Network Science **2** (1): 26–65, 2014.
doi:10.1017/nws.2014.3

Uncovering the structure and temporal dynamics of information propagation

MANUEL GOMEZ RODRIGUEZ
Department of Empirical Inference, MPI for Intelligent Systems, Tübingen, Baden-Wurttemberg, Germany
(e-mail: manuelgr@tuebingen.mpg.de*)*

JURE LESKOVEC
Department of Computer Science, Stanford University, Stanford, CA, USA
(e-mail: jure@cs.stanford.edu*)*

DAVID BALDUZZI
Machine Learning Laboratory, ETH Zürich, Zürich, Switzerland
(e-mail: david.balduzzi@inf.ethz.ch*)*

BERNHARD SCHÖLKOPF
Department of Empirical Inference, MPI for Intelligent Systems, Tübingen, Baden-Wurttemberg, Germany
(e-mail: bs@tuebingen.mpg.de*)*

Abstract

Time plays an essential role in the diffusion of information, influence, and disease over networks. In many cases we can only observe when a node is activated by a contagion—when a node learns about a piece of information, makes a decision, adopts a new behavior, or becomes infected with a disease. However, the underlying network connectivity and transmission rates between nodes are unknown. Inferring the underlying diffusion dynamics is important because it leads to new insights and enables forecasting, as well as influencing or containing information propagation. In this paper we model diffusion as a continuous temporal process occurring at different rates over a latent, unobserved network that may change over time. Given information diffusion data, we infer the edges and dynamics of the underlying network. Our model naturally imposes sparse solutions and requires no parameter tuning. We develop an efficient inference algorithm that uses stochastic convex optimization to compute online estimates of the edges and transmission rates. We evaluate our method by tracking information diffusion among 3.3 million mainstream media sites and blogs, and experiment with more than 179 million different instances of information spreading over the network in a one-year period. We apply our network inference algorithm to the top 5,000 media sites and blogs and report several interesting observations. First, information pathways for general recurrent topics are more stable across time than for on-going news events. Second, clusters of news media sites and blogs often emerge and vanish in a matter of days for on-going news events. Finally, major events, for example, large scale civil unrest as in the Libyan civil war or Syrian uprising, increase the number of information pathways among blogs, and also increase the network centrality of blogs and social media sites.

Keywords: *diffusion networks*, *information cascades*, *information propagation*, *meme tracking*, *information networks*, *social networks*, *news media*, *blogs*

1 Introduction

Many interacting systems can be effectively modeled in terms of signals propagating over underlying networks. In recent years, there has been an increasing effort to uncover, model, and understand a broad range of propagation processes arising over a wide variety of network structures: propagation of information (Adar & Adamic, 2005; Leskovec et al., 2007a; Gomez-Rodriguez et al., 2010; Romero et al., 2011), adoption of new products (Leskovec et al., 2006; Watts & Dodds, 2007; Aral & Walker, 2012), diffusion of technical innovations (Rogers, 1995), spread of chain letters (Liben-Nowell & Kleinberg, 2008), promotion of products via viral marketing (Kempe et al., 2003; Leskovec et al., 2007b; Lappas et al., 2010; Du et al., 2013), spread of computer viruses (Wang et al., 2000) and infectious diseases (Lipsitch et al., 2003; Hufnagel et al., 2004; Wallinga & Teunis, 2004), and even the diffusion of human travel (Brockmann et al., 2006).

Observing a diffusion process often reduces to recording when nodes (people, blogs, etc.) get infected by a virus, mention a piece of information, buy a product, adopt a new behavior, or, more generally, adopt a *contagion*. However, the mechanism underlying the process is often hidden. For example, in epidemiology we can often observe when a person becomes ill, but we cannot tell who infected her or how many exposures were necessary for the infection to take hold. In information propagation, we observe when a blog mentions a piece of information. However, if, as is often the case, the blogger does not link to her source, we do not know from where she acquired the information, or how long it took her to post it. Finally, viral marketers can track when customers buy products or subscribe to services, but typically cannot observe who influenced customers' decisions, how long they took to make up their minds, or when they passed recommendations on to other customers. In all these scenarios, we observe *where and when* but *not how or why* information (be it in the form of a virus, a meme, or a decision) propagates through a population. We often observe the result of the diffusion process but not the process itself. However, understanding and inferring the dynamics of the underlying diffusion process is important because it enables stopping diseases, predicting information propagation, or maximizing sales of products.

A way to capture the dynamics of the underlying process is to infer the links of the underlying network, which provides a skeleton or a medium for the process to spread. So we can assume that a dynamic process propagates over the links of a hidden or unobserved network. Given the times when nodes adopt a set of contagions, the goal would then be to infer the structure of the underlying network. Importantly, networks are often dynamic and change depending upon the activations that previously propagated through them (Romero et al., 2011). For example, a blog can abruptly increase its popularity after one of its posts turns *viral*. This may create new edges in the information transmission network, and so the content the blog produces in future will likely spread to larger parts of the network. Similarly, at any given time an unexpected event may occur and a topic or piece of news may become popular for a limited period of time. This will again cause pathways to emerge and vanish, and thus contribute to a time-varying underlying network. Therefore, to understand these temporal changes, one needs algorithms that can reconstruct the time-varying structure and underlying temporal dynamics of networks so

that we can analyze the information pathways of real-world events, topics, or content.

1.1 Inferring dynamic networks

We consider a problem where dynamic processes unfold over an unobserved time-varying network, and the goal is to reconstruct the unobserved network and its temporal dynamics. In particular, we tackle the problem when the network is slowly changing over time. In this network, each dynamic process corresponds to a *contagion*, which spreads from node to node over the edges of the network. We say that a node becomes *active* when adopts (or gets infected) by the contagion. We only observe the times when nodes get activated and our goal is to infer the structure and the dynamics of the underlying unobserved network from these temporal traces.

Here we present a method for inferring the mechanisms underlying diffusion processes based on observed information diffusion data. To do so, we construct a model of diffusion that operates under the following setting:

a. A contagion spreads from node to node over the edges of the network.
b. As a node adopts a contagion it becomes *active*. Activations are binary, i.e., a node is either activated or it is not.
c. Once the node gets activated by a contagion, it (probabilistically) spreads the contagion along each of its outgoing edges.
d. Activations can have different speeds and delays: The likelihood of node a activating node b t time-units after its activation is modeled via a probability density function depending on a, b, and t.
e. Contagions propagate in isolation of each other and do not interact with each other.

Now, given that we observe the times of *all* activations, for many contagions, during a recorded time window, our aim is to infer the links of the underlying network over which the contagions spread. For every edge of the network we also aim to estimate how its transmission rate or strength varies over time, which in turn means we infer the dynamics of the underlying network that acts as a medium for the propagation of the contagions.

In more detail, we first formulate a generative probabilistic model of diffusion that aims to realistically describe how activations occur over time in a static network. The model considers information that propagates over the edges of the network. We then generalize the model to support dynamic networks whose structure changes over time. Solving both the static and dynamic networks inference problem reduces to solving a convex optimization problem. The convex problem decouples into many smaller problems, which can be efficiently solved using a stochastic gradient-based method (Robbins & Monro, 1951). The decoupling further allows for a fast parallel implementation, which scales to large datasets and allows for inferring networks of hundreds of thousands of nodes.

We first test our algorithm using synthetic data and show that the method is robust across network topologies, transmission models, and variations in the transmission rates over time. We then apply our algorithm to synthetic data and to a real Web information propagation dataset of 179 million different information contagions

Table 1. *Notation.*

Symbol	Description
$G(V, E)$	Directed diffusion network with node set V and edge set E
C	Set of all recorded cascades
C_t	Set of recorded cascades by time t
T^c	Observation window, time horizon, or time interval for cascade c
$\mathbf{t}^c$	Activation times for cascade c during a time interval of length T^c
$\mathbf{t}^{\leqslant T^c}$	Observed activation times for cascade c during a time interval of length T^c
t_i^c	Activation time of node i in cascade c
$\alpha_{i,j}$	Pairwise transmission rate between node i and node j (refer to Table 2)
$\mathbf{A}$	Pairwise transmission rates for all pair of nodes (i, j)
$f(t_i \mid t_j, \alpha_{j,i})$	Pairwise transmission likelihood of edge $j \to i$
$F(t_i \mid t_j, \alpha_{j,i})$	Cumulative density function of edge $j \to i$
$S(t_i \mid t_j; \alpha_{j,i})$	*Survival function* of edge $j \to i$
$H(t_i \mid t_j; \alpha_{j,i})$	*Hazard function*, or instantaneous activation rate, of edge $j \to i$
$g(\mathbf{A})$	Prior likelihood on the transmission rates $\mathbf{A}$
$\mathscr{A}$	Support of the prior likelihood on the transmission rates $g(\mathbf{A})$

spreading among 3.3 million blogs and news media sites over a one-year period from March 2011 to February 2012.[1]

Experiments on large-scale real news and social media data lead to interesting qualitative insights and findings. For example, we find that the information pathways over which general recurrent topics propagate remain stable across time, while unexpected events lead to dramatically changing information pathways. Clusters of mainstream news and blogs often emerge and vanish in a matter of days, and our online algorithm is able to uncover such structures. News events that involve civil unrest, as the Libyan civil war, Egypt's revolution, or the Syrian uprising, result in a greater increase in information transfer among blogs than among mainstream media. Perhaps surprisingly, the amount of mainstream media and blogs among the most influential nodes for most topics or news events are comparable. However, we find that growing numbers of influential blogs on some topics or news events are often temporally correlated with increasing social unrest (e.g., the Occupy Wall Street movement in September–November 2011).

1.2 Related works

The problem of inferring links of diffusion was first studied by Adar & Adamic (2005), who formulated it as a supervised classification problem and used Support Vector Machines combined with rich textual features to predict the occurrence of individual links. Although rich textual features are used, links are predicted independently and no information about the temporal dynamics of the network is provided.

Several network inference algorithms have been developed recently (Gomez-Rodriguez et al., 2010; Gomez-Rodriguez et al., 2012; Myers & Leskovec, 2010; Snowsill et al., 2011; Netrapalli & Sanghavi, 2012; Gomez-Rodriguez & Schölkopf,

[1] The data and the implementation of our algorithm are publicly available at the supporting website: http://snap.stanford.edu/infopath.

2012c; Du et al., 2012). Some approaches infer only the network structure (Gomez-Rodriguez et al., 2010; Gomez-Rodriguez et al., 2012; Snowsill et al., 2011; Gomez-Rodriguez & Schölkopf, 2012c), while others infer not only the network structure but also the prior probability of activation of edges in the network (Myers & Leskovec, 2010) or the transmission rates (Du et al., 2012). To the best of our knowledge, previous works have always assumed the transmission rates between all nodes to be fixed and networks to be static so that information propagates over pathways that remain constant over time.

The work most closely related to ours (Gomez-Rodriguez et al., 2010; Myers & Leskovec, 2010) also uses a generative probabilistic model for inferring diffusion networks. Gomez-Rodriguez et al. (2010) (NetInf) infers network connectivity using submodular optimization, and Myers & Leskovec (2010) (ConNIe) infer not only the connectivity but also a prior probability of activation for every edge using a convex program and some heuristics. However, both papers force the transmission rate between all nodes to be fixed—and not inferred—and the networks to be static, i.e., the network structure and transmission rates do not change over time: they consider the pathways over which information propagates to be time-invariant. In contrast, our model allows transmission at different rates across different edges, and dynamic networks that change over time. Thus, we can now infer the temporal dynamics of the underlying (possibly dynamic) network.

The main technical innovation of this paper is to model diffusion as a discrete network of continuous, conditionally independent, temporal processes occurring at different rates. Transmission of activations depends on the complex intricacies of the underlying mechanisms (e.g., a person's susceptibility to viral infections depends on weather, diet, age, stress levels, prior exposures to similar pathogens, and so on). However, we avoid modeling the mechanisms underlying individual activations, and instead develop a data-driven approach, suitable for large-scale analyses, that infers the diffusion process using only the visible spatiotemporal traces (cascades) it generates. We therefore model diffusion using only time-dependent pairwise transmission likelihood between pairs of nodes, transmission rates, and activation times, but not prior probabilities of activation that depend on unknown external factors. We believe that developing a data-driven approach is a key point for understanding diffusion processes. Moreover, continuous temporal dynamics of diffusion networks has not been modeled or inferred in previous works.

The remainder of the paper is organized as follows. Section 2 presents our continuous time model of diffusion and the network inference problem. Section 3 shows how to optimally perform inference using stochastic gradient descent. Section 4 evaluates our method qualitatively and quantitatively on synthetic data. Section 5 evaluates the performance of our method on real diffusion data and begins to extract qualitative insights into information propagation in online media. We conclude with a discussion of our results in Section 6.

2 Problem formulation

In this section we formulate our continuous time model of diffusion, starting from the data it is designed for, and concluding with a precise statement of the network inference problem for both static and dynamic networks.

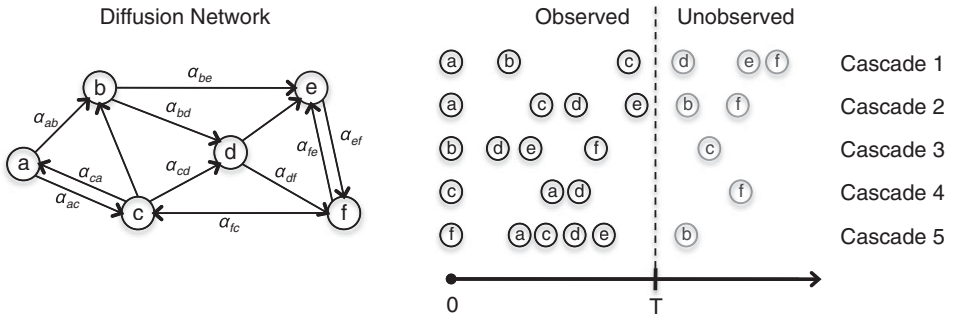


Fig. 1. We observe a set of cascades (right) within an unknown diffusion network (left). For each cascade c, we only observe the times in which nodes get infected up to time T, but not who infected whom. Our goal is to infer the network and transmission rates $\alpha_{i,j}$ based on the observed cascades.

2.1 Data

We observe multiple waves of contagions that propagate on a fixed population of N nodes. As the contagion spreads from activated to non-activated nodes, it creates a *cascade*. For each contagion c, we observe a cascade $\mathbf{t}^c$, which is simply a record of observed node activation times during an observation time window of length T^c. In an information propagation setting, each cascade corresponds to a different piece of information and the activation time of a node is simply the time when the node first heard of or mentioned the piece of information.

We record a set C of cascades $\{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}$. A cascade $\mathbf{t}^c = (t_1^c, \ldots, t_N^c)$ is an N-dimensional vector recording whether and, if so, when each of N nodes got activated by the contagion c during a time interval of length T^c. Thus, $t_k^c \in [t_0, t_0 + T^c] \cup \{\infty\}$, where symbol ∞ labels nodes that are not activated by the contagion c during observation window $[t_0, t_0 + T^c]$—it does not imply that nodes are never activated—and t_0 is the activation time of the first node. Lengthening the observation window T^c increases the number of observed activations within a cascade c and results in a more representative sample of the underlying dynamics. However, these advantages must be weighed against the cost of observing for longer periods. For simplicity, we assume $T^c = T$ for all cascades; the results generalize trivially. Contagions often propagate simultaneously (Myers & Leskovec, 2012; Prakash et al., 2012) over the same network, but we assume each contagion to propagate independently of each other. Finally, we also assume that all activated nodes except the first one are activated by network diffusion, i.e., by previously activated nodes, ignoring external influences (Myers et al., 2012). We illustrate this process in Figure 1.

