Uncovering the Temporal Dynamics of Diffusion Networks

Manuel Gomez-Rodriguez[1,2] MANUELGR@STANFORD.EDU
David Balduzzi[1] BALDUZZI@TUEBINGEN.MPG.DE
Bernhard Schölkopf[1] BS@TUEBINGEN.MPG.DE
[1]MPI for Intelligent Systems and [2]Stanford University

Abstract

Time plays an essential role in the diffusion of information, influence and disease over networks. In many cases we only observe when a node copies information, makes a decision or becomes infected – but the connectivity, transmission rates between nodes and transmission sources are unknown. Inferring the underlying dynamics is of outstanding interest since it enables forecasting, influencing and retarding infections, broadly construed. To this end, we model diffusion processes as discrete networks of continuous temporal processes occurring at different rates. Given cascade data – observed infection times of nodes – we infer the edges of the global diffusion network and estimate the transmission rates of each edge that best explain the observed data. The optimization problem is convex. The model naturally (without heuristics) imposes sparse solutions and requires no parameter tuning. The problem decouples into a collection of independent smaller problems, thus scaling easily to networks on the order of hundreds of thousands of nodes. Experiments on real and synthetic data show that our algorithm both recovers the edges of diffusion networks and accurately estimates their transmission rates from cascade data.

1. Introduction

Diffusion and propagation processes have received increasing attention in a broad range of domains: information propagation (Adar & Adamic, 2005; Gomez-Rodriguez et al., 2010; Meyers & Leskovec, 2010), social networks (Kempe et al., 2003; Lappas et al., 2010), viral marketing (Watts & Dodds, 2007) and epidemiology (Wallinga & Teunis, 2004).

Observing a diffusion process often reduces to noting when nodes (people, blogs, etc.) reproduce a piece of information, get infected by a virus, or buy a product. Epidemiologists can observe when a person becomes ill but they cannot tell who infected her or how many exposures and how much time was 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 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 of individuals. The mechanism underlying the process is hidden. However, the mechanism is of outstanding interest in all three cases, since understanding diffusion is necessary for stopping infections, predicting meme propagation, or maximizing sales of a product.

This article presents a method for inferring the mechanisms underlying diffusion processes based on observed infections. To achieve this aim, we construct a model incorporating some basic assumptions about the spatiotemporal structures that generate diffusion processes. The assumptions are as follows. First, diffusion processes occur over static (fixed) but unknown networks (directed graphs). Second, infections are binary, *i.e.*, a node is either infected or it is not; we do not model partial infections or the partial propagation of information. Third, infections along edges of the network occur independently of each other. Fourth, an infection can occur at different times: the likelihood of node a infecting node b at time t is modeled via a probability density function depending on a, b and t. Finally, we observe *all* infections occurring in the network during the recorded time window. Our aim is to infer the connectivity of the network and the likelihood of infections across its edges after observing the times at which nodes in the

Appearing in *Proceedings of the 28^{th} International Conference on Machine Learning*, Bellevue, WA, USA, 2011. Copyright 2011 by the author(s)/owner(s).

network become infected.

In more detail, we formulate a generative probabilistic model of diffusion that aims to describe realistically how infections occur over time in a static network. Finding the optimal network and transmission rates maximizing the likelihood of an observed set of infection cascades reduces to solving a convex program. The convex problem decouples into many smaller problems, allowing for natural parallelization so that our algorithm scales to networks with hundreds of thousands of nodes. We show the effectiveness of our method by reconstructing the connectivity and continuous temporal dynamics of synthetic and real networks using cascade data.

Related work. The work most closely related to ours (Gomez-Rodriguez et al., 2010; Meyers & 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 Meyers & Leskovec (2010) (CONNIE) infer not only the connectivity but also a prior probability of infection 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. In contrast, our model allows transmission at different rates across different edges so that we can infer temporally heterogeneous interactions within a network, as found in real-world examples. Thus, we can now infer the temporal dynamics of the underlying network.

