Feature-Enhanced Probabilistic Models for Diffusion Network Inference

Liaoruo Wang*, Stefano Ermon*, and John E. Hopcroft

Department of Computer Science
Cornell University, Ithaca, NY 14853, USA
{lwang, ermonste, jeh}@cs.cornell.edu

Abstract. Cascading processes, such as disease contagion, viral marketing, and information diffusion, are a pervasive phenomenon in many types of networks. The problem of devising intervention strategies to facilitate or inhibit such processes has recently received considerable attention. However, a major challenge is that the underlying network is often unknown. In this paper, we revisit the problem of inferring latent network structure given observations from a diffusion process, such as the spread of trending topics in social media. We define a family of novel probabilistic models that can explain recurrent cascading behavior, and take into account not only the time differences between events but also a richer set of additional features. We show that MAP inference is tractable and can therefore scale to very large real-world networks. Further, we demonstrate the effectiveness of our approach by inferring the underlying network structure of a subset of the popular Twitter following network by analyzing the topics of a large number of messages posted by users over a 10-month period. Experimental results show that our models accurately recover the links of the Twitter network, and significantly improve the performance over previous models based entirely on time.

1 Introduction

Cascading processes, such as the spread of a computer virus or an infectious disease, are a pervasive phenomenon in many networks. Diffusion and propagation processes have been studied in a broad range of disciplines, such as information diffusion [1–4], social networks [5, 6], viral marketing [7, 8], epidemiology [9], and ecology [10]. In previous work, researchers have mostly focused on a number of optimization problems derived from cascading processes, where the goal is to devise intervention strategies to either maximize (e.g., viral marketing) or minimize (e.g., network interdiction, vaccination programs) the propagation. However, these studies often assume that the underlying network is known to the observer, which in practice is not true in many situations.

In this paper, we revisit the problem of inferring latent network structure given observations of a diffusion process. For example, by observing a disease epidemic, we want to infer the underlying social contact network, or by observing the spread of trending topics, we want to estimate the connectivity of the social media. Fig. 1 illustrates a case of information diffusion in the popular Twitter network. The nodes

* indicates equal contribution.

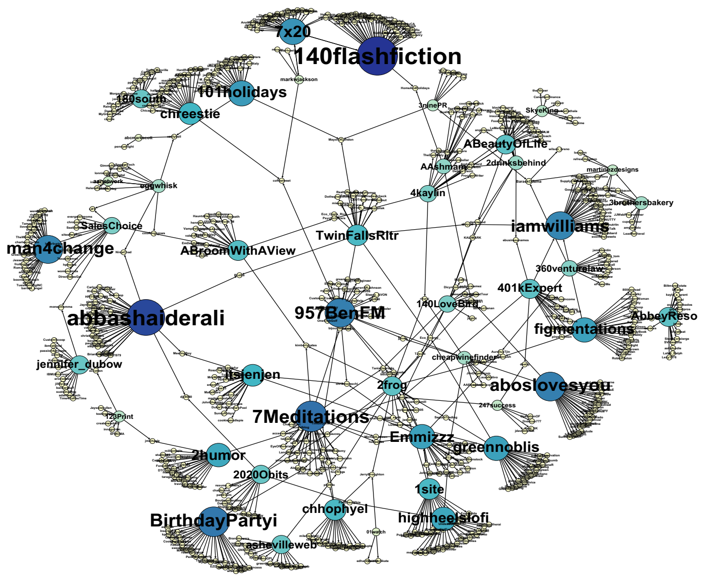

Fig. 1. Information diffusion in the Twitter network (see PDF for colored version).

represent a subset of the Twitter users that have posted about a common trending topic, and the directed edges represent the "following" relation between the users. There is a clear pattern in the figure. The bigger and darker nodes, followed by the smaller and lighter nodes, form the hubs of the diffusion process. By looking at the time-stamps of the messages and at the underlying network structure, we observe that most information flows initiate at a hub node and spread across the network to reach other hub nodes and their followers. However, it is non-trivial to come up with such a picture simply by looking at the time-stamps of the messages, since without knowing the underlying network structure, we cannot decide from whom a node copied the information from. Intuitively, messages carry implicit information about the social relations among users. For instance, users who repeatedly post messages about the same topic within a short period of time, are more likely to be connected. Thus, a motivating application of this paper is to what extent we can estimate the relations in social networks by analyzing the messages published by users over time.

This type of latent network inference problem based on the time-stamps of infection (or, information-reproduction) events has received increasing interest over the past few years [1–3]. Previous work was largely based on two major assumptions: 1) the diffusion process is causal (i.e., not affected by events in the future), and 2) infection events closer in time are more likely to be causally related (e.g., according to an exponential, Rayleigh, or power-law distribution). While the causality assumption is indeed crucial and always satisfied in practice, we realize that there are many other factors that can be highly informative as far as the causality relations are concerned. For example, the time-stamps at which two users publish their tweets are important to decide whether they are related, but other factors such as the language or the content of the messages can be as important. Even if the two messages are close in time, they are unlikely to be related if the messages are written in different languages. Further, previous models in the literature are mostly focused on monotonic processes, while real-world processes are often recurrent. For instance, it is very common for one user to post about the same topic multiple times on Twitter, or purchase the same item regularly on Amazon.

Contributions. Motivated by these challenges, we define a family of novel probabilistic models that generalize previous models based solely on time. We propose a primary approach MONET that can handle recurrent diffusion processes. Further, we consider a richer set of additional features for infection events, defining novel feature-enhanced models that can better explain the observed data. With distributed optimization and convex objective functions, we can efficiently solve the problem of inferring the most probable latent network structure. Using additional features such as the languages of the messages and Jaccard indexes between the messages, we can accurately recover the links of the Twitter network by analyzing the topics of a large number of messages posted by a subset of the Twitter users over a 10-month period. Experimental results show that our models significantly improve the accuracy of the estimates over previous models by as much as 78.7%.

