j,i}}\right)$
      end while
    end for
    $k = k + 1$
  end while
end for

6 Experiments

In this section, we will evaluate our model on both synthetic data set and real-world data set,which is crawled from China's most popular online social media Sina Weibo. We choose two non-trivial models NETRATE and MoNET as baselines. For NETRATE, MoNET, and our model, we all use Rayleigh distribution to fit the diffusion process.

6.1 Experiment on synthetic data

We made synthetic data by generating social networks and information cascades on it which mimic information diffusion scenarios in real-world social networks. In such a situation, the structure of the network and transmission rate along each edge is predefined and can serve as ground truth when we quantify the performance of different models.

Network generation. We used the Kronecker graph model [25] to generate three types of directed diffusion networks: (1) core-periphery network with parameters ([0.9, 0.5; 0.5, 0.3]), (2) hierarchical Kronecker network with parameters [0.9, 0.1; 0.1, 0.9], and (3) random network with parameters [0.5, 0.5; 0.5, 0.5]). Each type of network has 2000 nodes.

Topics generation. Each node i is assigned with a topic distribution, which can be represented as a K-dimension vector θ_i. It is convenient to use a symmetric Dirichlet distribution $\mathrm{Dir}(\beta_i)$ to generate θ_i. The value β_i can be drawn from uniform distribution (0, 1). Then, we sample K-dimension vector θ_i from $\mathrm{Dir}(\beta_i)$ for each node i. Note that $\beta_i < 1$ guarantees a sparse distribution over topics for node i, that is, node i's interest focuses on a small set of topics.

Cascade generation. For a cascade c, root node i is uniformly chosen at random. Then, the topic distribution of meme content is drawn from $\mathrm{Dir}(\beta_i)$. The edge $e_{i,j}$ is assigned with a transmission rate $\alpha_{i,j}$, which has a negative correlation with $z^c_{i,j}$. Without loss of generality, we set $\alpha_{i,j} = (z^c_{j,i})^{-1} + \epsilon$, where $\epsilon \sim N(0, \sigma^2)$ represents a noise. The infected time t^c_i of root node i is set to 0. Once node i is infected, its neighbor has a possibility of being infected by it with a time delay $\Delta^c_{j,i} = t^c_j - t^c_i$, where $\Delta^c_{j,i}$ is drawn from Rayleigh distribution specified by parameter $\alpha_{j,i}$. The diffusion processes carry on in a breadth-first fashion until they exceed a predefined observation time window T^c or there are no more nodes that can be infected.

Experiment setup. For each type of network, we generate five sets of cascades respectively, with the cascade number varying from 1000, 3000, 5000, 10000 to 15000. The parameter λ in (23) is tuned by fourfold cross validation.

We implement NIMFC based on Stanford Network Analysis Platform (SNAP)[1)].

Evaluation metrics. We evaluate the performance of three models via two measures. First, we consider $F1$ score with definition of $F1 := \frac{2\cdot\mathrm{precision}\cdot\mathrm{recall}}{\mathrm{precision}+\ \mathrm{recall}}$, where precision is the fraction of edges in the inferred network $\widehat{G}$ that are also present in the true network G^*, and recall is the fraction of edges of the true network G^* that are also present in the inferred network $\widehat{G}$. The second metrics is mean relative error (MRE). $\mathrm{MRE} = E[|\alpha^* - \widehat{\alpha}|/\alpha^*]$, where α^* is the true transmission rate and $\widehat{\alpha}$ is the estimated transmission rate. The first measure is used to quantify the ability of three models for recovering latent network structure, while the other gives a deeper examination on the performance of inferring detailed transmission rate for the right inferred edges.

Experiment results. In Figure 4 (a)–(c), we display $F1$ scores for NETARTE, MoNET, and NIMFC on three types of synthetic networks as we increase the cascades number from 1000 to 15000. We compare the performance of three models for three types of networks, respectively. In all cases, we can see that as cascades number increase, the $F1$ scores of three models increase fast first and then get steady slowly. Three models all suffer from sparsity of data set when the number of cascades in smaller than 5000. Note that NIMFC outperforms NETARTE and MoNET consistently for each type of networks. For random network, the performance improvement of NIMFC is not as large as that in core–periphery network and hierarchy network. We expect that in social network with strong community structure, like core-periphery network and hierarchy network structure, the cascades structure shows strong topological properties, which can help to yield a better result, while for random network without community structure, the topological properties are weak.