Given a set of node activation times of many different contagions, our goal is to infer the underlying (possibly dynamic) network over which contagions propagated. Importantly, the time-stamps assigned to nodes in each cascade induce a directed acyclic graph (DAG) involving those nodes, which need not to be acyclic in the containing network topology. Thus, it is meaningful to refer to parents and children within a cascade, but not on the network. The DAG structure dramatically simplifies the computational complexity of the inference problem.

Table 2. *Pairwise transmission models.*

Model	Transmission likelihood $f(t_i\|t_j;\alpha_{j,i})$		Log survival $\log S(t_i\|t_j;\alpha_{j,i})$	Hazard $H(t_i\|t_j;\alpha_{j,i})$
Exp	$\begin{cases}\alpha_{j,i}\cdot e^{-\alpha_{j,i}(t_i-t_j)}\\ 0\end{cases}$	if $t_j < t_i$ otherwise	$-\alpha_{j,i}(t_i-t_j)$	$\alpha_{j,i}$
Pow	$\begin{cases}\frac{\alpha_{j,i}}{\delta}\left(\frac{t_i-t_j}{\delta}\right)^{-1-\alpha_{j,i}}\\ 0\end{cases}$	if $t_j+\delta < t_i$ otherwise	$-\alpha_{j,i}\log\left(\frac{t_i-t_j}{\delta}\right)$	$\alpha_{j,i}\cdot\frac{1}{t_i-t_j}$
Ray	$\begin{cases}\alpha_{j,i}(t_i-t_j)e^{-\frac{1}{2}\alpha_{j,i}(t_i-t_j)^2}\\ 0\end{cases}$	if $t_j < t_i$ otherwise	$-\alpha_{j,i}\frac{(t_i-t_j)^2}{2}$	$\alpha_{j,i}\cdot(t_i-t_j)$

2.2 Pairwise transmission likelihood

The first step in modeling diffusion dynamics is to consider pairwise interactions. For every pair of nodes (j,i), we define a pairwise transmission rate $\alpha_{j,i}$ which models how frequently information spreads from a node j to a node i; the *strength* of an edge (j,i). We pay attention to a quite general case of heterogeneous pairwise transmission rates, i.e., activations can occur at different transmission rates over different edges of a network. As $\alpha_{j,i}\to 0$, the likelihood of transmission tends to zero and the expected transmission time becomes arbitrarily long. Allowing edge transmission rates to dynamically increase and decay over time will enable us to infer time-varying (dynamic) diffusion networks.

Now we define $f(t_i|t_j;\alpha_{j,i})$ as the conditional likelihood of transmission between nodes j and i. The transmission likelihood depends on the activation times (t_j,t_i) and a pairwise transmission rate $\alpha_{j,i}$. A node cannot be activated by another node activated later in time. In other words, a node j that has been activated at a time t_j may activate a node i at a time t_i only if $t_j < t_i$, otherwise $f(t_i|t_j;\alpha_{j,i}) = 0$. The shape of the conditional likelihood of transmission may depend on the particular setting (information, influence, diseases, etc.) in which propagation takes place. In some scenarios, it may be possible to estimate a non-parametric likelihood, while in others, expert knowledge may be used to decide upon a parametric model. For simplicity, we consider three well-known parametric models: exponential (Exp), power-law (Pow), and Rayleigh (Ray) models (see Table 2). In the power-law model, to have a bounded likelihood, we set δ as the minimum allowed time difference. Without loss of generality, we consider $\delta = 1$ in the power-law model from now on.

Exponential and power-laws are monotonic models that have been previously used in modeling diffusion networks and social networks (Gomez-Rodriguez et al., 2010; Myers & Leskovec, 2010). Power-law model activates with long tails. The Rayleigh model is a non-monotonic parametric model previously used in epidemiology (Kaplan, 1989; Wallinga & Teunis, 2004). It is well adapted to modeling fads, where infection likelihood rises to a peak and then drops extremely rapidly. In all three models, as $\alpha_{j,i}\to 0$, the likelihood of infection tends to zero.

We recall some additional notation that is standard in survival analysis and epidemiology (Lawless, 1982). The cumulative density function, denoted as $F(t_i|t_j;\alpha_{j,i})$, is computed from the transmission likelihoods. Given that node j was activated at time t_j, the *survival function* of edge $j\to i$ is the probability that node j does *not*

cause node i to activate by time t_i:

$$S(t_i|t_j;\alpha_{j,i}) = 1 - F(t_i|t_j;\alpha_{j,i}).$$

The *hazard function*, or instantaneous activation rate, of edge $j \to i$ is the ratio

$$H(t_i|t_j;\alpha_{j,i}) = -\frac{S'(t_i|t_j;\alpha_{j,i})}{S(t_i|t_j;\alpha_{j,i})} = \frac{f(t_i|t_j;\alpha_{j,i})}{S(t_i|t_j;\alpha_{j,i})}.$$

The log-survival and hazard functions of our models are simple (see Table 2).

2.3 Probability of survival given a cascade

We compute the probability that a node survives as unactivated until time t_i, given that some of its parents are already activated. Consider a cascade $\mathbf{t} := (t_1, \ldots, t_N)$. Since each activated node k may activate i independently, the probability that nodes $1 \ldots N$ do *not* activate node i by time t_i is the product of the survival functions of the activated nodes $1 \ldots N | t_k \leqslant t_i$ targeting i,

$$S(t_i|t_1, \ldots, t_N \setminus t_i; \mathbf{A}) = \prod_{t_k \leqslant t_i} S(t_i|t_k;\alpha_{k,i}) \tag{1}$$

where $\mathbf{A} := \{\alpha_{j,i} \mid i, j = 1, \ldots, n, i \neq j\}$.

2.4 Likelihood of a cascade

Consider a cascade $\mathbf{t} := (t_1, \ldots, t_N)$. We first compute the likelihood of the observed activations $\mathbf{t}^{\leqslant T} = (t_1, \ldots, t_N | t_i \leqslant T)$. Since we assume that activations are conditionally independent given the parents of the activated nodes, the likelihood factorizes over nodes as

$$f(\mathbf{t}^{\leqslant T}; \mathbf{A}) = \prod_{t_i \leqslant T} f(t_i|t_1, \ldots, t_N \setminus t_i; \mathbf{A}). \tag{2}$$

Computing the likelihood of a cascade thus reduces to computing the conditional likelihood of activating each node given the rest of the cascade. As in the independent cascade model (Kempe et al., 2003), we assume that a node gets activated once the *first* parent activates the node. Given an activated node i, we compute the probability of a potential parent j to be the first parent by applying Equation (1),

$$f(t_i|t_j;\alpha_{j,i}) \times \prod_{j \neq k, t_k < t_i} S(t_i|t_k;\alpha_{k,i}). \tag{3}$$

We now compute the conditional likelihoods of Equation (2) by summing over the likelihoods of the mutually disjoint events that each potential parent is the first parent,

$$f(t_i|t_1, \ldots, t_N \setminus t_i; \mathbf{A}) = \sum_{j : t_j < t_i} f(t_i|t_j;\alpha_{j,i}) \times \prod_{j \neq k, t_k < t_i} S(t_i|t_k;\alpha_{k,i}). \tag{4}$$

By Equation (2) the likelihood of the activations in a cascade is

$$f(\mathbf{t}^{\leqslant T}; \mathbf{A}) = \prod_{t_i \leqslant T} \sum_{j : t_j < t_i} f(t_i|t_j;\alpha_{j,i}) \times \prod_{k : t_k < t_i, k \neq j} S(t_i|t_k;\alpha_{k,i}). \tag{5}$$

Removing the condition $k \neq j$ makes the product independent of j,

$$f(\mathbf{t}^{\leqslant T}; \mathbf{A}) = \prod_{t_i \leqslant T} \prod_{k:t_k < t_i} S(t_i|t_k; \alpha_{k,i}) \times \sum_{j:t_j < t_i} \frac{f(t_i|t_j; \alpha_{j,i})}{S(t_i|t_j; \alpha_{j,i})}, \tag{6}$$

and we can replace the ratios in Equation (6) with hazard functions:

$$f(\mathbf{t}^{\leqslant T}; \mathbf{A}) = \prod_{t_i \leqslant T} \prod_{k:t_k < t_i} S(t_i|t_k; \alpha_{k,i}) \times \sum_{j:t_j < t_i} H(t_i|t_j; \alpha_{j,i}). \tag{7}$$

Now we note that Equation (7) only considers activated nodes. However, the fact that some nodes are *not* activated during the observation window is also informative. We therefore add the multiplicative survival term from Equation (1):

$$f(\mathbf{t}; \mathbf{A}) = \prod_{t_i \leqslant T} \prod_{t_m > T} S(T|t_i; \alpha_{i,m}) \times \prod_{k:t_k < t_i} S(t_i|t_k; \alpha_{k,i}) \sum_{j:t_j < t_i} H(t_i|t_j; \alpha_{j,i}). \tag{8}$$

Assuming independent cascades, the likelihood of a set of cascades $C = \{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}$ is the product of the likelihoods of individual cascades given by Equation (8):

$$f(\{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}; \mathbf{A}) = \prod_{\mathbf{t}^c \in C} f(\mathbf{t}^c; \mathbf{A}). \tag{9}$$

The resulting continuous time model of diffusion is a particular case of Aalen's additive regression model, frequently used in survival theory analysis (Aalen et al., 2008) and recently used for link prediction in social network data (Vu et al., 2011). In Aalen's model, the hazard function, or instantaneous activation rate, of a node i is parametrized as $\alpha_{i,0}(t) + \alpha(t)_i^T \mathbf{s}_i(t)$, where $\alpha(t)$ is a vector that accounts for the effect of a collection of observable covariates $\mathbf{s}(t)$ and $\alpha_{i,0}(t)$ is a baseline. Using Equation (4) and the definition of hazard function, it is easy to show that the hazard function of node i at time t_i for the three pairwise transmission models, exponential, power-law, and Rayleigh, has the following form:

$$H(t_i|t_1, \ldots, t_N \setminus t_i; \mathbf{A}) = \alpha_i^T \mathbf{s}_i(t_i; t_1, \ldots, t_N \setminus t_i) = \sum_{j:j \neq i} \alpha_{j,i} s_i(t_i; t_j), \tag{10}$$

where the baseline is zero, $\alpha_i = (\alpha_{1,i}, \ldots, \alpha_{N,i})$ accounts for the effect of a collection of observable covariates $s_i(t_i; t_j)$, and the covariates depend on the pairwise transmission model (exponential, power-law, or Rayleigh) and the previously activated nodes as follows:

$$\begin{aligned} s_i(t_i; t_j) &= I(t_j < t_i) && \text{exponential likelihood;} \\ s_i(t_i; t_j) &= \max(0, 1/(t_i - t_j)) && \text{power-law likelihood;} \\ s_i(t_i; t_j) &= \max(0, t_i - t_j) && \text{Rayleigh likelihood.} \end{aligned}$$

However, Aalen's additive regression model entails some drawbacks in comparison with our model. It is computationally more expensive since it is necessary to solve one least square problem per activation time per node. In addition, some of these least square problems are often underdetermined, and the model can stray into negative values for hazard rates.

In contrast with our approach, an alternative multiplicative model of diffusion has been recently proposed to model information propagation (Gomez-Rodriguez & Schölkopf, 2012b). The model considers the hazard function to be multiplicative

on the previously infected nodes and it is a particular case of Cox's multiplicative regression model (Aalen et al., 2008).

2.5 Three network inference problems

Given a static network with constant transmission rates $\alpha_{j,i}$, the network inference problem reduces to solving a maximum likelihood problem.

Problem 1 (*Static network inference*)
Given an observed set of cascades $C = \{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}$, our goal is to find the underlying transmission rates $\alpha_{j,i}$ by solving the following maximum likelihood (ML) optimization problem:

$$\begin{array}{ll} \text{minimize}_{\mathbf{A}} & -\sum_{c \in C} \log f(\mathbf{t}^c; \mathbf{A}) \\ \text{subject to} & \alpha_{j,i} \geqslant 0,\ i, j = 1, \ldots, N, i \neq j, \end{array} \tag{11}$$

where $\mathbf{A} := \{\alpha_{j,i} \,|\, i, j = 1, \ldots, n, i \neq j\}$ are the variables. The edges of the network are the pairs of nodes with transmission rates $\alpha_{j,i} > 0$.

Now we generalize the network inference problem to dynamic networks with transmission rates $\alpha_{j,i}(t)$ that may change over time.

Problem 2 (*Dynamic network inference*)
Given a time t and a set of recorded cascades by time t, $C_t = \{\mathbf{t}^1, \ldots, \mathbf{t}^{|C_t|}\}$, our goal is to find the optimal transmission rates $\alpha_{j,i}(t)$ by solving the following maximum likelihood optimization problem:

$$\begin{array}{ll} \text{minimize}_{\mathbf{A}(t)} & -\sum_{c \in C_t} w_c(t) \log f(\mathbf{t}^c; \mathbf{A}(t)) \\ \text{subject to} & \alpha_{j,i}(t) \geqslant 0,\ i, j = 1, \ldots, N, i \neq j \end{array} \tag{12}$$

where $w_c(t) \geqslant 0$ are weights that penalize old cascades (the older a cascade c, the smaller its weight $w_c(t)$) and $\mathbf{A}(t) := \{\alpha_{j,i}(t) \,|\, i, j = 1, \ldots, n, i \neq j\}$ are the variables. The intuition here is that the diffusion network smoothly changes over time and that recent cascades have higher importance in determining current network structure than old cascades. Thus, at any point in time we can solve the above optimization problem to obtain the structure of the diffusion network at that particular time.

The dynamic network inference problem defined by Equation (12) reduces to the static network inference problem defined by Equation (11) when we set all weights $w_c(t)$ to be equal and constant over time.

Finally, in some scenarios we may have access to additional information that lets us estimate a prior likelihood on the transmission rates $\alpha_{j,i}(t)$. For example, in an example from information networks, a blog may sometimes link to its sources, and therefore we can compute a prior on the transmission rates from the sources to the blog using those links. In such cases, we can solve instead a maximum a posteriori (MAP) optimization problem.

Problem 3 (*Network inference with prior likelihood*)
Given a time t and a set of recorded cascades by time t, $C_t = \{\mathbf{t}^1, \ldots, \mathbf{t}^{|C_t|}\}$, and a prior likelihood $g(\mathbf{A}(t))$ on the transmission rates $\alpha_{j,i}(t)$, our goal is to find the optimal transmission rates $\alpha_{j,i}(t)$ by solving the following maximum a posteriori optimization problem:

$$\begin{array}{ll} \text{minimize}_{\mathbf{A}(t)} & -\sum_{c \in C} w_c(t) \log f(\mathbf{t}^c; \mathbf{A}(t)) - \log g(\mathbf{A}(t)) \\ \text{subject to} & \mathbf{A}(t) \in \mathscr{A} \\ & \alpha_{j,i}(t) \geqslant 0,\ i, j = 1, \ldots, N, i \neq j \end{array} \tag{13}$$

where $w_c(t) \geqslant 0$ are weights that penalize old cascades (the older a cascade c, the smaller its weight $w_c(t)$), $\mathbf{A}(t) := \{\alpha_{j,i}(t) \,|\, i, j = 1, \ldots, n, i \neq j\}$ are the variables, and $\mathscr{A}$ is the support of the prior likelihood $g(\cdot)$.