2 Problem Definition

We consider a diffusion process across a network represented by a directed, weighted graph $G = (V, E)$. Let $\mathbf{A} = \{\alpha_{jk} | j, k \in V, j \neq k\}$ be the adjacency matrix of weights. A directed edge (j, k) has weight $\alpha_{jk} \geqslant 0$ that denotes the pairwise transmission rate from node j to node k. For example, in the case of an infectious disease spreading through a population, V represents a group of individuals and E represents the strength of the social contacts among them. In the case of an invasive species colonizing a new territory, V represents patches of land and E represents the connectivity between them. We assume that the diffusion process is stochastic but *causal*, that is, it depends on the past history but not on the future. Specifically, we consider a diffusion process that starts with one or more nodes, and spreads across the network subject to an independent local probabilistic model of "infection", where a node infects its neighbors independently of the status of other nodes in the network [5].

When studying such diffusion processes, the underlying network is often unknown (latent). However, we assume that one can observe a set of *cascades* of "infection" (or, information-reproduction) events. A *cascade* is a sequence of infection events

$$\pi = \{(v_0, t_0), \cdots, (v_N, t_N)\}$$

during a given time interval T, where $v_i \in V$ is a node that becomes infected at time t_i. T is the *horizon* of cascade π. Note that different cascades may have different horizons. For example, in the Twitter network, each cascade corresponds to a trending topic, and we have an entry (v_i, t_i) for each tweet posted by user v_i at time t_i. Given a probabilistic model $P(\pi|G)$ that gives the probability of observing a certain cascade π when the underlying network is G, the problem of inferring the latent network structure from

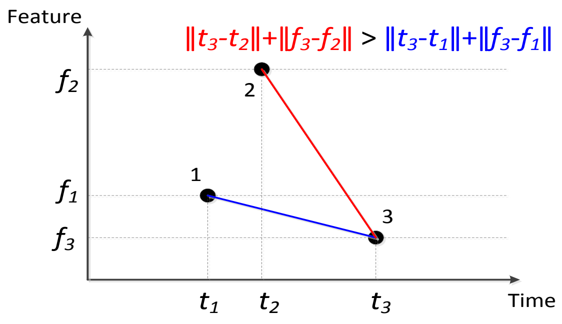

Fig. 2. Feature-enhanced probabilistic model.

observed cascades has received considerable attention [1–3]. It is usually assumed that V is known but E is unknown, and the goal is to find a (weighted) graph

$$G^* = \arg\max_G \prod_{c=1}^{M} P(\pi^c|G)$$

that maximizes the probability of observing cascades $\pi^1, \cdots, \pi^M$, which are assumed to be i.i.d. (independent, identically distributed) realizations of the underlying diffusion process.

The building block to define $P(\cdot)$ is the likelihood function $f(t_k|t_j;\alpha_{jk})$ that gives the probability density that node v_j infected at time t_j infects node v_k at time t_k (see below for more details). Such models are centered on the time differences between the infection events of a cascade, and exploit the causal nature of the diffusion process by setting the likelihood to zero whenever $t_k - t_j < 0$. Further, they assume that events closer in time are more likely to be causally related. For instance, if node v_k became infected shortly after node v_j was infected, this is considered as an indication that the two events are causally related (i.e., v_k was infected by v_j). These time-based models are the foundation of our work.

While time is indeed a crucial element of the network inference problem, in practical applications, observations of a diffusion process often carry additional key information. For example, the diagnosis of an infection often comes with additional information about the specific strain. When a topic or a rumor spreads through a social network, one can also observe the context in which it appears. This motivates our definition of a *generalized cascade*

$$\pi_g = \{(v_0, t_0, \mathbf{f}_0), \cdots, (v_N, t_N, \mathbf{f}_N)\} \tag{1}$$

where $v_i \in V$ is a node infected at time t_i, and $\mathbf{f}_i \in \mathcal{F}$ is a feature vector describing the additional information available for the i-th infection event. Using the additional information contained in a generalized cascade, we can define a generalized feature-enhanced probabilistic model where the probability of a transmission event depends not only on the time differences, but also on the additional features. Specifically, we use a probability density function $f(t_k, \mathbf{f}_k|t_j, \mathbf{f}_j;\alpha_{jk})$ as a building block, which depends causally on the relative time difference $t_k - t_j$ as well as on the additional features $\mathbf{f}_k$ and $\mathbf{f}_j$. Fig. 2 gives an example of the difference between previous models that are based solely on time and a case of our feature-enhanced models where $f(t_k, \mathbf{f}_k | t_j, \mathbf{f}_j; \alpha_{jk})$ depends on $||t_k - t_j|| + ||\mathbf{f}_k - \mathbf{f}_j||$. Node 3 is considered to be more related to node 1 than node 2 by our feature-enhanced models, while it is determined to be more related to node 2 by models based only on time.

Furthermore, previous models are focused on monotonic diffusion processes, while most real-world processes are recurrent. For example, it is common for one user to post about the same topic multiple times on Twitter, or purchase the same item multiple times on Amazon. Repeated posts of the same topic show a higher level of interest in that topic, and exchanged posts of the same topic between a group of nodes also show a higher level of connectivity in that group. We take these factors into account in our feature-enhanced models and assign respective reward/penalty to each scenario.

There are two different ways of modeling a diffusion process where nodes can be infected multiple times in one cascade. The first model considers an infection event as the result of all previous events, and thus we call it *non-splitting*. By contrast, the second model considers an infection event of a node as the result of all previous events *up to its last infection*. This model is memoryless and thus we call it *splitting*. We mainly focus on the non-splitting model in this section, but the results can be extended to the splitting case. We will later present experimental results in Section 4 for both non-splitting and splitting models.

2.1 Generalized Cascade Model

We first recall some standard notation from previous literature, and then define our feature-enhanced models based on generalized cascades.