Figure 4 (d)–(f) show the MREs of inferred transmission rates for NETARTE, MoNET and NIMFC on three types of synthetic networks. For the right inferred edges, we know the true transmission rates in advance, so we can test how close the inferred transmission rate is to the real one quantitatively. We observe that when cascade number is small (less than 3000), NIMFC shows no advantage over the

1) http://snap.stanford.edu.

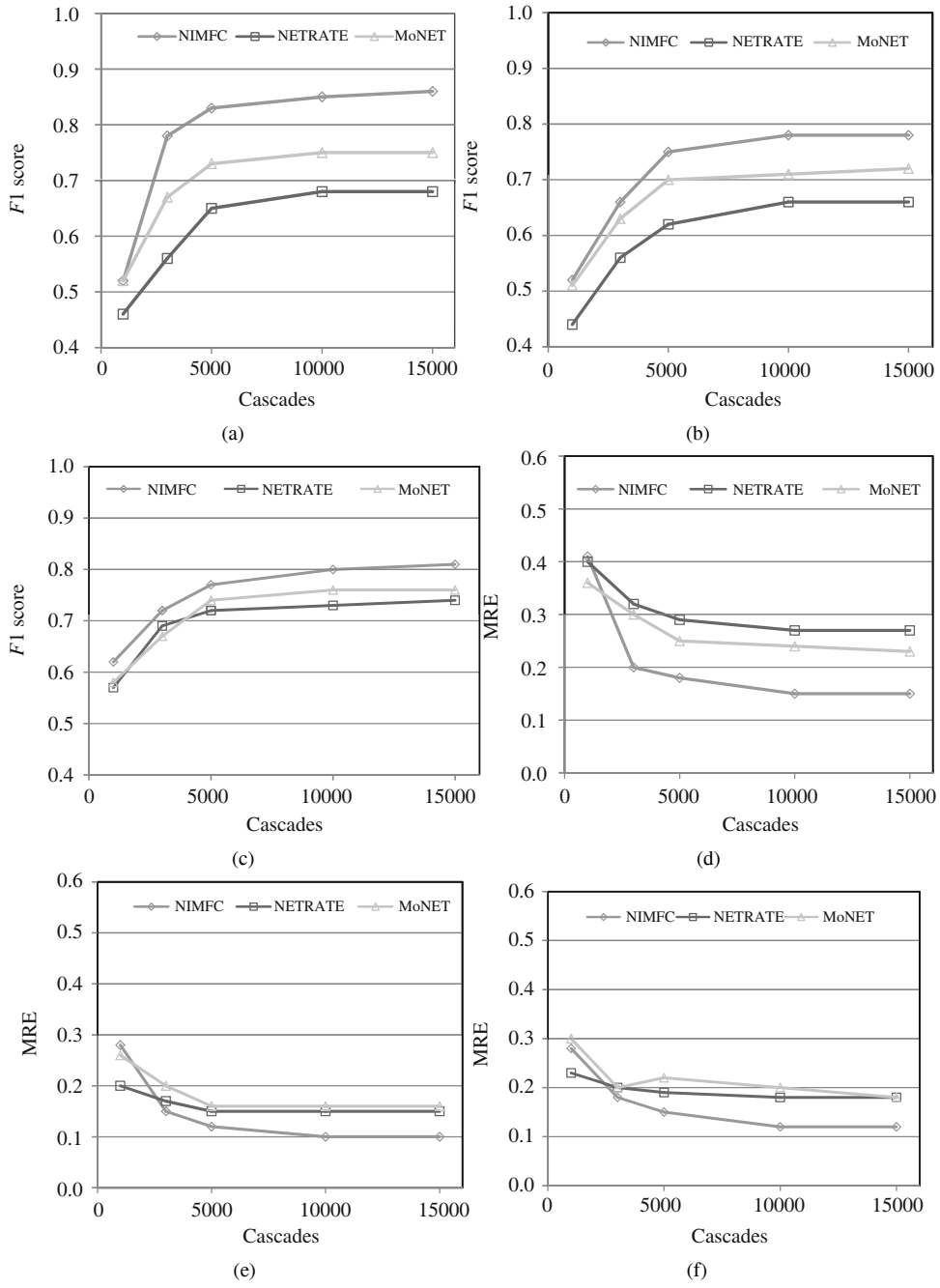


Figure 4 (a) $F1$ scores of network inferring for core–periphery structure network; (b) $F1$ scores of network inferring for hierarchy structure network; (c) $F1$ scores of network inferring for random network; (d) MREs of transmission rate inferring for core–periphery structure network; (e) MREs of transmission rate inferring for hierarchy structure network; (f) MREs of transmission rate inferring for random network.

baselines, even with a worse performance. As cascade number increases, the performance improvement of NIMFC over NETARTE and MoNET is augmented, and after cascade number increases to 10000, the improvement gets steady. The reason behind is with a large number of cascades, NIMFC can model node topic distribution and node CA feature more accurately, while with sparse data set, such additional information may be noisy and useless. We see that when the cascades number is larger than 10000, the MRE of NIMFC can reduce to below 15% for all three types of networks. That means our model can give an evaluation of transmission rate with a mean accuracy exceeding 85%.

Running time of NIMFC. We run the inference problem on a DELL server with 2 Xeon 2.26 GHz

Table 2 Performance comparison in terms of network structure inferring

Model	Recall	Precision	$F1$ score	Log-likelihood (on test data)
NATERATE	0.62	0.59	0.60	−4.6
MoNATE	0.68	0.61	0.64	−4.1
NIMFC	0.85	0.82	0.83	−2.2

CPUs and 16 GB memory. The main process in Algorithm 1 contains two parts: precomputation of π_{ji}^c and iterations of coordinate descent procedure. For a network of 2000 nodes, the precomputation of π_{ji}^c can be completed in no more than 1 min. The running time of coordinate descent procedure depends on the cascades number and cascades length, and in our experiment the average time is about 2 min. Note that, here we only use one core of the CPU. For larger scale networks, because the N subproblems are independent, we can speed up the optimization procedure by running it on multicores in a parallel style.