3 Proposed algorithm: NetRate

The solutions to the static and dynamic networks inference problems defined by Equations (11) and (12) are unique, computable, and consistent.

Theorem 1
Given log-concave survival functions and concave hazard functions in the parameter(s) of the pairwise transmission likelihoods, the static and dynamic networks inference problems defined by Equations (11) and (12) are convex in $\mathbf{A}$.

Proof
By Equation (9), the log-likelihood of a cascade is

$$L(\mathbf{t}^c; \mathbf{A}) = \Psi_1(\mathbf{t}^c; \mathbf{A}) + \Psi_2(\mathbf{t}^c; \mathbf{A}) + \Psi_3(\mathbf{t}^c; \mathbf{A}) \tag{14}$$

where

$$\Psi_1(\mathbf{t}^c; \mathbf{A}) = \sum_{i:t_i \leqslant T} \sum_{t_m > T} \log S(T|t_i; \alpha_{i,m})$$

$$\Psi_2(\mathbf{t}^c; \mathbf{A}) = \sum_{i:t_i \leqslant T} \sum_{j:t_j < t_i} \log S(t_i|t_j; \alpha_{j,i})$$

$$\Psi_3(\mathbf{t}^c; \mathbf{A}) = \sum_{i:t_i \leqslant T} \log \left(\sum_{j:t_j < t_i} H(t_i|t_j; \alpha_{j,i}) \right).$$

If all pairwise transmission likelihoods between pairs of nodes in the network have log-concave survival functions and concave hazard functions in the parameter(s) of the pairwise transmission likelihoods, then convexity of Equations (11) and (12) follows from linearity, composition rules for concavity, and concavity of the logarithm. □

Corollary 2
The static and dynamic networks inference problems defined by Equations (11) and (12) are convex for the exponential, power-law, and Rayleigh models.

Theorem 3
The maximum likelihood estimator $\hat{\alpha}$ given by the solution of Equation (11) is consistent.

Proof Sketch. We check the criteria for consistency of identification, continuity, and compactness (Newey & McFadden, 1994). The log-likelihood in Equation (14) is a continuous function of $\mathbf{A}$ for any fixed set of cascades $\{\mathbf{t}^1 \dots \mathbf{t}^{|C|}\}$, and each α defines a unique function $\log f(\cdot|\mathbf{A})$ on the set of cascades. Finally, note that $L \to -\infty$ for both $\alpha_{ij} \to 0$ and $\alpha_{ij} \to \infty$ for all i, j, so we lose nothing imposing upper and lower bounds, thus restricting to a compact subset.

Similarly, the solution to the maximum a posteriori optimization problem defined by Equation (13) is also unique, computable, and consistent if the prior likelihood on $\mathbf{A}$ is log-concave. In the remainder of the paper, we focus on the maximum likelihood approach for both static and dynamic networks, and we call our network inference method NetRate.

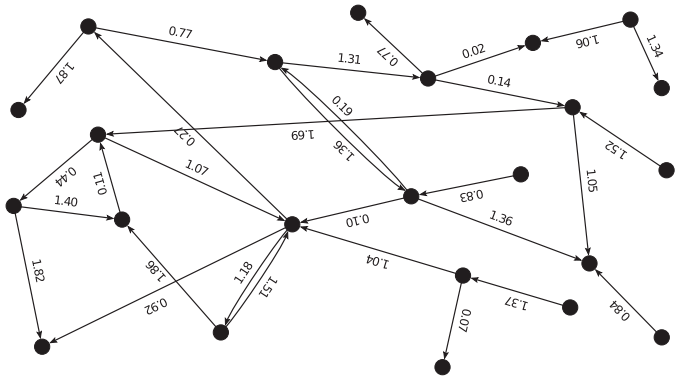
3.1 Properties of NetRate

We highlight some common features of the solutions to the network inference problem for the exponential, power-law, and Rayleigh models. First, to illuminate the discussion, we revisit the terms constituting the log-likelihood Equation (14) for three transmission models in Table 2.

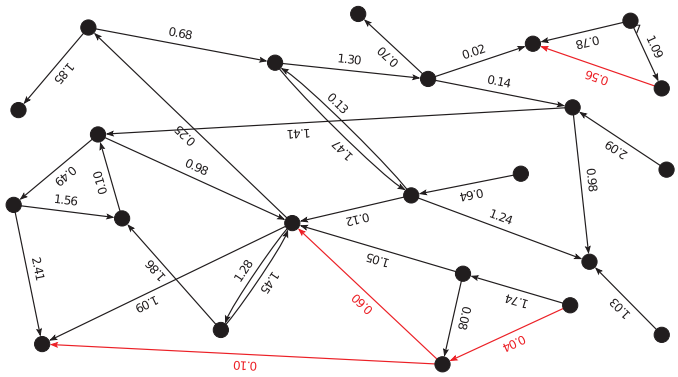
The Ψ_1 and Ψ_2 terms contribute a positively weighted l_1-norm on vector $\mathbf{A}$ that encourages sparse solutions (Boyd & Vandenberghe, 2004). The penalty arises naturally within the probabilistic model so that heuristic penalty terms to encourage sparsity are not necessary. Each term of the l_1-norm is linearly (exponential model), logarithmically (power-law), or quadratically (Rayleigh) weighted by activation times. Sparse solutions are desirable since real networks are usually sparse (Gomez-Rodriguez et al., 2010).

The Ψ_2 term penalizes edges $k \to i$ based on the activation time difference $t_i - t_k$. Edges transmitting activations slowly are heavily penalized and conversely. The Ψ_1 term penalizes edges $i \to j$ targeting *unactivated* nodes j based on the time $T - t_i$ until the observation window cutoff. Lengthening the observation window produces harsher penalties—however, it also allows further activations. The penalties are finite, i.e., if no activation of node j is observed, we can only say that it has survived until time T. There is insufficient evidence to claim that j will never be activated since our data are *right-censored* (Aalen et al., 2008). NetRate does not use empirically ungrounded parameters (such as number of edges k and penalty factor ρ used by NetInf and ConNIe respectively) to leap from not observing an activation to inferring it is impossible. Instead, NetRate infers that the most likely explanation of the observed data does not require transmission across certain edges.

The Ψ_3 term ensures that activated nodes have at least one parent, since otherwise the objective function would be negatively unbounded, i.e., $\log 0 = -\infty$. Moreover, our formulation encourages a natural diminishing property on the number of parents of a node—since the logarithm grows *slowly*, it weakly rewards activated nodes for having many parents. A similar diminishing property on the number of parents of a node has been found in previous work in network inference based on submodular maximization (Gomez-Rodriguez et al., 2010). However, they consider all pairwise transmission rates to be equal, ignoring the temporal dynamics of diffusion.



(a) True network G



(b) Inferred network $\hat{G}$

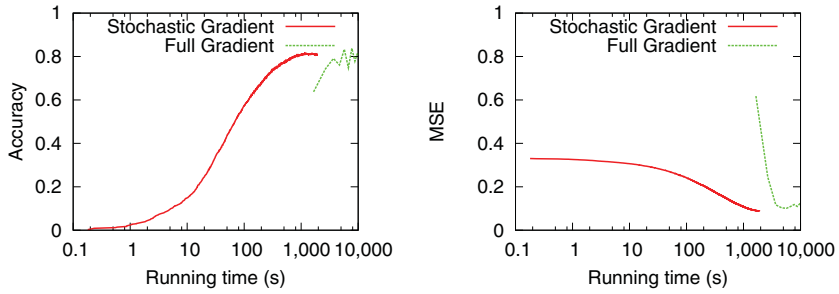
Fig. 2. Accuracy and mean square error (MSE) against running time for a 1,024-node, 3,161-edge static core-periphery Kronecker network with exponential model for 10,000 cascades. Longer running times correspond to more iterations. A stochastic gradient implementation of NetRate is approximately one order of magnitude faster than a full gradient implementation. (color online)

3.2 *Solving* NetRate

Initially, we solved both the static and dynamic networks inference problem using `CVX`, a general-purpose package for specifying and solving convex programs (Grant & Boyd, 2010), and we publicly released an open source implementation.[2] Then, in order to increase scalability, we developed a stochastic gradient descent implementation of our method, which we called InfoPath, and we also publicly released an open source implementation.[3] Figure 2 illustrates how our stochastic gradient implementation of NetRate (also known as InfoPath) is approximately one order of magnitude faster than a full gradient descent implementation. For the sake of fairness, since InfoPath was coded in C++, we compared with a full gradient

[2] A Matlab implementation of NetRate using CVX is available in a supporting website (NetRate, 2011).

[3] A C++ stochastic gradient descent implementation of NetRate, which we called InfoPath, is available in a supporting website (InfoPath, 2013).



(a) Accuracy (b) MSE

Fig. 3. Accuracy of NetRate in a small core-periphery Kronecker network. Panel (a) shows the true network G, and panel (b) shows the inferred network by NetRate from 200 cascades. Red edges denote mistakes, and the number over each edge denotes the (inferred) pairwise transmission rate. NetRate recovers all the true edges and outputs only four false edges. (color online)

(non-stochastic) descent implementation of NetRate in C++ instead of the Matlab code which uses CVX, which was slower.

Stochastic gradient descent methods have been shown to be extremely successful for taking advantage of the structure exhibited by the optimization problems stated in Equations (11) and (12). They have received increasing attention in the machine learning literature (Agarwal & Duchi, 2011; Bach & Moulines, 2011; Blatt et al., 2008; Duchi et al., 2011). Although many convex optimization methods based on stochastic gradient descent have been proposed, we have found that in practice the basic projected stochastic gradient method (Robbins & Monro, 1951) works well enough for our problem. Other more sophisticated methods, such as the stochastic average gradient (Roux et al., 2012) or incremental average gradient (Blatt et al., 2008), do not offer a significant advantage. Therefore, we proceed with the basic stochastic gradient method in the remainder of the paper.

In the static network inference problem defined by Equation (11), the projected stochastic gradient descent method (Robbins & Monro, 1951) uses iterations of the form:

$$\alpha_{j,i}^{k} = \left(\alpha_{j,i}^{k-1} - \gamma_k \nabla_{\alpha_{j,i}} L_{c_k}(\mathbf{A}^{k-1})\right)^{+} \tag{15}$$

where $\nabla_{\alpha_{j,i}} L_{c_k}(\cdot)$ is the gradient of the log-likelihood $L_c(\cdot)$ with respect to the transmission rate $\alpha_{j,i}$, γ_k is a step-size, $(z)^{+} = \max(0, z)$, and cascade c_k is sampled (with replacement) uniformly at random from C. The gradients for all the three edge transmission models are given in Table 3.

In the dynamic network inference problem defined by Equation (12), the projected stochastic gradient descent method (Robbins & Monro, 1951) uses iterations of the form:

$$\alpha_{j,i}^{k}(t) = \left(\alpha_{j,i}^{k-1}(t) - \gamma_k \nabla_{\alpha_{j,i}} L_{c_k}(\mathbf{A}^{k-1}(t))\right)^{+} \tag{16}$$

where $\nabla_{\alpha_{j,i}} L_{c_k}(\cdot)$ is the gradient of the log-likelihood $L_c(\cdot)$ with respect to the transmission rate $\alpha_{j,i}$, γ_k is a step-size, $(z)^{+} = \max(0, z)$, and cascade c_k is sampled (with replacement, not uniformly) from C_t. In this case, instead of using all historic data and then explicitly penalizing each cascade by a different weighting factor $w_c(t)$, we use a different, more scalable approach. We sample cascades with replacement

Table 3. *Cascade gradients for transmission models.*

Model	Cascade gradient for unactivated	Cascade gradient for activated
	$\nabla_{\alpha_{j,i}} L_c(\mathbf{A})$	$\nabla_{\alpha_{j,i}} L_c(\mathbf{A})$
Exp	$T - t_j^c$	$(t_i^c - t_j^c) - \frac{1}{\sum_{k:t_k^c < t_i^c} \alpha_{k,i}}$
Pow	$\log\left(\frac{T-t_j^c}{\delta}\right)$	$\log\left(\frac{t_i^c - t_j^c}{\delta}\right) - \frac{(t_i^c - t_j^c)^{-1}}{\sum_{k:t_k^c < t_i^c} \alpha_{k,i}(t_i^c - t_k^c)^{-1}}$
Ray	$\frac{(T-t_j^c)^2}{2}$	$\frac{(t_i^c - t_j^c)^2}{2} - \frac{t_i^c - t_j^c}{\sum_{k:t_k^c < t_i^c} \alpha_{k,i}(t_i^c - t_k^c)}$

where the probability of a cascade being sampled decays with the age of the cascade. This way recent cascades get sampled more often and thus implicitly hold higher importance when inferring the network. In practice, we achieve a significant speed up using this approach. Moreover, in our dynamic network inference problem, the transmission rates usually vary smoothly. This means that stochastic gradient descent is a natural method since we can use the inferred network from the previous time step as initialization for the inference procedure in the current time step. We find that setting the starting point $\alpha_{j,i}^0$ of each transmission rate $\alpha_{j,i}$ to the last outputted estimate of the transmission rate allow us to further speed up the algorithm.

Importantly, in each iteration k of the projected stochastic gradient method for both static and dynamic networks, we only need to compute the gradients $\nabla_{\alpha_{j,i}} L_{c_k}(\mathbf{A}^k)$ for edges (j,i) such that node j has been activated in cascade c_k, and the iteration cost and convergence rate are independent of $|C|$ (Bach & Moulines, 2011; Nemirovski et al., 2009). Rigorous theoretical analysis of convergence turns out to be a challenging problem, which we leave for future work. However, we would like to point out that such analysis typically assumes the gradients $\nabla_{\mathbf{A}} L_c(\mathbf{A}^k)$ to be either bounded above by a constant M, $||\nabla_{\mathbf{A}} L_c(\mathbf{A})|| \leqslant M$, or Lipschitz-continuous with constant L, $||\nabla_{\mathbf{A}} L_c(\mathbf{A}_2) - \nabla_{\mathbf{A}} L_c(\mathbf{A}_1)|| \leqslant L||\mathbf{A}_2 - \mathbf{A}_1||$. In our problem, these conditions are violated if at any iteration k, there is a node i activated in cascade c_k such that $H(t_i^{c_k} | t_j^{c_k}; \alpha_{j,i}^{k-1}) = 0 \ \forall j : t_j^{c_k} < t_i^{c_k}$, i.e., node i has no parents that *explain* the activation at $t_i^{c_k}$, and the objective function is positively unbounded. In practice, we avoid this scenario by introducing a lower bound on *feasible* transmission rates so that $\alpha_{j,i} \geqslant \varepsilon$. A transmission rate $\alpha_{j,i}$ is *feasible* if there is at least one cascade in which both nodes j and i get activated. When outputting a solution, we simply omit transmission rates with value ε.