Recap. Recall the standard notation from [1] and [11]. Given that node j was infected at time t_j, the *survival function* of edge (j, k) is the probability that, by time t_k, node k was not infected by node j. That is,

$$S\left(t_k | t_j; \alpha_{jk}\right) = 1 - F\left(t_k | t_j; \alpha_{jk}\right), \tag{2}$$

where α_{jk} denotes the transmission rate from node j to node k, and $F\left(t_k | t_j; \alpha_{jk}\right)$ is the cumulative distribution function. Further, the *hazard function* (or, instantaneous infection rate) of edge (j, k) is given by

$$H\left(t_k | t_j; \alpha_{jk}\right) = \frac{f\left(t_k | t_j; \alpha_{jk}\right)}{S\left(t_k | t_j; \alpha_{jk}\right)}, \tag{3}$$

where

$$f\left(t_k | t_j; \alpha_{jk}\right) = \left. \frac{d}{dt} F\left(t | t_j; \alpha_{jk}\right) \right|_{t_k}$$

is the likelihood function. Table 1 shows the survival and hazard functions based on the exponential, Rayleigh, and power-law distribution.

Multiple Occurrences. Real-world diffusion processes are often recurrent, that is, we often observe multiple occurrences of the same node in one cascade. With multiple

Table 1. Parametric Models [1].

Model	**Likelihood Function** $f\left(t_k \| t_j; \alpha_{jk}\right)$	**Survival Function** $S\left(t_k \| t_j; \alpha_{jk}\right)$	**Hazard Function** $H\left(t_k \| t_j; \alpha_{jk}\right)$
Exponential	$\begin{cases} \alpha_{jk} e^{-\alpha_{jk}\left(t_k - t_j\right)} & \text{if } t_j < t_k \\ 0 & \text{otherwise} \end{cases}$	$e^{-\alpha_{jk}\left(t_k - t_j\right)}$	α_{jk}
Rayleigh	$\begin{cases} \alpha_{jk}\left(t_k - t_j\right) e^{-\alpha_{jk}\frac{\left(t_k - t_j\right)^2}{2}} & \text{if } t_j < t_k \\ 0 & \text{otherwise} \end{cases}$	$e^{-\alpha_{jk}\frac{\left(t_k - t_j\right)^2}{2}}$	$\alpha_{jk}\left(t_k - t_j\right)$
Power Law	$\begin{cases} \frac{\alpha_{jk}}{\delta}\left(\frac{t_k - t_j}{\delta}\right)^{-1-\alpha_{jk}} & \text{if } t_j < t_k - \delta \\ 0 & \text{otherwise} \end{cases}$	$\left(\frac{t_k - t_j}{\delta}\right)^{-\alpha_{jk}}$	$\frac{\alpha_{jk}}{t_k - t_j}$

occurrences of node k and node j, the survival function for the non-splitting case is

$$\begin{aligned} S\left(t_k|t_j;\alpha_{jk}\right) &= \prod_{k:t_k^{(1)}\leqslant T^c} \prod_{1\leqslant i\leqslant N_k^c} \prod_{j\neq k:t_j^{(1)}<t_k^{(i)}} S\left(t_k^{(i)}|t_j;\alpha_{jk}\right) \\ &= \prod_{k:t_k^{(1)}\leqslant T^c} \prod_{1\leqslant i\leqslant N_k^c} \prod_{j\neq k:t_j^{(1)}<t_k^{(i)}} \prod_{1\leqslant \ell\leqslant N_j^c(t_k^{(i)})} S\left(t_k^{(i)}|t_j^{(\ell)};\alpha_{jk}\right), \end{aligned}$$

where T^c is the horizon of cascade π^c, and $t_k^{(i)}, i \in \{0, \cdots, N_k^c, N_k^c + 1\}$ denote the time-stamps of node k infections in cascade π^c. We assign two special time-stamps for every node: $t_k^{(0)} = 0$ and $t_k^{(N_k^c+1)} = T^c$. N_k^c denotes the number of node k infections in cascade π^c. $N_j^c(t_k^{(i)})$ denotes the number of node j infections before the i-th infection of node k. Similarly, the hazard function is given by

$$H\left(t_k^{(i)}|t_j^{(\ell)};\alpha_{j,k}\right) = \frac{f\left(t_k^{(i)}|t_j^{(\ell)};\alpha_{j,k}\right)}{S\left(t_k^{(i)}|t_j^{(\ell)};\alpha_{j,k}\right)}.$$

Additional Features. Consider two feature vectors $\mathbf{f}_k, \mathbf{f}_j \in \mathcal{F}$ associated with node $k, j \in V$ in a cascade. Let $d\left(\mathbf{f}_k, \mathbf{f}_j\right)$ denote the distance between the two feature vectors. We include an extra term $e^{-d(\mathbf{f}_k,\mathbf{f}_j)}$ in the likelihood function to reflect this distance factor. For example, given an exponential distribution, we have

$$f\left(t_k, \mathbf{f}_k|t_j, \mathbf{f}_j;\alpha_{jk}\right) = \begin{cases} \gamma\alpha_{jk} e^{-d\left(\mathbf{f}_k,\mathbf{f}_j\right)} e^{-\alpha_{jk}\left(t_k - t_j\right)}, & \text{if } t_j < t_k \\ 0, & \text{otherwise} \end{cases}$$

where γ is a normalization constant. Thus, the survival function is given by

$$\begin{aligned} S\left(t_k|t_j, \mathbf{f}_j;\alpha_{jk}\right) &= 1 - F\left(t_k|t_j, \mathbf{f}_j;\alpha_{jk}\right) = 1 - \int_{\mathcal{F}}\int_{t_j}^{t_k} f\left(t, \mathbf{f}|t_j, \mathbf{f}_j;\alpha_{jk}\right) \mathrm{d}t\, \mathrm{d}\mathbf{f} \\ &= e^{-\alpha_{jk}\left(t_k - t_j\right)} \int_{\mathcal{F}} \gamma e^{-d\left(\mathbf{f},\mathbf{f}_j\right)} \mathrm{d}\mathbf{f} = e^{-\alpha_{jk}\left(t_k - t_j\right)}. \end{aligned} \tag{4}$$