The main innovation of this paper is to model diffusion as a spatially discrete network of continuous, conditionally independent temporal processes occurring at different rates. Infection transmission 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). We avoid modeling the mechanisms underlying individual infections, 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 infection times, but not prior probabilities of infection that depend on unknown external factors. To the best of our knowledge, continuous temporal dynamics of diffusion networks has not been modeled or inferred in previous work. We believe this is a key point for understanding diffusion processes.

2. Problem formulation

This paper develops a method for inferring the spatiotemporal dynamics that generate observed infections. In this section we formulate our model, starting from the data it is designed for, and concluding with a precise statement of the network inference problem.

Data. Observations are recorded on a fixed population of N nodes and consist of a set C of cascades $\{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}$. Each cascade $\mathbf{t}^c$ is a record of observed infection times within the population during a time interval of length T^c. A cascade is an N-dimensional vector $\mathbf{t}^c := (t_1^c, \ldots, t_N^c)$ recording when nodes are infected, $t_k^c \in [0, T^c] \cup \{\infty\}$. Symbol ∞ labels nodes that are not infected during observation window $[0, T^c]$ – it does not imply that nodes are never infected. The 'clock' is reset to 0 at the start of each cascade. Lengthening the observation window T^c increases the number of observed infections 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.

The time-stamps assigned to nodes by a cascade induce the structure of a directed acyclic graph (DAG) on the network (which is *not* acyclic in general) by defining node i is a parent of j if $t_i < t_j$. 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. Also, since the underlying network is inferred from many cascades (each of which imposes its own DAG structure), the inferred network is typically not a DAG.

Pairwise transmission likelihood. The first step in modeling diffusion dynamics is to consider pairwise interactions. We assume that infections can occur at different rates over different edges of a network, and aim to infer the transmission rates between pairs of nodes in the network.

Define $f(t_i|t_j, \alpha_{j,i})$ as the conditional likelihood of transmission between a node j and node i. The transmission likelihood depends on the infection times (t_j, t_i) and a pairwise transmission rate $\alpha_{j,i}$. A node cannot be infected by a node infected later in time. In other words, a node j that has been infected at a time t_j may infect a node i at a time t_i only if $t_j < t_i$. Although in some scenarios it may be possible to estimate a non-parametric likelihood empirically, for simplicity we consider three well-known parametric models: exponential, power-law and Rayleigh (see Table 1). Transmission rates are denoted as $\alpha_{j,i} \geq 0$ and δ is the minimum allowed time difference in the power-law to have a bounded likelihood. As $\alpha_{j,i} \to 0$ the likelihood of infection tends to zero and the expected transmission time becomes arbitrarily long. Without loss of generality, we consider $\delta = 1$ in the power-law model from now on.

Exponential and power-laws are monotonic models that

Table 1.

Model	**Transmission likelihood** $f(t_i \mid t_j; \alpha_{j,i})$	**Log survival function** $\log S(t_i \mid t_j; \alpha_{j,i})$	**Hazard function** $H(t_i \mid t_j; \alpha_{j,i})$
Exponential (EXP)	$\begin{cases} \alpha_{j,i} \cdot e^{-\alpha_{j,i}(t_i - t_j)} & \text{if } t_j < t_i \\ 0 & \text{otherwise} \end{cases}$	$-\alpha_{j,i}(t_i - t_j)$	$\alpha_{j,i}$
Power law (POW)	$\begin{cases} \frac{\alpha_{j,i}}{\delta}\left(\frac{t_i - t_j}{\delta}\right)^{-1-\alpha_{j,i}} & \text{if } t_j + \delta < t_i \\ 0 & \text{otherwise} \end{cases}$	$-\alpha_{j,i} \log\left(\frac{t_i - t_j}{\delta}\right)$	$\alpha_{j,i} \cdot \frac{1}{t_i - t_j}$
Rayleigh (RAY)	$\begin{cases} \alpha_{j,i}(t_i - t_j) e^{-\frac{1}{2}\alpha_{j,i}(t_i - t_j)^2} & \text{if } t_j < t_i \\ 0 & \text{otherwise} \end{cases}$	$-\alpha_{j,i} \frac{(t_i - t_j)^2}{2}$	$\alpha_{j,i} \cdot (t_i - t_j)$