3.2.1 *Aging edges in dynamic networks*

Our algorithm automatically penalizes edges (j,i) when the source node j gets activated and the target node i does not. In other words, in each iteration k of the (stochastic) gradient descent method, we update transmission rates $\alpha_{j,i}^k$ if node j gets activated in cascade c_k. Therefore, an edge (j,i) gets penalized if node j gets activated in at least one cascade c_k. In the dynamic setting, we introduce the additional assumption that *unused* edges decay exponentially. In online media, for example, bloggers typically pay less attention to news sites or blogs that have not been activated recently. If a node j has not been activated recently, we would like the

Algorithm 1 Stochastic gradient implementation of NetRate for static networks

Require: C, K
  while $k < K$ **do**
    $c_k \leftarrow$ uniform-sampling(C);
    for all $(j,i) : t_j^{c_k} < t_i^{c_k}$ **do**
      $\alpha_{j,i}^k = \left(\alpha_{j,i}^{k-1} - \gamma_k \nabla_{\alpha_{j,i}} L_{c_k}(\mathbf{A}^{k-1})\right)^+$;
    end for
    k = k+1;
  end while
  $\mathbf{A}^* \leftarrow \mathbf{A}^{K-1}$;
  return $\mathbf{A}^*$;

unused edges (j,i) to decay and eventually vanish, or equivalently the transmission rates $\alpha_{j,i}$ to converge to zero. We incorporate this observation by multiplying the transmission rates of unused edges by an *aging* factor ρ every time t we solve the dynamic network inference problem. Our implementation penalizes edges (j,i) where node j never gets activated. We use an aging factor $\rho = 0.95$ in our experiments.

3.2.2 Cascade sampling in dynamic networks

In Equation (16), instead of sampling cascades uniformly at random and explicitly penalizing each cascade by a different weighting factor $w_c(t)$, we achieve a significant speed up by sampling cascades using a procedure that penalizes old cascades and sets $w_c(t) = 1$ for all cascades. There are many different sampling procedures. For simplicity, we use windowed uniform or windowed exponential sampling. Windowed means that when solving the network inference problem for time t, we only sample cascades that started in the time window $(t - T_s, T_s)$. Here we encounter an important tradeoff. The shorter the sampling time window T_s in the stochastic gradient descent, the quicker our algorithm tracks changes in transmission rates. However, a short sampling time window results in less reliable estimates because we sample fewer cascades. To track changes quickly, we therefore need to observe many cascades over time.

3.2.3 Distributed optimization

The optimization problem splits into N subproblems, one for each node i, in which we find $N-1$ rates $\alpha_{j,i}$, $j = 1, \ldots, N \setminus i$. The computation can be performed in parallel, obtaining local solutions that are globally optimal. Importantly, each node's computation only requires the activation times of other nodes in cascades it belongs to. This allows to scale NetRate beyond hundreds of thousands of nodes.

3.2.4 Unfeasible rates

If a pair (j,i) is not in any common cascades, $\alpha_{j,i}$ only arises in the non-positive term Ψ_3 in Equation (14), so the optimal $\alpha_{j,i}$ is zero. We therefore simply modify the optimization problem by setting $\alpha_{j,i}$ to zero—we remove $\alpha_{j,i}$ from the optimization

Algorithm 2 Stochastic gradient implementation of NetRate for dynamic networks

Require: C_t, K, T, ρ
  while $k < K$ **do**
    $c_k \leftarrow$ cascade-sampling(C_t, T);
    for all $(j,i) : t_j^{c_k} < t_i^{c_k}$ **do**
      $\alpha_{j,i}^k = \left(\alpha_{j,i}^{k-1} - \gamma_k \nabla_{\alpha_{j,i}} L_{c_k}(\mathbf{A}^{k-1})\right)^+$;
    end for
    for all $(j,i) : \alpha_{j,i}^{k-1} > 0, t_j^{c_k} \to \infty$ **do**
      $\alpha_{j,i}^k = \rho \alpha_{j,i}^{k-1}$;
    end for
    k = k+1;
  end while
  $\mathbf{A}^* \leftarrow \mathbf{A}^{K-1}$;
  return $\mathbf{A}^*$;

problem. In a network with hundreds of thousands of nodes (and billions of edges), this tweak can speed up inference by several orders of magnitude.

4 Experimental evaluation on synthetic data

In this section, we validate NetRate by evaluating its performance on static and dynamic synthetic networks that mimic the structure of social networks. In the next section, we will perform a large-scale real-world evaluation, and present some qualitative analysis of the dynamics of real-world online networks.

We first describe the experimental setup that we used for static and dynamic networks. Second, we compare the performance of NetRate with the state of the art in static networks. Third, we analyze its performance in static networks as a function of cascade coverage, time horizon, transmission rate distributions, exogenous factors, noise, and thresholding. Finally, we analyze the performance of NetRate in dynamic networks as a function of the transmission rate temporal trend, and as a function of the sampling window when using the stochastic gradient descent implementation, InfoPath.

4.1 Experimental setup

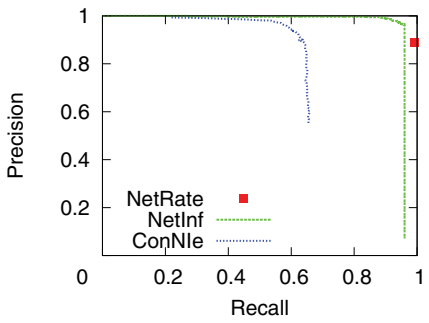
We focus on synthetic networks that mimic the structure of real-world diffusion networks—in particular, social networks. We consider two models of directed real-world social networks: the Forest Fire (scale free) model (Barabási & Albert, 1999) and the Kronecker Graph model (Leskovec et al., 2010) to generate diffusion networks. We generate three types of Kronecker Graph models with very different structures: random (Erdős & Rényi, 1960) (parameter matrix [0.5, 0.5; 0.5, 0.5]), hierarchical (Clauset et al., 2008) ([0.9, 0.1; 0.1, 0.9]), and core-periphery (Leskovec et al., 2008) ([0.9, 0.5; 0.5, 0.3]). First, we consider static networks with fixed transmission rates over time. We generate a static network G^* using either the Forest Fire or the Kronecker Graph model, and draw transmission rates for edges (j,i) from a uniform distribution, a Gaussian distribution or a Rayleigh distribution. We control

the transmission rate variance across edges in the network by tuning the parameter values of the distributions. The transmission rate for an edge (j, i) models how fast the information spreads from node j to node i in social networks. If not specified, $\alpha \sim U(0.01, 1)$ for the exponential and Rayleigh models and $\alpha \sim U(0.01, 2)$ for the power-law. Then we generate a set of cascades over G^*. Root nodes of cascades are chosen at random. Once a node is activated, the transmission likelihoods of outgoing edges determine the activation times of its neighbors. We record the time of the first activation if a node is activated more than once. Activations are not observed after a pre-specified time horizon T. Then, given these activation times (i.e., set of cascades), we aim to recover G^* using NetRate. For example, Figure 3(a) shows a small diffusion network G^* of 23 nodes and 30 directed edges. Using the exponential model we generated 200 cascades. Now, given the cascades, NetRate returns the network $\hat{G}$ in Figure 3(b). Our method recovered G^* almost perfectly by making only four errors (red edges), and it outputs pairwise transmission rates (numbers over edges) that are very close to the true values.

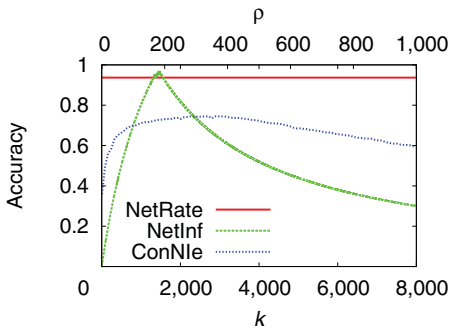
Then we consider dynamic networks with variable transmission rates over time. We make every edge of each network G^* to follow a particular edge transmission rate evolution pattern to obtain time-varying networks, $G^*(t)$. We consider five edge evolution patterns: Slab, Square, Chainsaw, Hump, and Constant (see Figure 12). Slab and Hump patterns model outgoing connections of sites that become popular for a short period of time. Square and Chainsaw patterns model incoming connections to sites that perform updates periodically at specific times of the day or specific days of the week. Constant pattern represents connections between sites that interact at any time and during a long period of time, usually large media sites. We consider Chainsaw, Hump, and Constant to be examples of *Type I* pattern, without discontinuities, and Slab and Square to be examples of Type II pattern, with discontinuities. Then we assign to each edge in the network an evolution pattern chosen uniformly at random from the set of the above five patterns. Then we generate transmission rate values $\alpha^*_{j,i}(t)$ for each edge according to its chosen evolution pattern. The evolving edge transmission rate $\alpha^*_{j,i}(t)$ models how quickly information spreads from one node to another. Finally, we generate 1,000 information cascades per time step. For each cascade we randomly pick the cascade root node. Given the node activation times from the recorded cascades, our goal then is to find the true edges of the network, and for each edge discover its transmission rate evolution pattern. In other words, inferring how each edge transmission rate $\alpha(t)$ evolves over time.

4.2 Performance in static networks

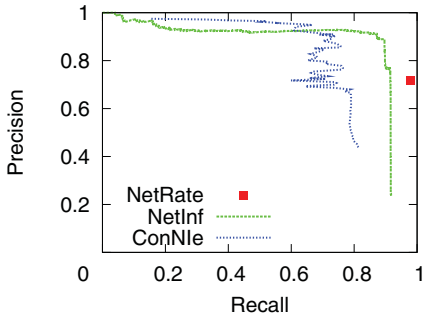
First, we evaluate NetRate against two state-of-the-art inference methods, NetInf and ConNIe, in static networks by comparing the inferred and true networks via three measures: precision, recall, and accuracy. Precision is the fraction of edges in the inferred network $\hat{G}$ present in the true network G^*. Recall is the fraction of edges of the true network G^* present in the inferred network $\hat{G}$. Accuracy is $1 - \frac{\sum_{i,j} |I(\alpha^*_{i,j}) - I(\hat{\alpha}_{i,j})|}{\sum_{i,j} I(\alpha^*_{i,j}) + \sum_{i,j} I(\hat{\alpha}_{i,j})}$, where $I(\alpha) = 1$ if $\alpha > 0$ and $I(\alpha) = 0$ otherwise. Inferred networks with no edges or only false edges have zero accuracy. Second, we evaluate how accurately NetRate infers transmission rates over edges by computing the



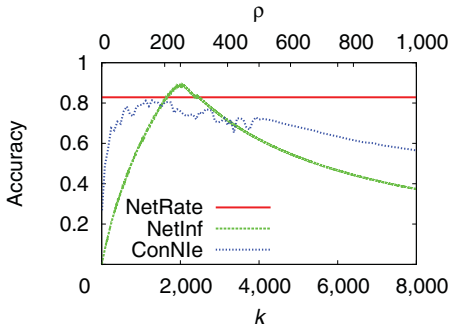
(a) Precision-recall (Hierarchical, EXP)



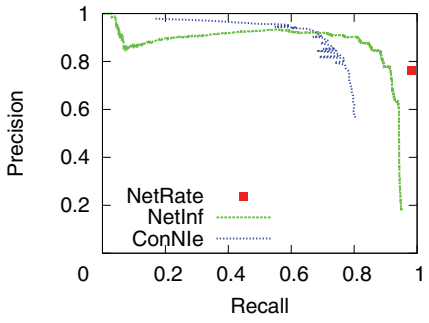
(b) Accuracy (Hierarchical, EXP)



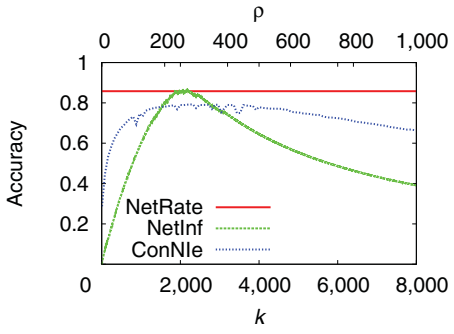
(c) Precision-recall (Random, RAY)



(d) Accuracy (Random, RAY)



(e) Precision-recall (Forest Fire, POW)



(f) Accuracy (Forest Fire, POW)

Fig. 4. Panels (a,c,e) plot precision against recall; panels (b,d,f) plot accuracy. For CONNIE and NETINF we sweep over parameters ρ (penalty factor) and k (number of edges) respectively to control the solution sparsity in both algorithms, thereby generating a family of inferred models. NETRATE has no tunable parameters and therefore yields a unique solution. (a,b): 1,024-node hierarchical Kronecker network with exponential model for 5,000 cascades. (c,d): 1,024-node random Kronecker network with Rayleigh model for 2,000 cascades. (e,f): 1,024-node Forest Fire network with power law model for 5,000 cascades. (color online)

normalized MAE (i.e., $E\left[|\alpha^* - \hat{\alpha}|/\alpha^*\right]$, where α^* is the true transmission rate and $\hat{\alpha}$ is the estimated transmission rate).

Figure 4 compares the precision, recall, and accuracy of NETRATE with NETINF and CONNIE for two types of static Kronecker networks: hierarchical community structure with exponential model for 5,000 cascades and random with Rayleigh

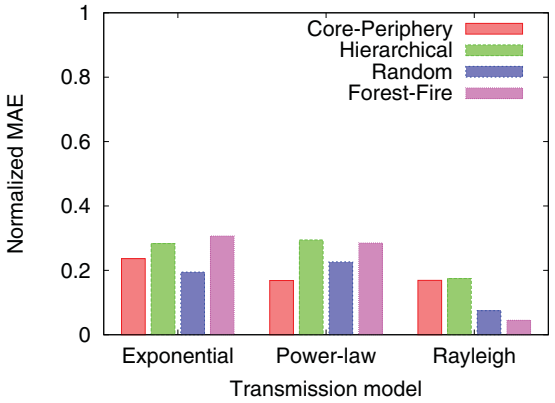


Fig. 5. Normalized mean absolute error (MAE) of NetRate for three types of Kronecker networks (1,024 nodes and 2,048 edges) and a Forest Fire network (1,024 edges and 2,422 edges) for 5,000 cascades. We consider all three models of transmission likelihoods: exponential (Exp), power-law (Pow), and Rayleigh (Ray). (color online)

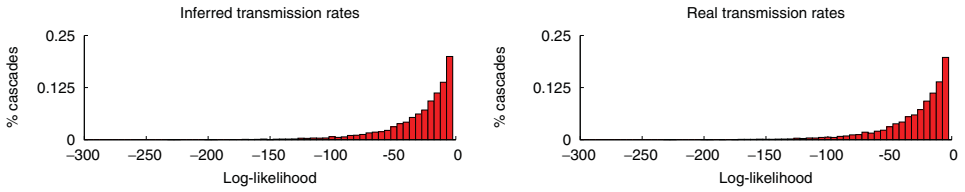
model for 2,000 cascades, and a static Forest Fire network with power-law model for 5,000 cascades over an observation window of length $T = 10$. In terms of precision-recall, NetRate outperforms ConNIe and NetInf for all the synthetic examples in the Pareto sense (Boyd & Vandenberghe, 2004). More specifically, if we set ConNIe's and NetInf's tunable parameters to provide solutions with the same precision as NetRate, NetRate's recall is always higher than the other two methods. Strikingly, ConNIe and NetInf do not achieve NetRate's recall for any precision value. NetRate outperforms ConNIe with respect to accuracy for any penalty factor ρ in all synthetic examples. It is also more accurate than NetInf for most values of k (number of edges). Importantly, NetInf and ConNIe yield a curve of solutions from which we have to select a point blindly (or at best heuristically), whereas NetRate yields a unique solution without any tuning.