Then, the hazard function is

$$H\left(t_k, \mathbf{f}_k|t_j, \mathbf{f}_j;\alpha_{jk}\right) = \frac{f\left(t_k, \mathbf{f}_k|t_j, \mathbf{f}_j;\alpha_{jk}\right)}{S\left(t_k|t_j, \mathbf{f}_j;\alpha_{jk}\right)} = \begin{cases} \gamma\alpha_{jk} e^{-d\left(\mathbf{f}_k,\mathbf{f}_j\right)}, & \text{if } t_j < t_k \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

2.2 MAP Inference

Given independent cascades, the likelihood of a set of cascades $\{\pi_g^1, \cdots, \pi_g^M\}$ is the product of the likelihood of each cascade:

$$\prod_{1 \leqslant c \leqslant M} f\left(\pi_g^c; \mathbf{A}\right), \tag{6}$$

where $\mathbf{A} = \{\alpha_{jk} | j, k \in V, j \neq k\}$ is a weighted adjacency matrix of transmission rates. Given a cascade π_g^c, the probability that node k was not infected by time T^c is the product of the survival functions of the infected nodes. The formulation can be extended according to the generalized cascade model discussed in Section 2.1. For example, if a node was infected multiple times during the observation window, this repeated lack of ability to infect node k should also be considered. Specifically, the probability that node k was not infected by time T^c is

$$\prod_{j:t_j^{(1)} \leqslant T^c} \prod_{1 \leqslant i \leqslant N_j^c} S\left(T^c | t_j^{(i)}, \mathbf{f}_j^{(i)}; \alpha_{jk}\right).$$

Given the parents of the infected nodes, infections are assumed to be conditionally independent. Thus, the likelihood of the observed cascade π_g^c is

$$f\left(\pi_g^c; \mathbf{A}\right) = \prod_{j:t_j^{(1)} \leqslant T^c} \prod_{1 \leqslant i \leqslant N_j^c} f\left(t_j^{(i)}, \mathbf{f}_j^{(i)} | \pi_g^c \backslash \left(j, t_j^{(i)}, \mathbf{f}_j^{(i)}\right); \mathbf{A}\right).$$

Given the i-th infection of node k, the likelihood of node j being its first parent is

$$\begin{aligned} f\left(t_k^{(i)}, \mathbf{f}_k^{(i)} | t_j, \mathbf{f}_j; \mathbf{A}\right) = & \prod_{s \neq j: t_s^{(1)} < t_k^{(i)}} \prod_{1 \leqslant p \leqslant N_s^c(t_k^{(i)})} S\left(t_k^{(i)} | t_s^{(p)}, \mathbf{f}_s^{(p)}; \alpha_{s,k}\right) \\ & \times \sum_{1 \leqslant \ell \leqslant N_j^c(t_k^{(i)})} f\left(t_k^{(i)}, \mathbf{f}_k^{(i)} | t_j^{(\ell)}, \mathbf{f}_j^{(\ell)}; \alpha_{j,k}\right) \prod_{q \neq \ell} S\left(t_k^{(i)} | t_j^{(q)}, \mathbf{f}_j^{(q)}; \alpha_{j,k}\right). \end{aligned}$$

Thus, the likelihood of the observed cascade π_g^c is

$$f\left(\pi_g^c; \mathbf{A}\right) = \prod_{k:t_k^{(1)} \leqslant T^c} \prod_{1 \leqslant i \leqslant N_k^c} \left(\sum_{j:t_j^{(1)} < t_k^{(i)}} f\left(t_k^{(i)}, \mathbf{f}_k^{(i)} | t_j, \mathbf{f}_j; \mathbf{A}\right) \right).$$

Combine the two equations above and include the condition $s = j$, we have

$$\begin{aligned} f\left(\pi_g^c; \mathbf{A}\right) = \prod_{k:t_k^{(1)} \leqslant T^c} \prod_{1 \leqslant i \leqslant N_k^c} \Bigg(& \prod_{s:t_s^{(1)} < t_k^{(i)}} S\left(t_k^{(i)} | t_s, \mathbf{f}_s; \alpha_{s,k}\right) \times \\ & \sum_{j:t_j^{(1)} < t_k^{(i)}} \sum_{1 \leqslant \ell \leqslant N_j^c(t_k^{(i)})} \frac{f\left(t_k^{(i)}, \mathbf{f}_k^{(i)} | t_j^{(\ell)}, \mathbf{f}_j^{(\ell)}; \alpha_{j,k}\right)}{S\left(t_k^{(i)} | t_j^{(\ell)}, \mathbf{f}_j^{(\ell)}; \alpha_{j,k}\right)} \Bigg). \end{aligned}$$

Add the information that some nodes were never infected during the horizon T^c, and then,

$$f\left(\pi_g^c;\mathbf{A}\right)=\prod_{k:t_k^{(1)}\leqslant T^c}\prod_{1\leqslant i\leqslant N_k^c}\prod_{t_m>T^c}S\left(T^c|t_k^{(i)},\mathbf{f}_k^{(i)};\alpha_{i,m}\right)\times$$
$$\prod_{s:t_s^{(1)}<t_k^{(i)}}S\left(t_k^{(i)}|t_s,\mathbf{f}_s;\alpha_{s,k}\right)\times\sum_{j:t_j^{(1)}<t_k^{(i)}}\sum_{1\leqslant\ell\leqslant N_j^c(t_k^{(i)})}H\left(t_k^{(i)},\mathbf{f}_k^{(i)}|t_j^{(\ell)},\mathbf{f}_j^{(\ell)};\alpha_{j,k}\right). \tag{7}$$