have been previously used in modeling diffusion networks and social networks (Gomez-Rodriguez et al., 2010; Meyers & Leskovec, 2010). Power-laws model infections with long-tails. The Rayleigh model is a non-monotonic parametric model previously used in epidemiology (Wallinga & Teunis, 2004). It is well-adapted to modeling fads, where infection likelihood rises to a peak and then drops extremely rapidly.

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

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

The hazard functions of our models are simple, Table 1.

Probability of survival given a cascade. We compute the probability that a node survives uninfected until time T, given that some of its parents are already infected. Consider a cascade $\mathbf{t} := (t_1, \ldots, t_N)$ and a node i *not* infected during the observation window, $t_i > T$. Since each infected node k may infect i independently, the probability that nodes $1 \ldots N$ do *not* infect node i by time T is the product of the survival functions of the infected nodes $1 \ldots N | t_k \leq T$ targeting i,

$$\prod_{t_k \leq T} S(T|t_k; \alpha_{k,i}). \tag{1}$$

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

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

Computing the likelihood of a cascade thus reduces to computing the conditional likelihood of the infection time of each node given the rest of the cascade. As in the independent cascade model (Kempe et al., 2003), we assume that a node gets infected once the *first* parent infects the node. Given an infected node i, we compute the likelihood of a potential parent j to be the first parent by applying Eq. 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 Eq. 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 Eq. 2 the likelihood of the infections in a cascade is

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

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

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

Eq. 6 only considers infected nodes. However, the fact that some nodes are *not* infected during the observation window is also informative. We therefore add the multiplicative survival term from Eq. 1 and also replace the ratios in Eq. 6 with hazard functions:

$$f(\mathbf{t};\mathbf{A}) = \prod_{t_i \leq T} \prod_{t_m > T} S(T|t_i;\alpha_{i,m}) \times \prod_{k:t_k<t_i} S(t_i|t_k;\alpha_{k,i}) \sum_{j:t_j<t_i} H(t_i|t_j;\alpha_{j,i}). \quad (7)$$

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 the individual cascades given by Eq. 7:

$$\prod_{\mathbf{t}^c \in C} f(\mathbf{t}^c;\mathbf{A}). \quad (8)$$

Network Inference Problem. Our goal is to find the transmission rates $\alpha_{j,i}$ of every pair of nodes such that the likelihood of an observed set of cascades $C = \{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}$ is maximized. Thus, we aim to solve:

$$\begin{array}{ll} \text{minimize}_{\mathbf{A}} & -\sum_{c\in C} \log f(\mathbf{t}^c;\mathbf{A}) \\ \text{subject to} & \alpha_{j,i} \geq 0,\, i,j = 1,\ldots,N, i \neq j, \end{array} \quad (9)$$

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

3. Proposed algorithm: NETRATE

The solution to Eq. 9 is 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 network inference problem defined by equation Eq. 9 is convex in* $\mathbf{A}$.

Proof. By Eq. 8, the log-likelihood of a set of cascades is

$$L\big(\{\mathbf{t}^1 \ldots \mathbf{t}^{|C|}\};\mathbf{A}\big) = \sum_c \Psi_1(\mathbf{t}^c;\mathbf{A}) + \Psi_2(\mathbf{t}^c;\mathbf{A}) + \Psi_3(\mathbf{t}^c;\mathbf{A}), \quad (10)$$

where for each cascade $\mathbf{t}^c \in \{\mathbf{t}^1, \ldots, \mathbf{t}^{|C|}\}$,

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

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

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

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 Eq. 9 follows from linearity, composition rules for concavity, and concavity of the logarithm. □

Corollary 2. *The network inference problem defined by equation Eq. 9 is convex for the exponential, power-law and Rayleigh models.*

Theorem 3. *The maximum likelihood estimator* $\hat{\alpha}$ *given by the solution of Eq. 9 is consistent.*

Proof Sketch. We check the criteria for consistency of identification, continuity and compactness (Newey & McFadden, 1994). The log-likelihood in Eq. 10 is a continuous function of $\mathbf{A}$ for any fixed set of cascades $\{\mathbf{t}^1 \ldots \mathbf{t}^{|C|}\}$, 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.