6.2 Experiment on real data

Data description. We crawled a subset of users from Sina Weibo, including their profiles and messages in a breadth-first way from March of 2013 to March of 2014. Then, a directed graph is built with each node representing a user and each edge representing a following relationship. The network has 61605 nodes and 1631228 edges. For cascade extraction, we characterize a topic by one hashtag or some similar hashtags. The forwarding of messages, including hashtags that are related to the same topic generalizes a cascade. For example, hashtags like #Ukraine riots# or #Ukraine crisis# represent riots breaking out in Ukraine in January 2014, while hashtag like #Andrew Ng joining Baidu# represents the topic of Andrew Ng, the founder of "Google Brain", joining Baidu. Note that topics like "Ukraine crisis" are more popular and discussed by many people, while topics like #Andrew Ng joining Baidu# may be more "local" and "specific". We prefer the latter when using them to infer the network structure. Another way for extracting cascade is to use Uniform Resoure Locator (URL) in Weibo messages. Finally, we extract 10600 cascades, and the average size of these cascades is 32.4. Totally, 35065 unique users participate in at least one cascade.

Metrics. we choose the following evaluation metrics. The first is $F1$ score which is described before. The other is average log-likelihood of cascades in test data set. Because in real data set, the ground truth of transmission rate is unknown to us, we use this measure to test how well the transmission rates inferred from the train dataset can fit the test dataset. To test the performance of pathway tracking of our model, we designed the third measure, that is, average accuracy of inferred cascade pathways. For a cascade c, assuming the number of nodes with right inferred parents is $\mathrm{TP}(c)$, and the number of nodes within c is $|c|$, then accuracy of c is $\mathrm{TP}(c)/|c|$. For NETRATE model and MoNET model, because they are not explicitly proposed to be used in pathway tracking, for the need of comparison, we use similar methods like (20) to infer the pathways.

Experiment results. First, we use all the data set as input to infer possible edges in real network. We get the best $F1$ score for each model and at the same time we record the recall and precision values, as shown in Table 2. Table 2 displays the results for the three models.