Eq. (7) gives the likelihood of cascade π_g^c for the non-splitting model. However, for the splitting case, this likelihood is given by

$$\begin{aligned}f\left(\pi_g^c;\mathbf{A}\right)=&\prod_{k:t_k^{(1)}\leqslant T^c}\prod_{1\leqslant i\leqslant N_k^c}\prod_{t_m>T^c}S\left(T^c|t_k^{(i)},\mathbf{f}_k^{(i)};\alpha_{i,m}\right)\times\\&\prod_{s:t_s^{(1)}<t_k^{(i)}}\prod_{N_s^c(t_k^{(i-1)})<p\leqslant N_s^c(t_k^{(i)})}S\left(t_k^{(i)}|t_s^{(p)},\mathbf{f}_s^{(p)};\alpha_{s,k}\right)\times\\&\sum_{j:t_j^{(1)}<t_k^{(i)}}\sum_{N_j^c(t_k^{(i-1)})\leqslant\ell\leqslant N_j^c(t_k^{(i)})}H\left(t_k^{(i)},\mathbf{f}_k^{(i)}|t_j^{(\ell)},\mathbf{f}_j^{(\ell)};\alpha_{j,k}\right).\end{aligned} \tag{8}$$

Eq. (8) is similar to Eq. (7) except that we only consider the segment between the $(i-1)$-th and i-th occurrence of node k for the survival and hazard function.

Problem Definition. Our goal is to infer the connectivity and estimate the infection rate α_{jk} for each pair of nodes (j,k) such that the likelihood of observed cascades $\{\pi_g^1,\cdots,\pi_g^M\}$ is maximized. Specifically,

$$\begin{aligned}&\text{minimize}_{\mathbf{A}} && -\sum_{1\leqslant c\leqslant M}\log f\left(\pi_g^c;\mathbf{A}\right)\\&\text{subject to} && \alpha_{jk}\geqslant 0,\ j,k\in V,j\neq k.\end{aligned} \tag{9}$$

where $\mathbf{A}=\{\alpha_{jk}|j,k\in V,j\neq k\}$ are the variables. The inferred edges of the network are those pairs of nodes with infection rate $\alpha_{jk}>0$.

3 Proposed Approach: MONET

In this section, we discuss the properties of the optimization problem arising from the MAP inference task in our feature-enhanced probabilistic models defined in Section 2. By Eq. (6) and Eq. (7), the log-likelihood of cascades $\{\pi_g^1,\cdots,\pi_g^M\}$ is

$$L\left(\{\pi_g^1,\cdots,\pi_g^M\};\mathbf{A}\right)=\sum_{1\leqslant c\leqslant M}\Phi_1(\pi_g^c;\mathbf{A})+\Phi_2(\pi_g^c;\mathbf{A})+\Phi_3(\pi_g^c;\mathbf{A}), \tag{10}$$

where for each cascade $\pi_g^c \in \{\pi_g^1, \cdots, \pi_g^M\}$,

$$\Phi_1(\pi_g^c; A) = \sum_{k:t_k^{(1)} \leqslant T^c} \sum_{1 \leqslant i \leqslant N_k^c} \sum_{s:t_s^{(1)} < t_k^{(i)}} \log S\left(t_k^{(i)} | t_s, \mathbf{f}_s; \alpha_{s,k}\right),$$

$$\Phi_2(\pi_g^c; A) = \sum_{k:t_k^{(1)} \leqslant T^c} \sum_{1 \leqslant i \leqslant N_k^c} \log \sum_{j:t_j^{(1)} < t_k^{(i)}} \sum_{1 \leqslant \ell \leqslant N_j^c(t_k^{(i)})} H\left(t_k^{(i)}, \mathbf{f}_k^{(i)} | t_j^{(\ell)}, \mathbf{f}_j^{(\ell)}; \alpha_{j,k}\right),$$

$$\Phi_3(\pi_g^c; A) = \sum_{k:t_k^{(1)} \leqslant T^c} \sum_{1 \leqslant i \leqslant N_k^c} \sum_{t_m^{(1)} > T^c} \log S\left(T^c | t_k^{(i)}, \mathbf{f}_k^{(i)}; \alpha_{k,m}\right).$$

The above equations are a strict generalization of the ones presented in [1], that is, we recover the same formulation where there are no multiple occurrences and we do not consider any additional features other than time. Further, we also generalize several results of the model in [1] to our feature-enriched setting, for both splitting and non-splitting cases. Formally,

Theorem 1. *The following results hold:*

- *Given any distance functions, log-concave survival functions, and concave hazard functions, the problem defined by Eq.* (9) *is convex in* $\mathbf{A}$.
- *The optimization problem defined by Eq.* (9) *is convex for the feature-enhanced models with exponential, Rayleigh, or power law distribution.*
- *The solution to Eq.* (9) *gives a consistent maximum likelihood estimator.*

The proof of Theorem 1 is similar to that of [1], and is omitted for space reasons.

We call our primary approach MONET, which provides non-splitting and splitting solutions for the network inference problem defined by Eq. (9) where nodes can be repeatedly infected.

Analyzing MONET. We discuss some properties of the solution to the optimization problem defined in Eq. (9) for the generalized feature-enhanced models with the exponential, Rayleigh, and power-law distribution. This is equivalent to maximizing the log-likelihood defined in Eq. (10). Clearly, the expression in Eq. (10) depends on the transmission rate α_{jk} and the relative time difference $t_k - t_j$ between each occurrence of node j and node k. Note that it does not depend on the absolute values of the time-stamps. In general, however, Eq. (10) depends on the absolute values of the feature vectors (i.e., it depends not only on the distance between observed feature vectors), due to the normalization constant γ.

As discussed in [1], Φ_1 and Φ_3 encourage sparse solutions by imposing negative weights on $\mathbf{A}$. Specifically, Φ_1 penalizes α_{jk} based on the relative time difference $t_k - t_j$ and Φ_3 penalizes α_{ki} for uninfected node i based on $T^c - t_k$ (i.e., until the horizon cut-off). Note that MONET only infers impossible edges based on 0 transmission rates. Due to finite observation window, the lack of ability to infect some node i within time T^c does not mean it is impossible to infect node i (i.e., there is no edge).