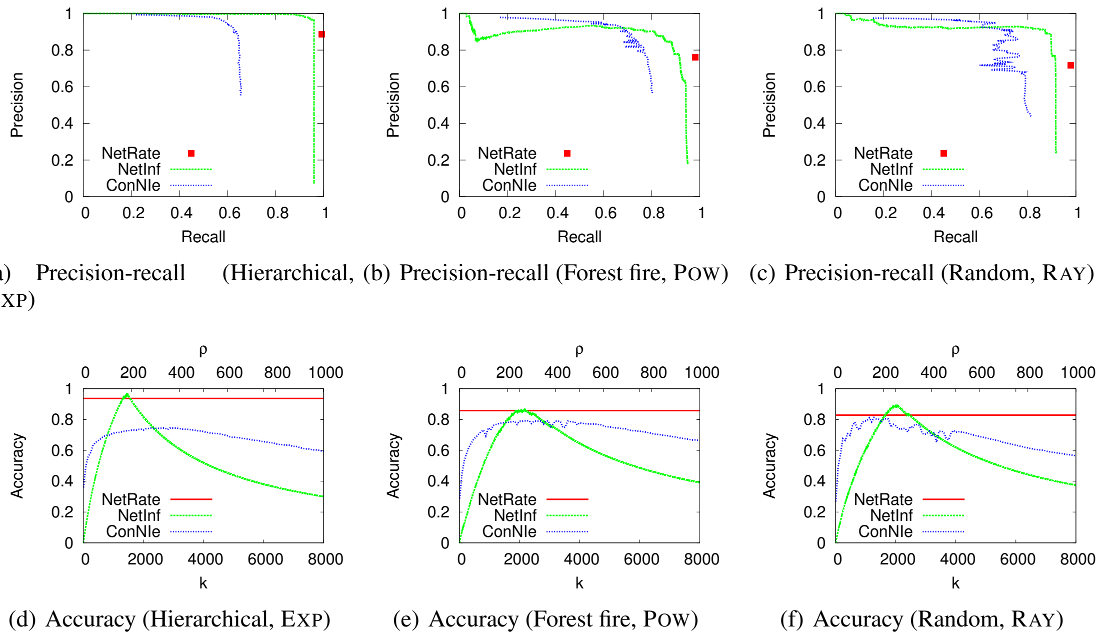
We refer to our network inference method as NETRATE.

Properties of NETRATE. We highlight some common features of the solutions to the network inference problem for the exponential, power-law and Rayleigh models.

All terms in Eq. 10 depend only on transmission rates $\alpha_{j,i}$ and infection time differences $(t_i - t_j)$, but not absolute infection times t_i or t_j. Our formulation thus does not depend on the absolute time of the root node of each cascade.

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 and therefore 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 infection times.

The Ψ_2 term penalizes edges $k \to i$ based on the infection times difference $t_i - t_k$. Edges transmitting infections slowly are heavily penalized and conversely. The Ψ_1 term penalizes edges $i \to j$ targeting *uninfected* nodes j based on the time $T - t_i$ till the observation window cutoff. Lengthening the observation window produces harsher penalties – however, it also allows further infections. The penalties are finite, *i.e.*, if no infection of node j is observed, we can only say that it has survived until time T. There is insufficient evidence to claim j will never be infected. 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 infection to inferring it is impossi-



(a) Precision-recall (Hierarchical, EXP) (b) Precision-recall (Forest fire, POW) (c) Precision-recall (Random, RAY)