Figure 5 shows the normalized MAE of the estimated transmission rates for the same networks, computed on 5,000 cascades. The normalized MAE is under 25% for almost all networks and transmission models—surprisingly low given we are estimating more than 2,000 non-zero real numbers.

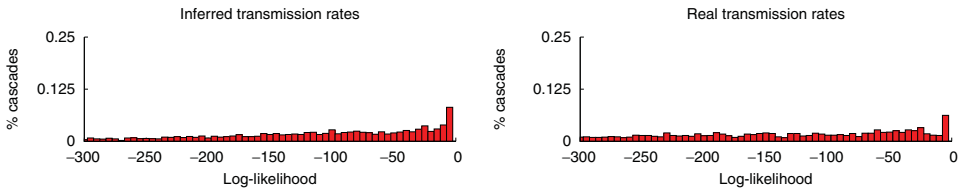
4.3 Solution quality

Given a diffusion network, we may expect that some cascades are more likely than others. Moreover, we would like that NetRate outputs inferred networks that produce the same cascade likelihoods as the ones given by the true networks. Therefore, we now compare the log-likelihood per cascade for true and inferred networks for different networks and transmission models.

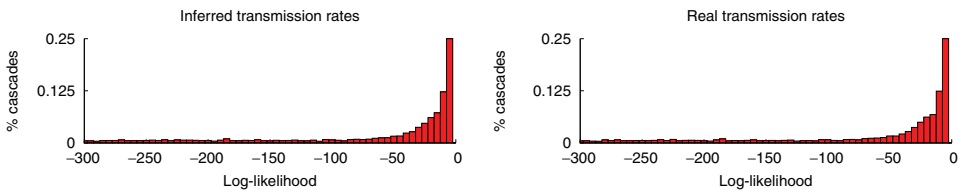
Figure 6 plots the distribution of log-likelihoods of the set of cascades that we used for network inference in the previous section. We compute the distribution of the log-likelihoods of the cascades for true and inferred networks. We observe that the distribution of log-likelihoods across cascades depends on the type of network and the transmission model. Both the hierarchical Kronecker with exponential model and the Forest Fire with power-law model result in many cascades having a high



(a) Hierarchical, EXP



(b) Random, RAY



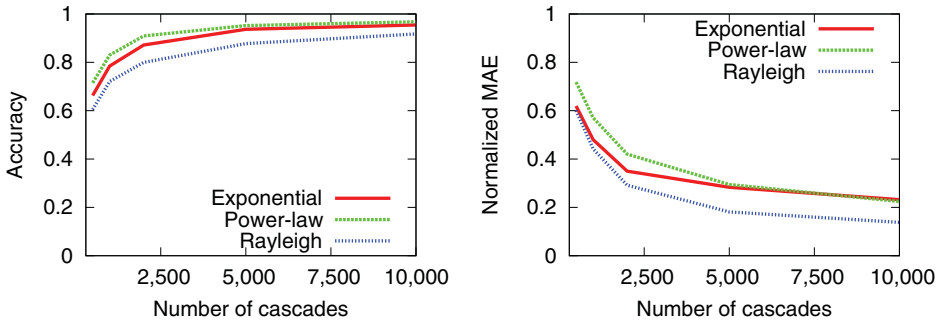
(c) Forest-Fire, POW

Fig. 6. Distribution of the log-likelihood of the cascades for (a) 5,000 cascades in a hierarchical Kronecker network (1,024 nodes, 2,048 edges) with exponential model, (b) 2,000 cascades in a random Kronecker network (1,024 nodes, 2,048 edges) with Rayleigh model, and (c) 5,000 cascades in a Forest Fire network (1,024 edges and 2,422 edges) with power-law model over an observation window of length $T = 10$. We compare the log-likelihoods of the cascades for true networks and inferred networks. All networks are static. (color online)

likelihood, especially in the case of the Forest Fire with power-law model, and a rapid decay of the number of cascades with the log-likelihood value. In contrast, the random Kronecker with Rayleigh model produces a set of cascades with log-likelihood values covering uniformly a much wider range. The distribution of the log-likelihoods of the cascades is always very similar for real and inferred networks.

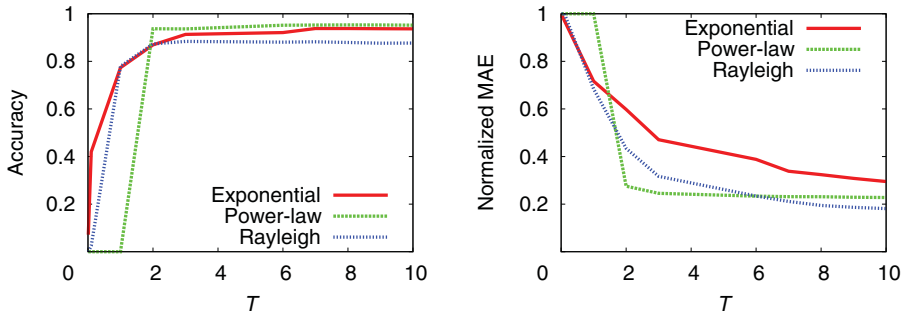
4.4 Performance versus cascade coverage

Observing more cascades leads to higher precision-recall and more accurate estimates of transmission rates. Figure 7 plots the accuracy and normalized MAE of estimated transmission rates against the number of observed cascades for a static hierarchical Kronecker network with all three transmission models over an observation window of length $T = 10$. Estimating transmission rates is considerably harder than simply discovering edges, and therefore more cascades are needed for accurate estimates. As many as 5,000 cascades are required to obtain normalized MAE values lower than 20%. Up to 5,000 cascades, the normalized MAE decreases quickly as a function of



(a) Accuracy (b) Normalized MAE

Fig. 7. Performance of NetRate versus cascade coverage for a static hierarchical Kronecker network (1,024 nodes and 2,048 edges) with exponential, power-law, and Rayleigh transmission models over an observation window of length $T = 10$. (color online)



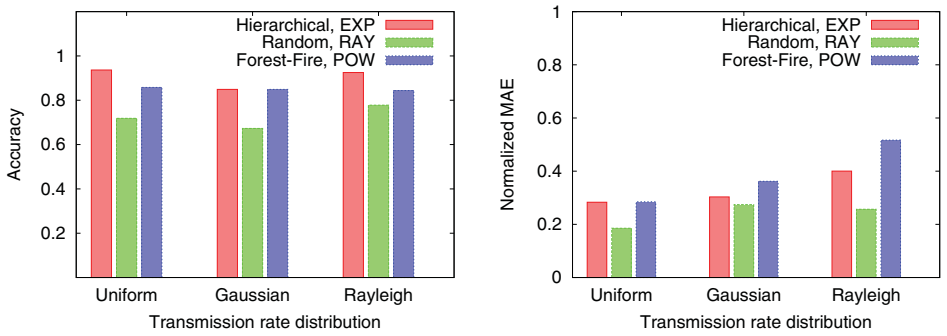
(a) Accuracy (b) Normalized MAE

Fig. 8. Performance of NetRate versus time horizon for a static hierarchical Kronecker network (1,024 nodes and 2,048 edges) with exponential, power-law, and Rayleigh transmission models. (color online)

the number of cascades. Beyond 5,000 cascades, it becomes more difficult to decrease further the normalized MAE by adding cascades.

4.5 Performance versus time horizon

Intuitively, the longer the observation window, the more accurately NetRate infers transmission rates. Figure 8 confirms this intuition by showing the accuracy and normalized MAE of estimated transmission rates for different time horizons T for a static hierarchical Kronecker with exponential, power-law, and Rayleigh transmission models for 5,000 cascades. The longer the time horizon T, the weaker the *right-censoring* in the diffusion data and the more accurately NetRate infers the transmission rates. However, once we reach a sufficiently long time horizon T, further increasing the recording time does not increase the performance significantly, since there are no unrecorded activations anymore.



(a) Accuracy (b) Normalized MAE

Fig. 9. Performance of NetRate versus transmission rate distribution. Panels plot (a) accuracy and (b) normalized MAE of the estimated transmission rates against the transmission rate distribution for a hierarchical Kronecker network (1,024 nodes and 2,048 edges) with exponential model for 5,000 cascades, a random Kronecker network (1,024 nodes and 2,048 edges) with Rayleigh model for 2,000 cascades, and a Forest Fire network (1,024 nodes and 2,422 edges) with power-law model for 5,000 cascades over an observation window of length $T = 10$. All networks are static. (color online)

4.6 Performance versus transmission rate distribution

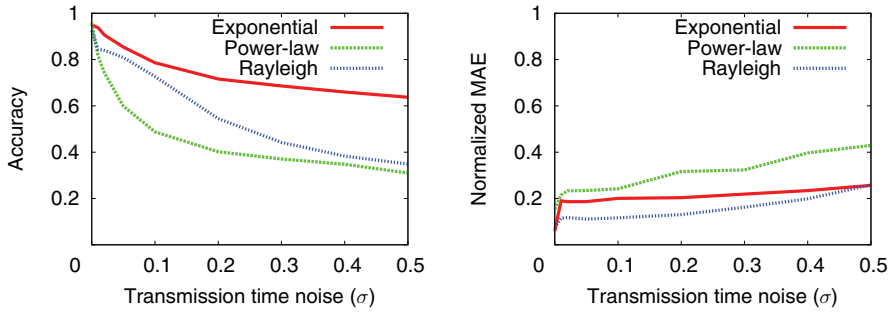
We have carried out experiments using synthetic networks in which the transmission rates of the edges are always drawn from a uniform distribution. Since this assumption may be often violated in real networks, we now consider networks in which we set the transmission rates of the edges by drawing samples from (i) a uniform distribution, (ii) a Gaussian distribution ($\mu = 0.5$, $\sigma = 0.5$; we reject any negative samples), and (iii) a Rayleigh distribution ($\sigma = 0.25$).

Figure 9 plots accuracy and normalized MAE of the estimated transmission rates against the transmission rate distribution for a static hierarchical Kronecker network with exponential model for 5,000 cascades, a static random Kronecker network with Rayleigh model for 2,000 cascades, and a static Forest Fire network with power-law model for 5,000 cascades over an observation window of length $T = 10$. In all networks, the accuracy remains relatively stable across transmission rate distributions. However, the more skewed the transmission rate distribution, the greater the normalized MAE (i.e., it is easier to estimate transmission rates drawn from a uniform distribution than from a Gaussian or a Rayleigh distribution).

4.7 Performance versus transmission time noise

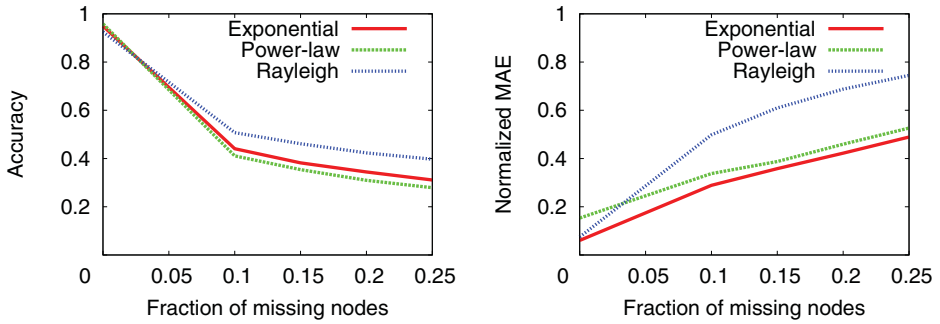
When we work with real data, it may happen that the true pairwise transmission likelihoods differ from the parametric models we assume, or that the observed activation times may have been corrupted by noise. We then study the accuracy and normalized MAE of NetRate as a function of the noise of the transmission times between activations. To this end, we add Gaussian noise to the transmission times between activations in the cascade generation process.

Figure 10 shows the accuracy and normalized MAE against the amount of Gaussian noise added to the transmission times between activations for a static random Kronecker network with exponential, power-law, and Rayleigh transmission



(a) Accuracy (b) Normalized MAE

Fig. 10. Performance of NetRate versus amount of additive Gaussian noise (standard deviation σ) in the transmission times for a static random Kronecker network (1,024 nodes and 2,048 edges) with exponential, power-law, and Rayleigh transmission models over an observation window of length $T = 10$. (color online)



(a) Accuracy (b) Normalized MAE

Fig. 11. Performance of NetRate versus fraction of missing nodes per cascade for a static random Kronecker network (1,024 nodes and 2,048 edges) with exponential, power-law, and Rayleigh transmission models over an observation window of length $T = 10$. (color online)

models for 5,000 cascades. In all three transmission models, the normalized MAE (i.e., transmission rate inference) is more robust against noise than the accuracy (i.e., network structure inference).

4.8 Performance versus missing activations

In many real-world scenarios, we do not observe all nodes that become activated during the observation window. For example, media sites and blogs may publish contents that only subscribers can read and members of a social network can restrict the visibility of certain posts. Therefore, we consider collections of cascades where a random fraction of each cascade is missing. This means that we first generate a set of cascades, but then only record node activation times of a fraction of nodes.

Figure 11 shows the accuracy and normalized MAE against the fraction of missing nodes per cascade for a static random Kronecker network with exponential, power-law, and Rayleigh transmission models for 5,000 cascades. Missing data degrade the performance of NetRate significantly, more than noise. Although there has been increasing effort devoted to correcting for missing data in information cascades,

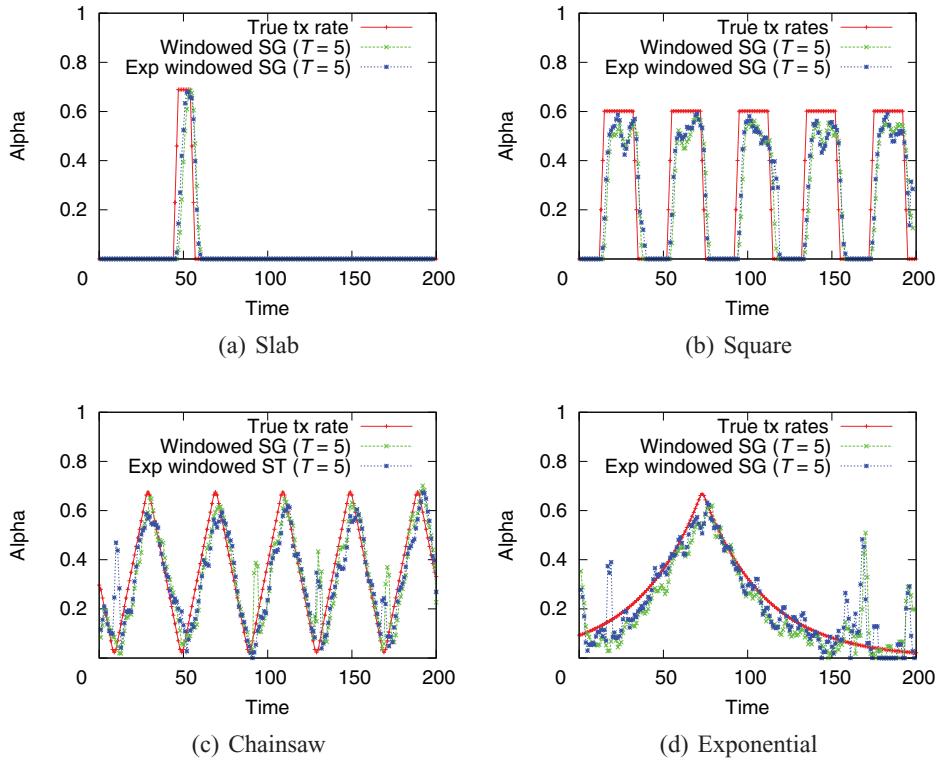


Fig. 12. True and inferred transmission rates over time for edges with different transmission rate trends for a 512-node, 1,024-edge core-periphery Kronecker network with exponential model for 200 time units with 1,000 cascades per time unit. Our method is able to track the changing transmission rate values over time. It works better when the transmission rate trend is continuous (c,d) than when there is discontinuity (a,b). (color online)

previous algorithms attempt to output cascades with the same structural properties of the original (complete) cascades from the incomplete cascades (Sadikov et al., 2011), or to simply estimate the cascade width and length (Chierichetti et al., 2011), but the inferred cascades may be actually very different from the original cascades. It remains an open problem how to correct for missing data in the context of network inference.