The term Φ_2 emphasizes the intuition that infected nodes must have at least one parent (appearing before them in a cascade) by which they were infected. If this is not ensured, $\Phi_2 = -\infty$ will be negatively unbounded. The additional features used in our models only affect the term Φ_2. Specifically, infected nodes tend to select those that are more similar to them as their parents. Further, in the non-splitting case, only the first occurrence of each node in the cascade is affected by this hard constraint (further occurrences can still be explained by the parent of the first occurrence). However, simply using the first explanation can be too penalizing, and adding more parents might improve the likelihood.

Computational Aspects. As in previous work, we can parallelize the solution to the optimization problem defined in Eq. (9). Given a network with n nodes, this optimization problem has $O(n^2)$ variables, but the objective function can be separated into n independent sub-problems with $O(n)$ variables each. For each node $k = 1, \cdots, n$, we optimize the k-th column of the matrix $\mathbf{A}$ of transmission rates, solving for $(n-1)$ unknown transmission rates $\{\alpha_{jk}\}$ where $j \neq k$. To compute the k-th column, we only require the infection times of the nodes in those cascades where node k appears. Optimal columns are joined to form a globally optimal transmission rate matrix.

If node j never appears before node k in any cascade, we have no evidence to suggest the existence of a directed edge (j, k). That is, α_{jk} only contributes to the non-positive term Φ_2 in Eq. (10). Thus, in every iteration, we set α_{jk} to the optimal value 0 to simplify the objective function $L\left(\{\pi_g^1, \cdots, \pi_g^M\}; \mathbf{A}\right)$.

Any convex optimization package can be used to solve the optimization problem. However, regular packages such as CVXOPT [12] could not handle the scale of our Twitter dataset and ran out of memory. Thus, we use the limited-memory BFGS algorithm with box constraints (L-BFGS-B) [13] to solve Eq. (9) and Eq. (10) by implicitly approximating the inverse Hessian matrix. We use the box constraints to enforce the non-negativity of the transmission rates.

4 Experimental Results

We evaluate the performance of our models by analyzing the diffusion of information in the popular Twitter network. Using a dataset crawled from January to October 2010 that contains 9,409,063 tweets published by 66,679 Twitter users, we analyze the cascading behavior of some trending topics and try to infer the underlying network structure.

4.1 Experimental Setup

Dataset Description. We conduct experiments on a subset of the Twitter network, which contains 66,679 nodes and 240,637 directed links. Each node represents a Twitter user and each edge represents a following relation. Contrary to previous work [1, 2], the adjacency matrix (ground truth) of this subgraph has also been crawled and thus is entirely known. In order to identify trending topics, we group the messages posted by these users according to their Hashtags[1]. We assume that messages containing the same Hashtag form a (generalized) cascade of a particular topic. Note that certain cascades corresponding to popular Hashtags might not be explained by our generative models. For example, #iphone is a widespread Hashtag that users often proactively include in

[1] Hashtags are words or phrases prefixed with the symbol # to label groups and topics.

their tweets rather than passively copy from another user. Therefore, we select a subset of not-so-popular Hashtags (e.g., #Mokpo and #GagaSouthAmerica2011), which are more specific and "local". That is, we consider those "local" cascades such that if node k writes about a Hashtag at time t_k, then it must have followed (or, have copied from) some node that wrote about the same Hashtag before time t_k.

This assumption is particularly important to our experiments. Since we only have a subset of the whole Twitter network, infected nodes observed in a cascade might have copied the information from some node that does not belong to this subset of users. Since we observe that MONET performs better on the cascades where the "locality" intuition holds, we trace the propagation of 500 Hashtags (that consist of 103,148 tweets) across the Twitter network from January to October 2010[2]. The average length of the cascades is 166, and there are a total of 2,521 unique users. On average, over 75% of the users post multiple times of the same Hashtag in each cascade. The inference is focused on the top 200 users that belong to the largest number of cascades. The size of the dataset is such that this inference problem can be solved by NETRATE and NETINF.

Feature Model. When collecting the Hashtags, we also record the entire message (or, tweet) containing the Hashtag. This represents the additional feature $\mathbf{f}_j$ for each node $j \in V$ in the generalized cascade model. In this paper, we use two primary distance metrics associated with texts: language and Jaccard index.

◇ **Language.** We observe that messages belonging to the same cascade (i.e., with the same Hashtag) are often written in several languages. For example, a cascade starting with an English tweet can spread to multilingual users who post tweets in Italian or Chinese but keep the original Hashtag. Intuitively, tweets in different languages, even if published closely in time, should not be considered as an implication of connectivity. Let $\ell(\cdot)$ be a function mapping a tweet to its language. We define a distance function with respect to language

$$d_L(\mathbf{f}_i, \mathbf{f}_j) = \begin{cases} 0, & \ell(\mathbf{f}_i) = \ell(\mathbf{f}_j); \\ 1, & \ell(\mathbf{f}_i) \neq \ell(\mathbf{f}_j). \end{cases}$$

The language information is computed using the n-gram model proposed in [14]. Note that this language identification algorithm provides noisy estimates.

◇ **Pairwise similarity.** We include pairwise similarity (a.k.a. Jaccard index) as another distance metric in our models. Given two tweets $\mathbf{f}_j$ and $\mathbf{f}_k$ posted by node j and node k, the distance function with respect to Jaccard index is defined as

$$d_J(\mathbf{f}_i, \mathbf{f}_j) = 1 - J_{jk} = 1 - \frac{|\mathbf{f}_j \cap \mathbf{f}_k|}{|\mathbf{f}_j \cup \mathbf{f}_k|},$$

where we consider the tweets as sets of words. Intuitively, besides the time factor, node k is more likely to have copied the information from node j if their tweets have higher similarity.

◇ **Combination.** We also consider both language and Jaccard similarity as a combined feature, defining another distance function

$$d_{L+J}(\mathbf{f}_i, \mathbf{f}_j) = w_J d_J(\mathbf{f}_i, \mathbf{f}_j) + w_L d_L(\mathbf{f}_i, \mathbf{f}_j).$$

[2] We will release an anonymized dataset due to Twitter's data privacy policy.

where w_J and w_L are the weights associated with the language and the Jaccard similarity feature. Since Jaccard similarity is typically small, we use a weight w_J to ensure that its contribution is comparable to that of the language distance.