(d) Accuracy (Hierarchical, EXP) (e) Accuracy (Forest fire, POW) (f) Accuracy (Random, RAY)

Figure 1. Panels (a-c) plot precision against recall; panels (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,d): 1,024 node hierarchical Kronecker network with exponential model for 5,000 cascades. (b,e): 1,024 node Forest Fire network with power law model for 5,000 cascades. (c,f): 1,024 node random Kronecker network with Rayleigh model for 2,000 cascades.

ble. Instead, NETRATE infers that infections are impossible across certain edges, *i.e.*, that some of the optimal rates $\alpha_{j,i}$ are 0, based solely on the observed data and the length of the time horizon.

The Ψ_3 term ensures infected 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 infected nodes for having many parents.

Optimizing NETRATE. We speed up the convex program by orders of magnitude via two improvements:

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 infection times of other nodes in cascades it belongs to.

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 Eq. 10, so the optimal $\alpha_{j,i}$ is zero. We therefore simply the objective function by setting $\alpha_{j,i}$ to zero.

Solving NETRATE. We solve Eq. 9 with CVX, a package for specifying and solving convex programs (Grant & Boyd, 2010).

4. Experimental evaluation

We evaluate the performance of NETRATE on: (i) synthetic networks that mimic the structure of social networks and (ii) real cascades extracted from the MemeTracker dataset[1]. We show that NETRATE discovers more than 95% of the edges in synthetic networks and more than 60% in real networks, accurately recovers transmission rates from diffusion data, and typically outperforms two previously developed inference algorithms, NETINF and CONNIE. We use the public implementations of NETINF and CONNIE.

4.1. Experiments on synthetic data

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 dif-

[1] Data available at `http://memetracker.org`

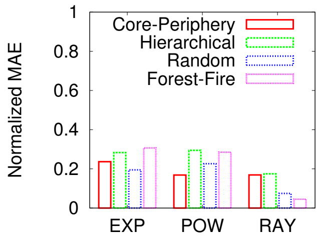


Figure 2. NetRate's normalized mean absolute error (MAE) for three types of Kronecker networks (1,024 nodes and 2,048 edges) and a Forest Fire network (1,024 edges and 2,422 edges) for 5,000 cascades. We consider all three models of transmission: exponential (Exp), power-law (Pow) and Rayleigh (Ray).

fusion networks. We generate three types of Kronecker graph 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 generate network G^* by drawing transmission rates for edges (j, i) from a uniform distribution. For the exponential and Rayleigh models $\alpha \in [0.01, 1]$ and for the power law $\alpha \in [0.01, 2]$. The transmission rate for an edge (j, i) models how fast the information spreads from node j to node i in social networks. Then, we generate a set of cascades over G^*. Root nodes of cascades are chosen uniformly at random. As noted previously, the optimization problem depends on the time differences $(t_i - t_j)$. Therefore, our formulation does not depend on the absolute time of the root node of each cascade. Once a node is infected, the transmission likelihoods of outgoing edges determine the infection times of its neighbours. We record the time of the first infection if a node is infected more than once. Infections are not observed after a pre-specified time horizon T.

Accuracy of NetRate. We evaluate NetRate against two other inference methods, NetInf and ConNIe, 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 normalized mean absolute error (MAE, *i.e.*, $E\big[|\alpha^* - \hat{\alpha}|/\alpha^*\big]$, where α^* is the true transmission rate and $\hat{\alpha}$ is the estimated transmission rate). Note that in NetRate, as for real cascades, the probability of infection depends on both the transmission rate and the observation window. In contrast, ConNIe assigns probability priors to edges that are defined without reference to an observation window. Therefore, the values assigned to edges by NetRate and ConNIe are not comparable, so we do not compute MAE for ConNIe.