An interesting open question is whether *localized* missing observations can be detected. For example, is it possible to detect when certain memes are suppressed on specific websites or in specific regions?

4.9 Performance in dynamic networks

In this section, we evaluate the performance on NetRate in dynamic (time-varying) networks. We first show qualitatively how our algorithm performs for different transmission rate trends, and then evaluate quantitatively its performance.

Figure 12 shows the true and inferred transmission rates for four different edges, each with a different evolution pattern, Slab, Square, Chainsaw, and Humb, in a 512-node, 1,024-edge core-periphery Kronecker network with 20% of the edges following each of the five rate trends. We generated and recorded an average of

1,000 cascades per time unit using an exponential pairwise transmission model. Our method is able to track the evolving edge transmission rate over time for all evolution patterns. It gives near perfect performance when edge transmission rate evolves continuously (Chainsaw, Hump). Interestingly, even when the edge transmission rate evolves discontinuously (Slab, Square), InfoPath manages to track it. Now we compute four different measures: precision, recall, and accuracy of inferred edges as well as mean squared error (MSE) in the edge transmission rate in order to evaluate the performance of our algorithm quantitatively. Precision at time t is the fraction of edges in the inferred network $\hat{G}(t)$ present in the true network $G^*(t)$. Recall at time t is the fraction of edges of the true network $G^*(t)$ present in the inferred network $\hat{G}(t)$. Accuracy at time t is defined as

$$1 - \frac{\sum_{i,j} |I(\alpha^*_{i,j}(t)) - I(\hat{\alpha}_{i,j}(t))|}{\sum_{i,j} I(\alpha^*_{i,j}(t)) + I(\hat{\alpha}_{i,j}(t))}$$

where $\alpha^*(t)$ is the true transmission rate at time t, $\hat{\alpha}(t)$ is the estimated transmission rate at time t, and $I(\alpha(t)) = 1$ if $\alpha(t) > 0$, and $I(\alpha(t)) = 0$ otherwise. Inferred networks with no edges or only false edges have zero accuracy. Last, MSE at time t is defined as $E\left[||\alpha^*(t) - \hat{\alpha}(t)||^2\right]$, where $\alpha^*(t)$ is the true transmission rate at time t and $\hat{\alpha}(t)$ is the estimated transmission rate.

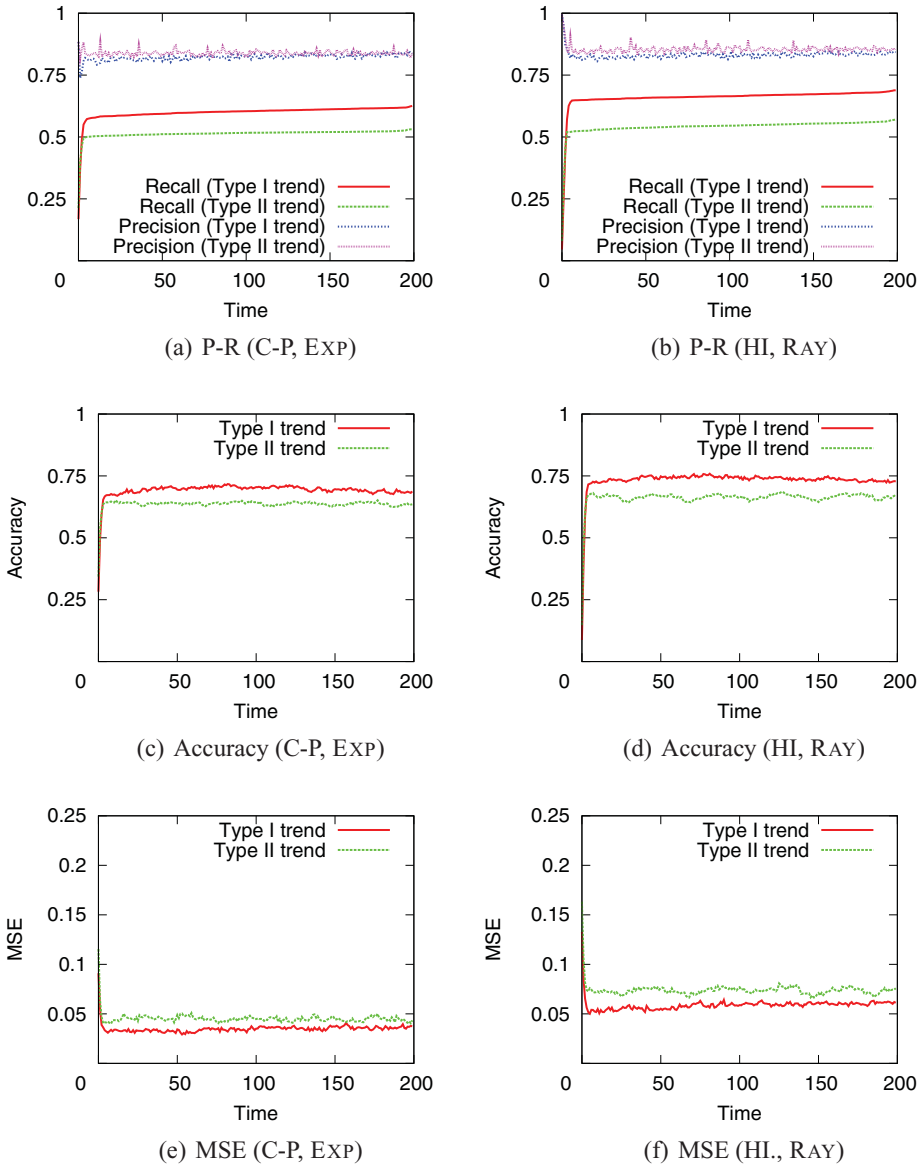
Figure 13 shows precision, recall, accuracy, and MSE over time for two 1,024-node, 2,048-edge time-varying Kronecker networks, core-periphery (parameter matrix $[0.9, 0.5; 0.5, 0.3]$) and hierarchical (Clauset et al., 2008) ($[0.9, 0.1; 0.1, 0.9]$), with exponential and Rayleigh pairwise transmission models respectively. We generated continuous (Chainsaw, Hump) and discontinuous (Slab, Square) evolution patterns for transmission rates, $\alpha^*_{j,i}(t) \in [0, 1]$ for all t, and we recorded 1,000 cascades per unit time. The performance of our method is stable across time, and as noted qualitatively, continuous trends are easier to track and estimate than discontinuous ones.

4.10 Performance versus sampling time window

Intuitively, the shorter the sampling time window T_s in the stochastic gradient descent implementation, the quicker our algorithm tracks changes in transmission rates in a dynamic network. However, a short sampling time window results in less reliable estimates because we sample fewer cascades.

Figure 14(a) shows the true and inferred transmission rates for a transmission rate which evolves as a Slab for different sampling time window lengths. The experimental results support the intuition. We observe that the shorter the sampling time window, the quicker we are able to track the step-up. However, when the sampling time window is too short, stochastic gradient descent does not sample cascades with activations of the source of the edge, and the rate decays only by aging.

Figure 14(b) shows accuracy across time for different sampling time window lengths for a 512-node, 1,024-edge time-varying core-periphery Kronecker network. Half of the edges have transmission rates that evolve as a Slab, and the other half of the edges have a constant transmission rate. We generated and recorded an average of 1,000 cascades per time unit using an exponential pairwise transmission model. Too short or too long sampling time windows result in lower accuracy.



(a) P-R (C-P, EXP) (b) P-R (HI, RAY)

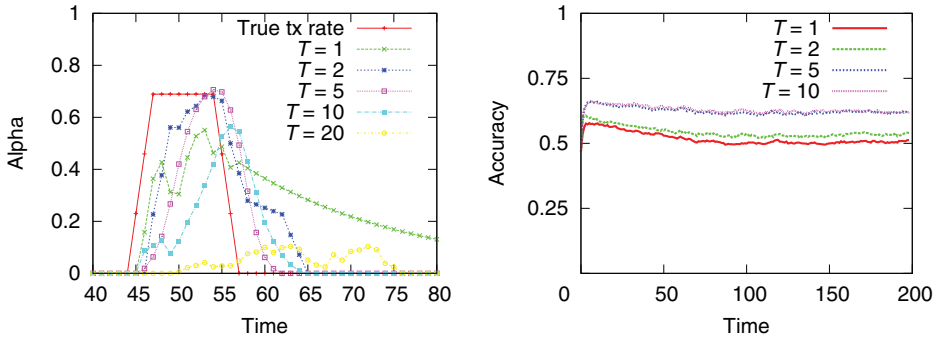
(c) Accuracy (C-P, EXP) (d) Accuracy (HI, RAY)

(e) MSE (C-P, EXP) (f) MSE (HI., RAY)

Fig. 13. Precision and recall (P–R), accuracy and mean square error (MSE) of our stochastic method against time. (a,c,e): 1,024-node, 2,048-edge time-varying core-periphery (C-P) Kronecker network with exponential model; (b,d,f): 1,024-node, 2,048-edge time-varying hierarchical (HI) Kronecker network with Rayleigh model. In both networks, type I (Chainsaw, Hump) and type II (Slab, Square) trends for transmission rates were generated, and 1,000 cascades per unit time were recorded. (color online)

5 Application to real-world data

In this section, we analyze dynamic networks based on real diffusion data, since information pathways change over time, depending upon the information content that propagates through them (Romero et al., 2011; Myers et al., 2012). For example, a real-world event may occur for a limited period of time and thus news related to the event spread quicker and to larger parts of the network in such a time period. At



(a) $\alpha_{j,i}$ with slab trend (b) Accuracy

Fig. 14. Performance versus sampling time window. Panel (a) shows the true and inferred transmission rates for a transmission rate with a Slab evolution pattern for different sampling time window lengths. Panel (b) shows accuracy across time for different sampling time window lengths for a 512-node, 1,024-edge time-varying core-periphery Kronecker network. Half of the edges have transmission rates that follow a Slab evolution pattern, and the other half of the edges have a constant transmission rate. We generated and recorded an average of 1,000 cascades per time unit using an exponential pairwise transmission model. (color online)

any given time, different real-world events, topics, and content propagates through the Web, leading to different emerging and vanishing information pathways, and thus an underlying time-varying network. In order to better understand these temporal changes, we aim to reconstruct time-varying networks and the information pathways for particular real-world events and topics. All the data, code and, additional results are available at the supporting websites (NetRate, 2011; InfoPath, 2013).

5.1 Dataset description

We use more than 300 million blogs and news articles from 3.3 million blogs and news media sites over a period of one year, from March 2011 to February 2012, available at the website (InfoPath, 2013). We trace the flow of information using short textual phrases (such as, "lipstick on a pig") that travel through the Web, which act as tracers for *memes* (Leskovec et al., 2009). A meme is an idea, behavior, or style that spreads from person to person within a culture (*Merriam-Webster's Collegiate Dictionary*, 2004). We consider each meme m as a separate cascade c_m. Since all documents that contain memes are time-stamped, a cascade c_m is simply a record of the times when sites first mentioned meme m. We extracted more than 179 million memes, longer than four words. Out of these, 34 million distinct memes appeared at least twice, resulting in 34 million meme cascades.

5.2 Experimental setup

Our aim is to consider sites that actively spread memes over the Web. We achieve this by selecting top 5,000 sites in terms of the number of memes they mentioned. Moreover, we are interested in inferring dynamic networks related to particular topics or events. So we assume, we are also given a keyword query Q related to the event/topic of interest. When inferring a network for a given query Q, we only

Table 4. *Topic and news world event statistics.*

Topic or news event	# Sites	# Contagions
Amy Winehouse	1,207	109,650
Fukushima	1,666	383,745
Gaddafi	1,358	440,646
Kate Middleton	1,427	191,777
NBA	2,087	1,543,630
Occupy	1,875	655,183
Strauss-Kahn	1,263	204,238
Syria	1,565	615,176

consider documents (and the memes they mention) that include keywords Q. Then we build information cascades using only those memes and apply our algorithm to infer the edges and evolving edge transmission rates. The edge transmission rates explain the propagation of information related to a given topic or real-world event Q. For each query Q we infer one network per day. Table 4 summarizes the number of sites and meme cascades for several topics and real-world events.[4]

5.3 Implementation and scalability

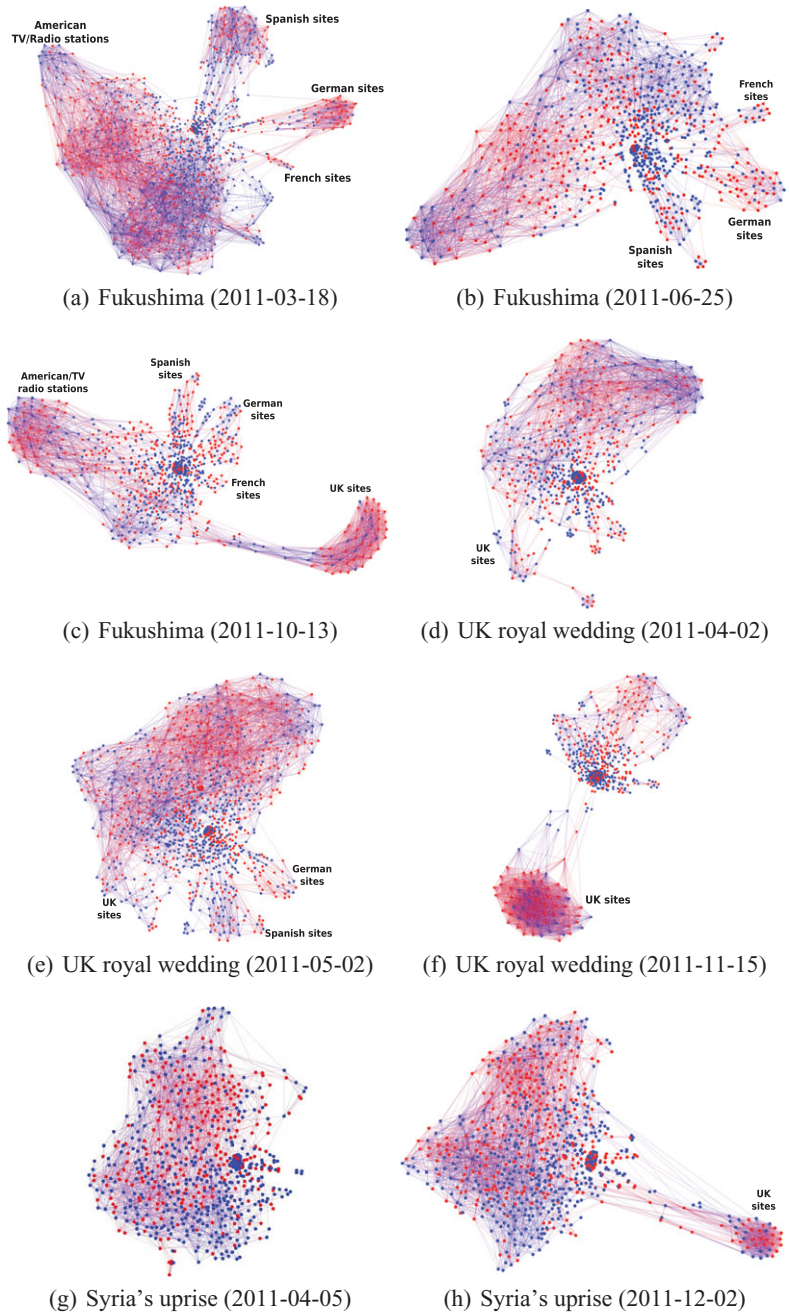
We developed an efficient distributed implementation of our algorithm using stochastic gradient descent in C++, which uses the graph library SNAP (SNAP, 2012). We deployed the implementation in a cluster with 1,000 CPU cores and 6-TB RAM. With this setup, we inferred 38 time-varying networks, one per topic or news world event, with a daily resolution for a period of one year from March 2011 to February 2012. Despite having thousands of nodes and hundreds of thousands of cascades, we inferred all networks in less than four hours. Note that inferring 38 time-varying with a daily resolution for a one-year period is equivalent to solving Equation (12) more than 13,000 times (38×365) for millions of pairwise transmission rates. We also tested our algorithm on larger datasets. For example, for "Occupy Wall Street movement," we were able to infer a 43,415-node time-varying network over a period of 18 months, from January 2011 to June 2012, using 1,381,793 information cascades.