Note that the normalization constant γ in Eq. (5) is hard to compute, since it involves a summation over all possible messages of up to 140 characters (the maximum length allowed by Twitter). In our experiments, we consider γ as fixed and independent of $\mathbf{f}_j$. In the case of language, this is equivalent to assuming that there are roughly the same number of possible messages for any given language.

Our optimization framework contains a hierarchical set of models for the MAP inference problem: splitting/non-splitting with multiple occurrences (MONET), language (MONET+L), Jaccard index (MONET+J), and their combination (MONET+LJ).

Evaluation Measures. To evaluate the performance of our feature-enhanced models, we consider the following aspects:

- ◇ **Baseline.** We use NETRATE [1] and NETINF [2] as two baselines to compare with our models. Since repeated occurrences are not allowed in NETRATE, we keep exactly one copy of each node and remove all other duplicates from each cascade. We use the true number of edges as an input parameter for NETINF. Due to license issues with the optimization software, we do not compare with CONNIE [3] in this paper, but its performance is comparable with that of NETRATE and NETINF according to previous literature.
- ◇ **Quantitative performance.** We use precision, recall, and F1-score to evaluate the performance of our models against the baselines. These measures focus on the number of correct pairs of nodes inferred. For example, given a pair of nodes (k, j) such that k is following j, if our method suggests that $\alpha_{jk} > 0$ (i.e., information flows from j to k), then consider it as a true positive (TP). False positives (FP) and false negatives (FN) are defined in a similar way.
- ◇ **Efficiency.** We evaluate the efficiency (i.e., elapsed time required for obtaining the optimum) of our feature-enhanced models.

All algorithms are implemented using Python with the Fortran implementation of L-BFGS-B available in Scipy [15], and all experiments are performed on a machine running CentOS Linux with a 6-core Intel x5690 3.46GHZ CPU and 48GB memory.

4.2 Quantitative Performance

We trace the propagation of a set of 500 Hashtags that consist of 103,148 tweets across a subset of the Twitter network that contains 66,679 nodes and 240,637 directed links. We want to infer the connectivity of the top 200 users that appear in the largest number of these 500 cascades. We evaluate our models against NETRATE and NETINF by comparing the inferred network and the ground truth via three metrics: precision, recall, and F1-score. F1-score, the harmonic mean of precision and recall, measures the accuracy of the estimates. The primary model MONET handles the basic scenario where nodes can have multiple occurrences in one cascade. As discussed in Section 2.1, MONET can be extended to consider a set of additional features, such as language (MONET+L), Jaccard similarity (MONET+J), and both (MONET+LJ).

Table 2. Performance comparison on Twitter (non-splitting/exponential).

METRIC	METHOD					
	NETINF	NETRATE	MONET	MONET+L	MONET+J	MONET+LJ
PRECISION	0.362	0.592	0.434	0.464	0.524	0.533
RECALL	0.362	0.069	0.307	0.374	0.450	0.483
F1-SCORE	**0.362**	**0.124**	**0.359**	**0.414**	**0.484**	**0.507**
TP	518	99	439	535	644	692
FP	914	62	573	618	586	606
FN	914	1333	993	897	788	740

Table 3. Performance comparison on Twitter (splitting/exponential).

METRIC	METHOD					
	NETINF	NETRATE	MONET	MONET+L	MONET+J	MONET+LJ
PRECISION	0.362	0.592	0.514	0.516	0.531	0.534
RECALL	0.362	0.069	0.599	0.605	0.618	0.635
F1-SCORE	**0.362**	**0.124**	**0.554**	**0.557**	**0.571**	**0.581**
TP	518	99	858	867	885	910
FP	914	62	810	812	781	793
FN	914	1333	574	565	547	522

Upper Bound of Recall. Similar to previous models, MONET requires node j to appear at least once before node k for $\alpha_{jk} > 0$ to be possibly inferred (i.e., information flows from j to k). For our dataset, no more than 86.4% of the edges in the ground truth can be recovered given the cascades. This represents an upper bound on recall for any probabilistic model based on the causality of the diffusion process.

Exponential Model. Table 2 and Table 3 compare the precision, recall, and F1-score of our non-splitting and splitting models introduced in Section 2.2 with NETRATE and NETINF according to the exponential distribution (see Table 1). NETRATE tends to be highly conservative when estimating the connectivity of the Twitter network, and thus has good precision but very low recall. NETINF knows how many edges there are in the true network, and slightly improves over NETRATE. Without knowing the ground truth, MONET balances the precision-recall trade-off and improves the accuracy over NETRATE by 65.5% for the non-splitting case and 77.6% for the splitting case. As expected, MONET+L, MONET+J, and MONET+LJ further improve the F1-score on top of MONET with the help of additional features. In particular, MONET+LJ improves the accuracy by as much as 78.7% over NETRATE and 37.7% over NETINF for the splitting case.

Rayleigh Model. Table 4 and Table 5 compare the precision, recall, and F1-score of our non-splitting and splitting models with NETRATE and NETINF according to the Rayleigh distribution (see Table 1). Similarly, without knowing the ground truth, MONET balances the precision-recall trade-off and improves the accuracy over NETRATE by 55.7% for the non-splitting case and 75.5% for the splitting case. MONET+L,

Table 4. Performance comparison on Twitter (non-splitting/Rayleigh).

METRIC	METHOD					
	NETINF	NETRATE	MONET	MONET+L	MONET+J	MONET+LJ
PRECISION	0.354	0.560	0.420	0.454	0.479	0.484
RECALL	0.354	0.072	0.218	0.262	0.286	0.294
F1-SCORE	**0.354**	**0.127**	**0.287**	**0.332**	**0.358**	**0.366**
TP	507	103	312	375	409	421
FP	925	81	430	451	445	449
FN	925	1329	1120	1057	1023	1011

Table 5. Performance comparison on Twitter (splitting/Rayleigh).