First, we compare the $F1$ scores. This measure is to test the performance of recovery of underlying network structure. The NETRATE model performs worst with a score of 0.60 and the MoNET model makes a better result with a score of 0.64. Our model gives the best result of 0.83, achieving a 38.3% relative improvement over NETRATE and a 29.7% relative improvement over MoNET. At this point, the recall value is 0.85 and the precision value is 0.82. Considering we are using a sparse and noisy data set with only 10600 cascades, we think this a remarkable result.

Similarly, from Table 2 we can see the results of log-likelihood of test data. NETRATE and MoNET get log-likelihood values of -4.6 and -4.1, respectively, while NIMFC gets a result of -2.2, which is about two orders of magnitude higher than that of NETRATE and MoNET. That means with node attributes, cascade structure, and information content merged into our model, we can infer the transmission rate for pairwise nodes more accurately.

For the second metrics, we use fourfold cross-validation approach, that is, to use 3/4 of the data set as

Table 3 Performance comparison in terms of information pathway tracking

Cascade length	< 40	40–100	> 100	Average accuracy
NATERATE	0.38	0.29	0.26	0.325
MoNATE	0.42	0.33	0.30	0.364
NIMFC	0.76	0.65	0.57	0.683

training data to learn the transmission rate of each edge and calculate the log-likelihood of the left 1/4 data set. The results of information pathway tracking are illustrated in Table 3. We list the accuracy for different ranges of cascades length and finally the weighted average accuracy.

First, one can see that as the cascades length increases, it gets harder to track the information pathway accurately for all three models. For cascades with length below 40, the results of three models are 0.38, 0.42, and 0.76, respectively. For cascades with length above 100, the results of three models reduce to 0.26, 0.30 and 0.57 respectively.

Second, baseline models perform poorly on the data set. They get weighted average accuracy of 0.325 and 0.364, respectively, while NIMFC yields a value of 0.683, which is 110.2% higher than that of NETRATE or 87.6% higher than that of MoNET. As explained before, this is mainly due to the fact that in real social networks like Weibo, node attributes and information content play an important role in information diffusion process. NETRATE only focuses on temporal features when modeling node pairwise transmission behavior, resulting in a situation something like "under-fitting". While MoNET takes into consideration the influence of node attributes, it ignores the factor of information content.

Third, the average accuracy of NIMFC model indicates that besides the correctly inferred pathways, there are still considerable information pathways that are not correctly inferred. For the pathways inferred incorrectly, a possible explanation is that in real world, user's behaviors of choosing to forward or not to forward a message, or choosing to forward what message, are often with some randomness and the information diffusion process may also be influenced by external factors. These factors increase the complexity when inferring information pathways, and they will be covered in our future work.

In summary, the above values in Tables 2 and 3 clearly show that the topological features of cascades, node attributes, and information content jointly play a crucial role in the information diffusion process. This fully justifies our will to take into account these heterogeneous features when inferring the network structure and tracking information pathways.

7 Conclusion and future work

In this paper, we developed NIMFC, a novel probabilistic model for social network structure and transmission rate inferring. Our model extracts multidimensional features from observed information cascades, including temporal, and topological features of cascades, node attributes and information content feature. We extend NIMFC model to track information diffusion pathways in social networks, which can give us a deeper insight into the dynamics of information diffusion. We compare the performance of NIMFC with state-of-the-art baselines on synthesis and real-world data set, and it turns out that by incorporating heterogeneous features of cascades, NIMFC can improve performance both in terms of network structure recovery and transmission rate inferring. These works may improve our understanding of the dynamics of information diffusion and serve as a base for managing propagation in social networks.

The proposed model is an open framework and some future works may include exploiting new features of nodes (e.g., location and education background) and information content (e.g., sentiment feature) for better inferring, taking into consideration external influence [26] outside the diffusion network to build a more realistic model and applying the inferred results to influence maximization, rumor detection, epidemic immunization, and others.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 61271252, 61202482), Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20124307110014),

and fund from the State Key Laboratory of Mathematical Engineering and Advanced Computing of China. We would like to thank the anonymous referees for their help in improving this paper.