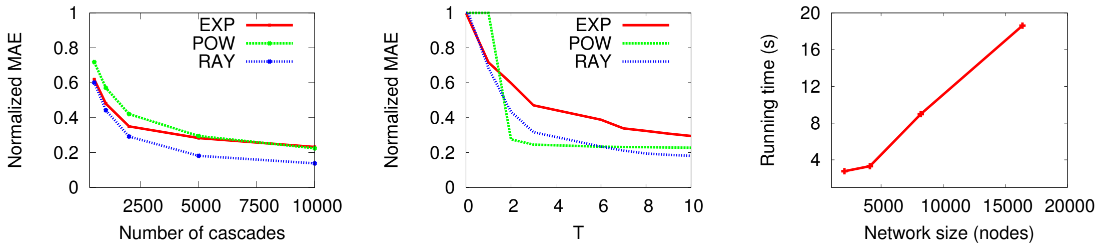
Figure 1 compares the precision, recall and accuracy of NetRate with NetInf and ConNIe for two types of Kronecker networks (hierarchical community structure and random) and a Forest Fire network over an observation window of length $T = 10$. In terms of precision-recall, NetRate outperforms ConNIe and NetInf for all the synthetic examples in the Pareto sense (Boyd & Vandenberghe, 2004). More specifically, if we set ConNIe 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 the 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 have to select a point blindly (or at best heuristically), whereas NetRate yields a unique solution without any tuning.

Figure 2 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.

NetRate performance vs. cascade coverage. Observing more cascades leads to higher precision-recall and more accurate estimates of the transmission rates. Figure 3(a) plots the MAE of inferred networks against the number of observed cascades for a hierarchical Kronecker network with all three transmission models. 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%.

NetRate performance vs. time horizon. Intuitively, the longer the observation window, the more accurately NetRate is able to infer transmission rates. Figure 3(b) confirms this intuition by showing the MAE of inferred networks for different time horizons T for a hierarchical Kronecker with exponential, power-law and Rayleigh transmission models for 5,000 cascades.

NetRate running time. Figure 3(c) plots the average running time to infer rates of all incoming edges to a node against number of nodes in a network (the number of edges is twice the number of nodes) on a single CPU. Further



(a) Cascade coverage (b) Time horizon (c) Running time

Figure 3. Panels (a,b) show NETRATE's normalized MAE vs number of cascades and time horizon respectively for a hierarchical Kronecker network with 1,024 nodes and 2,048 edges with exponential (EXP), power-law (POW) and Rayleigh (RAY) transmission models. Panel (c) plots NETRATE's average running time to infer rates of all incoming edges to a node against network size (number of nodes) for a hierarchical Kronecker network.

improvements can be achieved since NETRATE naturally splits into a collection of subproblems, one per node. A cluster with 25 CPUs can therefore infer a network with 16,000 nodes (and 32,000 edges) in less than 4 hours.

4.2. Experiments on real data

Dataset description. As in previous work tackling diffusion networks, we use the MemeTracker dataset, which contains more than 172 million news articles and blog posts from 1 million online sources. We use hyperlinks between blog posts to trace the flow of information. A site publishes a piece of information and uses hyper-links to refer to the same or closely related pieces of information published by other sites. These other sites link to still others and so on. A cascade is thus a collection of time-stamped hyperlinks between sites (in blog posts) that refer to the same or closely related pieces of information. We record one cascade per piece – or closely related pieces – of information. We extract the top 500 media sites and blogs with the largest number of documents, 5,000 hyperlinks and 116,234 cascades.

Accuracy on real data. As the ground truth is unknown on real data, we proceed as follows. We create a network where there is an edge (u, v) if a post on a site u linked to a post on a site v. We consider this as the ground truth network G^* and we use the hyperlink cascades to infer the network $\hat{G}$ and evaluate how many edges our method estimates properly. We assume an exponential model.