5.4 Visualizing the information pathways

Figure 15 plots diffusion networks for three different 2011 world events: Fukushima nuclear disaster, UK royal wedding, and civil uprise in Syria. Each network is shown at three different time points. Red nodes represent mainstream media sites, and blue nodes represent blogs (Leskovec et al., 2009).

Based on the figure, we draw several interesting observations. Most often, information propagates through a core-periphery network structure. Such structure emerges by few central media sites and blogs driving the adoption of memes across

[4] Additional time-varying diffusion networks for other topics and news events are available at the supporting website (InfoPath, 2013).



(a) Fukushima (2011-03-18) (b) Fukushima (2011-06-25)

(c) Fukushima (2011-10-13) (d) UK royal wedding (2011-04-02)

(e) UK royal wedding (2011-05-02) (f) UK royal wedding (2011-11-15)

(g) Syria's uprise (2011-04-05) (h) Syria's uprise (2011-12-02)

Fig. 15. Dynamic diffusion networks for different 2011 world events. Red nodes are mainstream media, and blue nodes are blogs. Additional plots for other topics and news events, and time-varying diffusion networks at a daily resolution are available at the supporting website (InfoPath, 2013). (color online)

the Web (Gomez-Rodriguez et al., 2010). However, the network structure often changes dramatically over time, and we find clusters that emerge and vanish in short periods of time. For example, the information networks for Syria's uprise illustrated in Figures 15(g) and (h) do nothave any clear clustering structure. However, on

December 2, 2011 (Figure 15(h)) a cluster suddenly emerges in the network. Further investigation reveals that the cluster comprises UK news sites and blogs that discuss recently implemented EU sanctions against Syria. Generally, it is common to observe sudden formation of clusters of sites from specific geographical areas. This is especially noticeable in the information network for Fukushima's disaster, in Figures 15(a)–(c). Such clusters are often formed due to language boundaries, since such boundaries prevent memes to flow across countries or continents. Moreover, we often observe that such clusters are caused by a common external event (Myers et al., 2012), such as the case of UK discussion on EU sanctions against Syria. Inferred dynamic networks can thus be used to investigate the flow of information as well as to detect external events that cause sudden perturbations to the diffusion network structure.

5.5 Evolution of edge transmission rates

Next, we aim to study the evolution of links among different types of sites. We label the nodes in our network as mainstream media and blog, and compute the number of links between different types of sites over time. Figure 16 gives results for several inferred diffusion networks for different topics and world events. We note several interesting patterns.

The connectivity changes tend to reflect the amount of attention that a news event or a topic triggers over time. Unexpected news events, such as the sex scandal of the director of the International Monetary Fund, Strauss-Kahn, on May 14, 2011 in Figure 16(g), or the death of British singer Amy Winehouse on July 23, 2011 in Figure 16(a), result in a dramatic increase in the number of edges over a short period of time. More general topics, such as the NBA in Figure 16(e), result in a network with more stable connectivity over time. Certain types of news are sometimes spreading earlier among blogs than mainstream media. This is especially the case for population-wide events such as the Fukushima nuclear disaster, civil war in Libya, and civil uprise in Syria (Figures 16(b), (c), and (h)). However, it happens more frequently that the largest amount of links are mainstream media to mainstream media, and the fewest links point from blogs to mainstream media. These results are intuitive and consistent with previous works (Gomez-Rodriguez et al., 2010; Leskovec et al., 2009) that observed that most often information flows from mainstream media to blogs (and rarely the other way around). However, as we see here for population-level events and social movements (such as in case of the civil unrest in the Middle East), social media plays a crucial role in information dissemination and organization of civil movements.

5.6 Evolution of node centrality

Having studied the dynamics of edges in the network, we now move toward investigating the network centrality of blogs and mainstream media sites over time for different topics and world events. To measure network centrality of node S in the network at time t, we first compute the shortest path length from S to any other node R in the network. Then centrality of node S is defined as $\sum_R 1/d(S,R)$, where $d(S,R)$ is the shortest path length from S to R (if R is not reachable from S, then

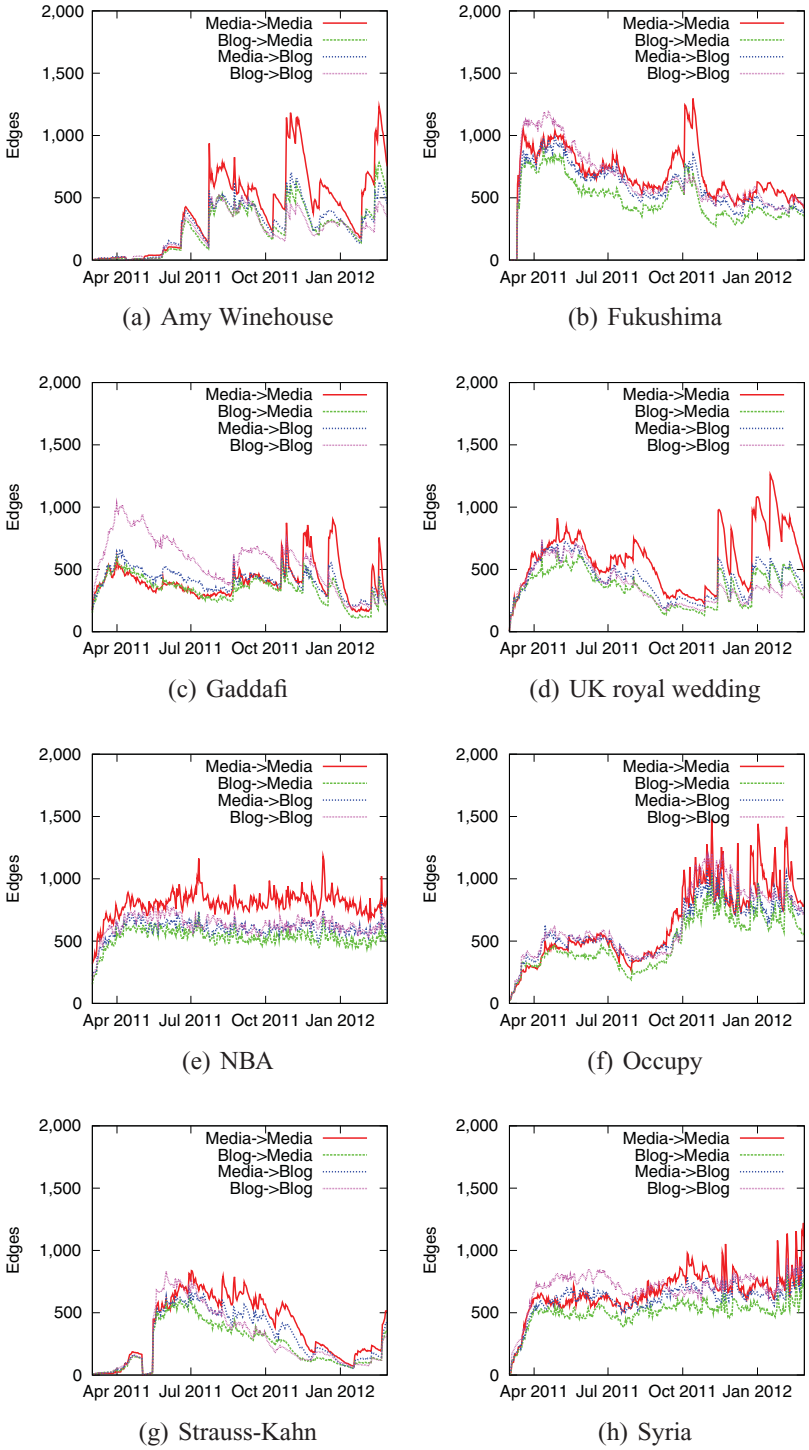


Fig. 16. Total number of links and number of links that point between different types of sites across time for several inferred diffusion networks for eight different topics or 2011 world events. We split the sites into mainstream media and blogs. (color online)

$d(S, R) = \infty$). For networks with core-periphery structure, nodes with high centrality are typically located in the "central" core of the network.

Figure 17 plots the percentage of blogs among the top-100 most central sites over time for eight different topics/events of 2011. Perhaps surprisingly, we observe there is about the same number of mainstream media and blogs in the top-100 most central nodes for most networks—the number of blogs in the top-100 does not typically decreases below 30% or increases over 70%. For some topics, mainstream media are always more *central* (e.g., baseball and NBA in Figures 17(a) and (b)). In contrast, for other topics, blogs dominate mainstream media over a significant amounts of time (e.g., Gaddafi in Figure 17(c)). Centrality of mainstream media and blogs can be relatively constant (Figures 17(a) and (b)) or more time-varying (Figures 17(c) and (h)). We find that a significant rise in the number of central blogs is often temporally correlated with an increasing social unrest (e.g., the Occupy Wall Street movement in September–November 2011 in Figure 17(f)).

5.7 Accuracy on real data

So far we have used memes to trace the flow of information over the Web and have made several qualitative observations about the structure and dynamics of information pathways in online media. We now proceed and attempt to also quantitatively evaluate our algorithm on real data. In case of real data the ground-truth information diffusion network is impossible to obtain. However, we can use the temporal dynamics of hyperlinks created between news sites as a proxy for real information flow. Thus, by observing the times when sites create hyperlinks, our goal is to infer the "targets" of the links (i.e., infer the hyperlink network from the hyperlink times).

We proceed as follows. First, we discretize the time in days, we generate one network $G^*(t)$ per day t, in which we add an edge (u, v) if a document on a site u linked to a document on a site v within the last day. Then we build a set of hyperlink cascades. A hyperlink cascade c_h starts when a site publishes a piece of information and then other sites use hyperlinks to refer to it. Since all our documents/posts are time-stamped, we can trace the hyperlinks in the reverse direction and obtain information cascades. We extracted almost 0.5 million hyperlink cascades from 3.3 million websites from July 2011 till December 2012. Our aim is to use hyperlink cascades to infer the time-varying network $G^*(t)$. We then evaluate how many edges our algorithm estimates correctly by computing accuracy, precision, and recall for each day.

Figure 18 shows precision, recall, and accuracy over time for a time-varying hyperlink network with 11,461 nodes and 19,915 edges created over time, using 495,655 hyperlink cascades from July 2011 to December 2011. We assume an exponential edge transmission model. We observe weekly periodicity and the overall encouraging performance of around 0.4 to 0.5 for all three performance metrics.

6 Conclusions

We have developed a flexible model of the temporal structure underlying diffusion processes. The model makes minimal assumptions about the physical, biological, or

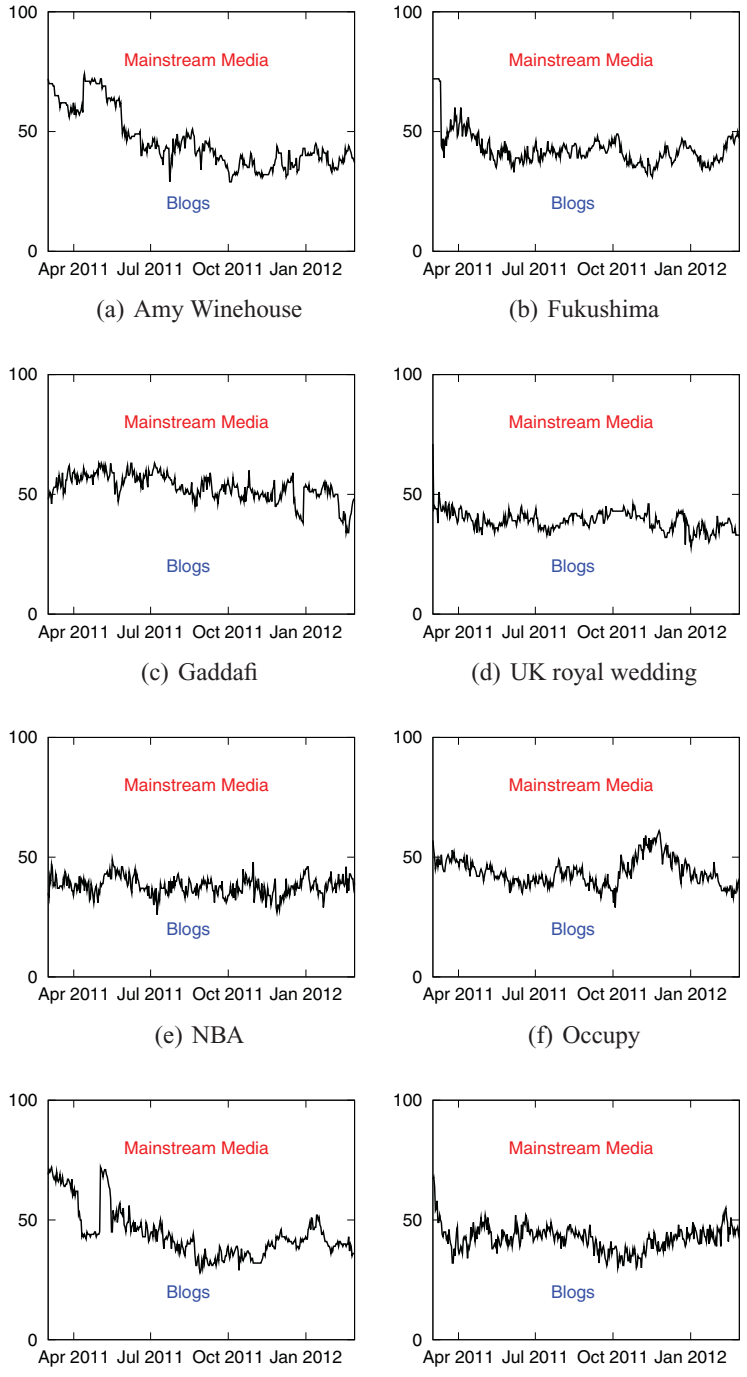
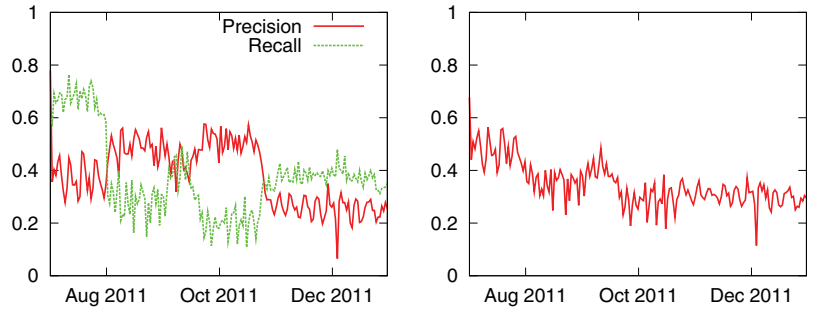


Fig. 17. Percentage of blogs and mainstream media in top-100 most *influential* sites for eight different topics or 2011 world events-inferred diffusion networks. Mainstream media are represented in red, and blogs are represented in blue. (color online)



(a) Precision-Recall (b) Accuracy

Fig. 18. Precision, recall, and accuracy of our stochastic method against time for a time-varying hyperlink network with 11,461 nodes and 19,915 total number of edges across time, using 495,655 hyperlink cascades from July 2011 to December 2011. (color online)

cognitive mechanisms responsible for diffusion. Instead, fitting the model reduces to inferring transmission rates between nodes of a network by finding the rates that maximizes the likelihood of the observed data—temporal traces left by cascades of activations. Qualitative assumptions about activations (e.g., are they long-tailed or faddish?) determine the choice of parametric model on the edges. The model allows mixing exponential, power-law, Rayleigh, or other models, including multi-modal likelihoods (Du et al., 2012) within a single inference algorithm. This provides tremendous flexibility in fitting real data, which may combine long-tailed, faddish, and other qualitative behaviors.