METRIC	METHOD					
	NETINF	NETRATE	MONET	MONET+L	MONET+J	MONET+LJ
PRECISION	0.354	0.560	0.480	0.493	0.495	0.499
RECALL	0.354	0.072	0.562	0.566	0.570	0.572
F1-SCORE	**0.354**	**0.127**	**0.518**	**0.527**	**0.530**	**0.533**
TP	507	103	805	811	816	819
FP	925	81	872	835	834	821
FN	925	1329	627	621	616	613

MONET+J, and MONET+LJ further improve the F1-score on top of MONET with the help of additional features. In particular, MONET+LJ improves the accuracy by as much as 76.2% over NETRATE and 33.4% over NETINF for the splitting case.

Remarks. We have similar observations for the performance comparison according to the power-law distribution, but the tables are omitted here due to the space limitation. Our results suggest that the splitting model performs better than the non-splitting one, with much more true positives and far fewer false negatives. This suggests that the information diffusion in the Twitter network is better approximated by a memoryless process. Specifically, how a message posted by a Twitter user will be retweeted is not relevant to that user's previous history. Further, the exponential model provides slightly more accurate estimates over the Rayleigh one. The performance improvement achieved using the language information is smaller compared to that achieved using Jaccard similarity, but MONET+L improves over MONET and MONET+LJ improves over MONET+J. This suggests that the language feature does provide some useful information, although its effectiveness is likely to be limited by the noisy estimates provided by the language detection algorithm we use in our experiments.

4.3 Efficiency

Solving each of the sub-problems defined in the basic model MONET (i.e., optimizing one column of the transmission rate matrix $\mathbf{A}$) takes about 2 minutes on average using L-BFGS-B with the history parameter $m = 10$. The running time, however, depends on the specific column being optimized, and ranges from a few seconds to several minutes. Introducing additional features requires an additional preprocessing time (in the order of minutes) to precompute the languages of the messages and the Jaccard indexes between the messages, but it does not significantly affect the running time of the optimization procedure.

5 Related Work

A substantial amount of work has been devoted to the task of studying cascading processes in large-scale networks. Largely motivated by marketing applications, the predominant focus over the past decade has been on optimization problems, where the goal is to maximize the spread of a certain cascade through a given network, either by selecting a good subset of nodes to initiate the cascade [5] or by applying a broader set of intervention strategies such as node and edge additions [7, 10]. As networks and networked systems are playing an increasingly important role in a number of disciplines, ranging from the interconnections between financial systems to epidemiology and ecology, researchers have recently begun to consider the problem of inferring the unknown (latent) underlying network given some observed cascading behavior [1–3]. Specifically, several generative probabilistic models have been developed to explain cascading behaviors, where the task of inferring the underlying network is tractable, involving the optimization of submodular [2] or convex objective functions [1, 3]. These models have been shown to perform well on a number of synthetic datasets, but there has been very limited experimentation on real-world scenarios. Moreover, the MemeTracker dataset [2] commonly used in previous work has no ground truth.

There are several obstacles when trying to apply these models to real-world problems, such as inferring the latent structure of a social network based on the diffusion of trending topics. Specifically, cascades are often formed by a mixed set of sub-cascades and it is difficult to obtain i.i.d. samples. However, real-world cascades also present a range of new opportunities to define richer probabilistic models. Previous work combined latent features with explicit ones to solve structural link prediction problems [16]. In this paper, we propose a feature-enhanced framework to address the scenario where nodes can be repeatedly infected. We develop a family of novel probabilistic models based not only on the time intervals between infection events, but also on a set of additional features, such as the content and the language of the messages exchanged in social media.

6 Conclusions

In this paper, we propose a family of feature-enhanced probabilistic models to infer the latent network structure from observations of a diffusion process. We develop a primary model called MONET with non-splitting and splitting solutions that can explain recurrent processes where nodes can be repeatedly infected (i.e., multiple occurrences in one cascade). Further, our models take into account not only the time differences between infection events, but also a richer set of features. The MAP inference problem, which still involves the optimization of a convex objective function, can be decomposed into smaller sub-problems that we can efficiently solve in parallel. Our experiments on the Twitter network show that our models successfully recover the underlying network structure, and significantly improve the performance over previous models based solely on time.

7 Acknowledgement

This research was partially supported by the U.S. AFOSR Grant FA9550-09-1-0675 and NSF Grant 0832782.

References

1. M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, pages 561–568, 2011.
2. M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD*, pages 1019–1028, 2010.
3. S.A. Myers and J. Leskovec. On the convexity of latent social network inference. In *NIPS*, 2010.
4. E. Adar and L.A. Adamic. Tracking information epidemics in blogspace. In *Web Intelligence*, pages 207–214, 2005.
5. D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.
6. T. Lappas, E. Terzi, D. Gunopulos, and H. Mannila. Finding effectors in social networks. In *KDD*, 2010.
7. M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.
8. D.J. Watts and P.S. Dodds. Influentials, networks, and public opinion formation. *Journal of Consumer Research*, 34(4), 2007.
9. J. Wallinga and P. Teunis. Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, 160(6), 2004.
10. D. Sheldon, B. Dilkina, A. Elmachtoub, R. Finseth, A. Sabharwal, J. Conrad, C. Gomes, D. Shmoys, W. Allen, O. Amundsen, et al. Maximizing the spread of cascades using network design. In *UAI*, 2010.
11. J.F. Lawless. Statistical models and methods for lifetime data. 1982.
12. J. Dahl and L. Vandenberghe. CVXOPT: A Python package for convex optimization. In *Proc. Eur. Conf. Op. Res*, 2006.
13. C. Zhu, R.H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, 1997.
14. W.B. Cavnar and J.M. Trenkle. N-gram-based text categorization. In *Proc. 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994.
15. E. Jones, T. Oliphant, and P. Peterson. Scipy: Open source scientific tools for Python. 2001. `http://www.scipy.org/`.
16. A. Menon and C. Elkan. Link prediction via matrix factorization. *Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.