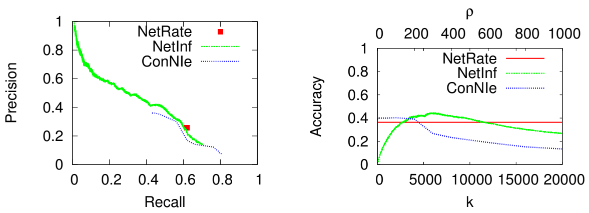
Figure 4 compares our results with NETINF and CONNIE. As in the synthetic experiments, NETRATE yields a unique solution whereas the other algorithms produce curves of solutions. Panel (a) shows that NETRATE performs comparably to NETINF and it outperforms CONNIE on precision and recall. Panel (b) plots accuracy: NETRATE's outperforms the other two algorithms on the majority of their outputted solutions, and almost matches their best performances. Since there are no principled methods for choosing single solutions for NETINF and CONNIE, there is no guarantee that the solutions chosen from the curves will be anywhere near the highest achievable value.

5. Conclusions

We have developed a flexible model, NETRATE, of the spatiotemporal structure underlying diffusion processes. The model makes minimal assumptions about the physical, biological or cognitive mechanisms responsible for diffusion. Instead, it infers transmission rates between nodes of a network by computing the model that maximizes the likelihood of the observed data – temporal traces left by cascades of infections. Qualitative assumptions about infections (*e.g.*, are they long-tailed?) determine the choice of parametric model on the edges. An interesting feature of NETRATE, to be investigated in future work, is the possibility of mixing exponential, power law, Rayleigh or other models within a single inference algorithm, thus providing 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; Meyers & Leskovec, 2010). The model has parameters with natural interpretations, and it leads to a well-defined convex maximum likelihood problem that can be solved efficiently. Importantly, we do not need to tune parameters by hand to control the sparsity of the inferred network (*i.e.*, number of edges to infer or penalty terms). Heuristic l_1-like penalty terms, as the ones used in Meyers & Leskovec (2010), are unnecessary since the probabilistic model naturally imposes sparsity.

We evaluated NETRATE on a wide range of synthetic diffusion networks with heterogeneous temporal dynamics which aim to mimic the structure of real-world social and



(a) Precision-recall (b) Accuracy

Figure 4. Real data. Precision-recall and accuracy of NetRate, NetInf and ConNIe, with an exponential model, on a 500 node hyperlink network with 5,000 edges using hyperlinks cascades.

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 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, it accurately estimates transmission rates, which other methods cannot estimate at all. The performance of ConNIe appears significantly worse than reported in Meyers & Leskovec (2010); a possible explanation for the degradation 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.

Finally, we evaluated NetRate on real data. Again, NetRate provides a unique solution to the network inference problem but in this case, as expected, the values of recall, precision and accuracy are modest – adopting a simple parametric pairwise transmission model is a simplistic assumption on real data. In terms of accuracy, it outperforms previous methods across a significant part of the full range of their tunable parameters.

NetRate provides a novel view of diffusion processes. We believe it can be fruitfully applied to several lines of research including influence maximization, control of epidemics, and causal inference.

References

Adar, E. and Adamic, L. A. Tracking information epidemics in blogspace. In *Web Intelligence*, pp. 207–214, 2005.

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

Boyd, S.P. and Vandenberghe, L. *Convex optimization*. Cambridge University Press, 2004.

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

Erdős, P. and Rényi, A. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Science*, 5:17–67, 1960.

Gomez-Rodriguez, M., Leskovec, J., and Krause, A. Inferring Networks of Diffusion and Influence. In *Proc. of the 16th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 1019–1028, 2010.

Grant, M. and Boyd, S. CVX: Matlab software for disciplined convex programming, version 1.21. `http://cvxr.com/cvx`, 2010.

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

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

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

Leskovec, J., Lang, K. J., Dasgupta, A., and Mahoney, M. W. Statistical properties of community structure in large social and information networks. In *Proc. of the 17th International Conference on World Wide Web*, 2008.

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

Meyers, S. and Leskovec, J. On the Convexity of Latent Social Network Inference. In *Advances in Neural Information Processing Systems*, 2010.

Newey, W. K. and McFadden, D. L. Large sample estimation and hypothesis testing. In *Handbook of Econometrics, Volume IV*, pp. 2111–2245. 1994.

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

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