Remarkably, introducing continuous temporal dynamics, allowing variable transmission rates across edges, and avoiding further assumptions dramatically simplified the problem compared with previous approaches (Gomez-Rodriguez et al., 2010; Myers & Leskovec, 2010). The model's parameters have natural interpretations, and it leads to a well-defined, convex maximum likelihood problem that can be solved efficiently. Importantly, we do not need to hand-tune parameters to control the sparsity of the inferred network (i.e., number of edges to infer or penalty terms). Indeed, heuristic l_1-like penalty terms, such as the ones used in Myers & Leskovec (2010), are unnecessary since the probabilistic model naturally imposes sparse solutions. Importantly, other research problems, such as the influence maximization problem (Gomez-Rodriguez & Schölkopf, 2012a), also get simplified under our continuous time model of diffusion.

We evaluated NetRate on a wide range of synthetic diffusion networks—both static and dynamic—with heterogeneous temporal dynamics which aim to mimic the structure of real-world social and information networks. NetRate provides a unique solution to the network inference problem with high recall, precision, and accuracy. A direct comparison with the current state of the art in synthetic networks is difficult, since these methods include a parameter controlling the sparsity of the inferred network that requires blind tuning. Nevertheless, NetRate is typically better in terms of accuracy than previous methods across the full range of their tunable parameters. In addition, NetRate accurately estimates transmission rates, which other methods cannot estimate at all. The performance of ConNIe appears significantly worse than

reported in Myers & Leskovec (2010); a possible explanation for the disparity is that in our work we consider networks with heterogeneous temporal dynamics. It is surprising how well NetInf performs in comparison with NetRate despite assuming uniform temporal dynamics and priors. In addition, we showed that NetRate is able to track changes in the topology of dynamic networks and provide online accurate estimates of the time-varying transmission rates.

Importantly, we run our algorithm on real data and study how real networks and information pathways evolve over time. We found that information pathways over which general recurrent topics propagate remain relatively stable across time. In contrast, unexpected events lead to dramatic changes on the information pathways. We observed that clusters of mainstream news and blogs often emerge and vanish in matter of days. We discovered that there is an early greater increase in information transfer among blogs than among mainstream news involving an increasing dramatic civil unrest, such as the Libyan civil war, Egypt's revolution, or the Syrian uprising. Finally, although we found that the amount of mainstream media and blogs among the most influential nodes for most topics or news events are comparable, the number of influential blogs on some topics or news events grows when there exists an increasing social unrest (e.g., the Occupy Wall Street movement in September–November 2011).

Our model provides a novel view of diffusion processes to build upon, and NetRate provides a computational lens that can dynamically infer the hidden underlying structure of diffusion networks on the basis of observed cascade data. Our work also opens various venues for future work. For example, rigorous theoretical analysis of the convergence of our stochastic gradient descent method would provide further insights into its performance. Moreover, we note that many times the changes in the inferred network structure could be attributed to sudden external real-world events. This opens two interesting questions. How can diffusion network inference be combined with methods for detecting external influence in networks (Myers et al., 2012)? Also, how can dynamic network inference be extended for detecting unexpected real-world events based on a stream of documents? In many real-world scenarios, we do not observe all nodes that become activated during the observation window (Sadikov et al., 2011); in other words, there are missing data. It would be interesting to extend network inference to account for missing data. Last, many times not only information but also sentiment attached to a piece of information spreads through the network (Miller et al., 2011). It would be interesting to think about inference of signed networks, where a positive/negative valence of an edge models sentiment relationship between a pair of nodes. Overall, such methods would allow us to improve our understanding of the current landscape of news coverage, the role that news media plays in framing the discussion of important topics, and the evolving ecosystem that news media occupies.

Acknowledgments

We thank Spinn3r for providing us with data. This research has been supported in part by NSF IIS-1016909, CNS-1010921, CAREER IIS-1149837, IIS-1159679, ARO MURI, DARPA SMISC, DARPA GRAPHS, Okawa Foundation, Docomo, Boeing,

Allyes, Volkswagen,Intel, Alfred P. Sloan Fellowship, Microsoft Faculty Fellowship, Barrie de la Maza Graduate Fellowship, and Max Planck Society.

References

Aalen, O. O., Borgan, Ø., & Gjessing, H. K. (2008). *Survival and event history analysis: A process point of view*. New York, NY: Springer-Verlag.

Adar, E., & Adamic, L. A. (2005). Tracking information epidemics in blogspace. In *Proceedings of The 2005 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 207–214). Washington, DC: IEEE.

Agarwal, A., & Duchi, J. C. (2011). Distributed delayed stochastic optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems 24 (NIPS-24)* (pp. 451–459). NIPS Foundation.

Aral, S., & Walker, D. (2012). Identifying influential and susceptible members of social networks. *Science*, **337**(6092), 337–341.

Bach, F., & Moulines, E. (2011). Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems 24 (NIPS-24)* (pp. 451–459). NIPS Foundation.

Barabási, A.-L., & Albert, R. (1999). Emergence of scaling in random networks. *Science*, **286**, 509–512.

Blatt, D., Hero, A. O., & Gauchman, H. (2008). A convergent incremental gradient method with a constant step size. *SIAM Journal on Optimization*, **18**(1), 29–51.

Boyd, S. P., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge, UK: Cambridge University Press.

Brockmann, D., Hufnagel, L., & Geisel, T. (2006). The scaling laws of human travel. *Nature*, **439**(7075), 462–465.

Chierichetti, F., Kleinberg, J., & Liben-Nowell, D. (2011). Reconstructing patterns of information diffusion from incomplete observations. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems 24 (NIPS-24)* (pp. 792–800). NIPS Foundation.

Clauset, A., Moore, C., & Newman, M. E. J. (2008). Hierarchical structure and the prediction of missing links in networks. *Nature*, **453**(7191), 98–101.

Du, N., Song, L., Gomez-Rodriguez, M., & Zha, H. (2013). Scalable influence estimation in continuous-time diffusion networks. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25 (NIPS-25)* (pp. 2789–2797). NIPS Foundation.

Du, N., Song, L., Smola, A., & Yuan, M. (2012). Learning networks of heterogeneous influence. In P. Barlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25 (NIPS-25)* (pp. 2789–2797). NIPS Foundation.

Duchi, J. C., Agarwal, A., Johansson, M., & Jordan, M. I. (2011). Ergodic subgradient descent. *Proceedings of the 40th Annual Allerton Conference on Communication, Control, and Computing* (pp. 701–706). IEEE.

Erdős, P., & Rényi, A. (1960). On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Science Series B*, **5**, 17–67.

Gomez-Rodriguez, M., Leskovec, J., & Krause, A. (2010). Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining* (pp. 1019–1028). ACM.

Gomez-Rodriguez, M., Leskovec, J., & Krause, A. (2012). Inferring networks of diffusion and influence. *ACM Transactions on Knowledge Discovery from Data*, **5**(4), 21:1–21:37, ACM.

Gomez-Rodriguez, M., & Schölkopf, B. (2012a). Influence maximization in continuous time diffusion networks. In John Langford & Joelle Pineau (Eds.), *Proceedings of the 29th International Conference on Machine Learning* (pp. 313–320). Omnipress.

Gomez-Rodriguez, M., & Schölkopf, B. (2012b). Modeling information propagation with survival theory. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25 (NIPS-25): Workshop in Algorithmic and Statistical Approaches for Large Social Networks*. NIPS Foundation.

Gomez-Rodriguez, M., & Schölkopf, B. (2012c). Submodular inference of diffusion networks from multiple trees. In John Langford & Joelle Pineau (Eds.), *Proceedings of the 29th International Conference on Machine Learning* (pp. 489–496). Omnipress.

Grant, M., & Boyd, S. (2010). CVX: Matlab software for disciplined convex programming, version 1.21. Retrieved from http://cvxr.com/cvx (2013).

Hufnagel, L., Brockmann, D., & Geisel, T. (2004). Forecast and control of epidemics in a globalized world. *Proceedings of the National Academy of Sciences of the United States of America*, **101**(42), 15124.

InfoPath. (2013). InfoPath. Retrieved from http://snap.stanford.edu/infopath/ (2013).

Kaplan, E. H. (1989). What are the risks of risky sex? Modeling the AIDS epidemic. *Operations Research*, **37**(2), 198–209.

Kempe, D., Kleinberg, J. M., & Tardos, É. (2003). Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 137–146). ACM.

Lappas, T., Terzi, E., Gunopulos, D., & Mannila, H. (2010). Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1059–1068). ACM.

Lawless, J. F. (1982). *Statistical models and methods for lifetime data*. New York, NY: Wiley.

Leskovec, J., Adamic, L. A., & Huberman, B. A. (2006). The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce* (pp. 228–237). ACM.

Leskovec, J., Backstrom, L., & Kleinberg, J. (2009). Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining* (pp. 497–506). New York, NY: ACM.

Leskovec, J., Chakrabarti, D., Kleinberg, J., Faloutsos, C., & Ghahramani, Z. (2010). Kronecker graphs: An approach to modeling networks. *Journal of Machine Learning Research*, **11**, 985–1042.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., & Glance, N. (2007b). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining* (pp. 420–429).

Leskovec, J., Lang, K. J., Dasgupta, A., & Mahoney, M. W. (2008). Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th International Conference on World Wide Web* (pp. 695–704). ACM.

Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., & Hurst, M. (2007a). Cascading behavior in large blog graphs. In *Proceedings of the SIAM Conference on Data Mining* (pp. 551–556). SIAM.

Liben-Nowell, D., & Kleinberg, J. (2008). Tracing the flow of information on a global scale using Internet chain-letter data. *Proceedings of the National Academy of Sciences*, **105**(12), 4633–4638.

Lipsitch, M., Cohen, T., Cooper, B., Robins, J. M., Ma, S., James, L., . . . Murray, M. (2003). Transmission dynamics and control of severe acute respiratory syndrome. *Science*, **300**(5627), 1966.

Merriam-Webster's collegiate dictionary. 2004. Springfield, MA: Merriam-Webster.

Miller, M., Sathi, C., Wiesenthal, D., Leskovec, J., & Potts, C. (2011). Sentiment flow through hyperlink networks. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*. AAAI.

Myers, S., & Leskovec, J. (2010). On the convexity of latent social network inference. In J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, & A. Culotta (Eds.), *Advances in Neural Information Processing Systems 23 (NIPS-23)* (pp. 1741–1749). NIPS Foundation.

Myers, S., & Leskovec, J. (2012). Clash of the contagions: Cooperation and competition in information diffusion. *Proceedings of the IEEE International Conference on Data Mining* (pp. 539–548). IEEE.

Myers, S., Leskovec, J., & Zhu, C. (2012). Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 33–41). ACM.

Nemirovski, A., Juditsky, A., Lan, G., & Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, **19**(4), 1574.

Netrapalli, P., & Sanghavi, S. (2012). Finding the graph of epidemic cascades. In *Proceedings of the 12th ACM SIGMETRICS/Performance. PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems* (pp. 211–222). ACM.

NetRate. (2011). NetRate. Retrieved from http://people.tue.mpg.de/manuelgr/netrate/ (2013).

Newey, W. K., & McFadden, D. L. (1994). Large sample estimation and hypothesis testing. In R. F. Engle & D. L. McFadden (Eds.), *Handbook of econometrics*, Vol. IV (pp. 2111–2245). Amsterdam, Netherlands: Elsevier Science B.V.

Prakash, B. A., Beutel, A., Rosenfeld, R., & Faloutsos, C. (2012). Winner takes all: Competing viruses or ideas on fair-play networks. In *Proceedings of the 21st International Conference on World Wide Web* (pp. 1037–1046). ACM.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The Annals of Mathematical Statistics*, **22**(3), 400–407.

Rogers, E. M. (1995). *Diffusion of innovations* (4th ed.). New York, NY: Free Press.

Romero, D. M., Meeder, B., & Kleinberg, J. (2011). Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter. In *Proceedings of the 20th International Conference on World Wide Web* (pp. 695–704). ACM.

Roux, N. L., Schmidt, M., & Bach, F. (2012). A stochastic gradient method with an exponential convergence rate for strongly-convex optimization with finite training sets. In P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25 (NIPS-25)* (pp. 2672–2680). NIPS Foundation.

Sadikov, S., Medina, M., Leskovec, J., & Garcia-Molina, H. (2011). Correcting for missing data in information cascades. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining* (pp. 55–64). ACM.

SNAP. (2012). SNAP: Stanford network analysis platform. Retrieved from http://snap.stanford.edu (2013).

Snowsill, T. M., Fyson, N., De Bie, T., & Cristianini, N. (2011). Refining causality: Who copied from whom? In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 466–474). ACM.

Vu, D. Q., Asuncion, A. U., Hunter, D. R., & Smyth, P. (2011). Continuous-time regression models for longitudinal networks. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, & K. Weinberger (Eds.), *Advances in Neural Information Processing Systems 24 (NIPS-24)* (pp. 2492–2500). NIPS Foundation.

Wallinga, J., & Teunis, P. (2004). Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of Epidemiology*, **160**(6), 509–516.

Wang, C., Knight, J. C., & Elder, M. C. (2000). On computer viral infection and the effect of immunization. In *Proceedings of the 16th Annual Conference on Computer Security Applications* (pp. 246–256). IEEE Computer Society.

Watts, D. J., & Dodds, P. S. (2007). Influentials, networks, and public opinion formation. *Journal of Consumer Research*, **34**(4), 441–458.