References

1 Xiong X B, Zhou G, Huang Y Z, et al. Dynamic evolution of collective emotions in social networks: a case study of Sinaweibo. Sci China Inf Sci, 2013, 56: 078101
2 Khondker H H. Role of the new media in the Arab Spring. Globalizations, 2011, 8: 675–679
3 Tumasjan A, Sprenger T O, Sandner P G, et al. Predicting elections with twitter: what 140 characters reveal about political sentiment. In: Proceedings of the 4th International AAAI Conference on Weblogs and Social Media. Menlo Park: AAAI, 2010. 178–185
4 Kempe D, Kleinberg J, Tardos E. Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACMSIGKDD International Conference on Knowledge Discovery and DataMining. New York: ACM, 2003. 137–146
5 Wang Y, Huang W J, Zong L, et al. Influence maximization with limit cost in social network. Sci China Inf Sci, 2013, 56: 078102
6 Nguyen N P, Yan G, Thai M T. Analysis of misinformation spread containment in online social networks. Comput Netw, 2013, 57: 2133–2146
7 Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2007. 420–429
8 Gomez-Rodriguez M, Balduzzi D, Scholkopf B. Uncovering the temporal dynamics of diffusion networks. In: Proceedings of the 28th International Conference on Machine Learning. New York: ACM, 2011. 561–568
9 Gomez-Rodriguez M, Leskovec J, Krause A. Inferring networks of diffusion and influence. In: Proceedings of the 16th ACMSIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM, 2010. 1019–1028
10 Myers S, Leskovec J. On the convexity of latent social network inference. In: Proceedings of the 24th Annual Conference on Neural Information Processing Systems. Cambridge: MIT Press, 2010. 1741–1749
11 Netrapalli P, Sanghavi S. Learning the graph of epidemic cascades. ACM SIGMETRICS Perform Eval Rev, 2012, 40: 211–222
12 Snowsill T, Fyson N, de Bie T, et al. Refining causality: who copied from whom? In: Proceedings of the 17th ACMSIGKDD International Conference on Knowledge Discovery in Data Mining. New York: ACM, 2011. 466–474
13 Wang L, Ermon S, Hopcroft J E. Feature-enhanced probabilistic models for diffusion network inference. Lect Notes Comput Sci, 2012, 7524: 499–514
14 Du N, Song L, Woo H, et al. Uncover topic-sensitive information diffusion networks. In: Proceedings of the 16th International Conference on Artificial Intelligence and Statistics. Cambridge: MIT Press, 2013. 229–237
15 Gomez-Rodriguez M, Leskovec J, Scholkopf B. Structure and dynamics of information pathways in online media. In: Proceedings of the 6th ACM International Conference on Web Search and Data Mining. New York: ACM, 2013. 23–32
16 Eagle N, Pentland A S, Lazer D. From the cover: inferring friendship network structure by using mobile phone data. In: Proceedings of the National Academy of Sciences. Washington DC: National Academies Press, 2009. 15274–15278
17 Kolar M, Song L, Ahmed A, et al. Estimating time-varying networks. Ann Appl Stat, 2010, 4: 94–123
18 Yang S, Zha H. Mixture of mutually exciting processes for viral diffusion. In: Proceedings of the 30th International Conference on Machine Learning. New York: ACM, 2013. 1–9
19 Aalen O O, Borgan O, Gjessing H K. Survival and event history analysis: a process point of view. Berlin/Heidelberg: Springer-Verlag, 2008. 5–36
20 Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation. J Mach Learn Res, 2003, 3: 993–1022
21 Crandall D, Cosley D, Huttenlocher D, et al. Feedback effects between similarity and social influence in online communities. In: Proceeding of the 14th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2008. 160–168
22 McPherson M, Smith-Lovin L, Cook J M. Birds of a feather: homophily in social networks. Annl Rev Sociol, 2001, 27: 415–444
23 Wallinga J, Teunis P. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. Amer J Epidemiol, 2004, 160: 509–516
24 Tseng P, Mangasarian C O L. Convergence of a block coordinate descent method for nondifferentiable minimization. J Opt Theory Appl, 2001, 109: 475–494
25 Leskovec J, Chakrabarti D, Kleinberg J, et al. Kronecker graphs: an approach to modeling networks. J Mach Learn Res, 2010, 11: 985–1042
26 Myers S, Zhu C, Leskovec J. Information diffusion and external influence in networks. In: Proceeding of the 18th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2012. 33–41