A Bayesian Framework for Estimating Properties of Network Diffusions

Varun R Embar*
IBM Research Division
IBM India Research Lab
Bangalore 560045, India
varembar@in.ibm.com

Rama Kumar Pasumarthi*
IBM Research Division
IBM India Research Lab
Bangalore 560045, India
sarpasum@in.ibm.com

Indrajit Bhattacharya
IBM Research Division
IBM India Research Lab
Bangalore 560045, India
indrajitb@in.ibm.com

ABSTRACT

The analysis of network connections, diffusion processes and cascades requires evaluating properties of the diffusion network. Properties of interest often involve variables that are not explicitly observed in real world diffusions. Connection strengths in the network and diffusion paths of infections over the network are examples of such hidden variables. These hidden variables therefore need to be estimated for these properties to be evaluated. In this paper, we propose and study this novel problem in a Bayesian framework by capturing the posterior distribution of these hidden variables given the observed cascades, and computing the expectation of these properties under this posterior distribution. We identify and characterize interesting network diffusion properties whose expectations can be computed exactly and efficiently, either wholly or in part. For properties that are not 'nice' in this sense, we propose a Gibbs Sampling framework for Monte Carlo integration. In detailed experiments using various network diffusion properties over multiple synthetic and real datasets, we demonstrate that the proposed approach is significantly more accurate than a frequentist plug-in baseline. We also propose a map-reduce implementation of our framework and demonstrate that this can analyze cascades with millions of infections in minutes.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Application—*Data mining*

Keywords

Social Influence Analysis; Information Cascades; Networks of Diffusion; Bayesian Analysis; Gibbs Sampling

1. INTRODUCTION

The study of networks and diffusions over them has a long history in epidemiology, sociology, econometrics and marketing. Interest in the problem has increased many fold over the last two decades in the context of information diffusion and social networks, first because of the growth of the internet, and then the social media revolution [2, 13]. The study typically involves three different objects of interest: a *network* that defines connection strengths between entities, a *diffusion process* that defines how 'infections' spread randomly over the network, and *cascades* that trace the spread of specific infections over the network. Many different problems have been studied in the context of these three objects of interest. A problem that has received a lot of attention is that of network inference [21, 7, 6, 8, 4, 9, 19, 22, 15], where the task is to infer the hidden network of connection strengths from the cascades.

However, inferring the network of diffusions is often an intermediate task in the analysis. The main objective is often to compute some *property* of the network and/or the cascades, such as centrality and reach of individual nodes, optimal seeds for viral marketing [14, 12, 10], community structures [17, 1], the likelier diffusion mechanism [18], etc.

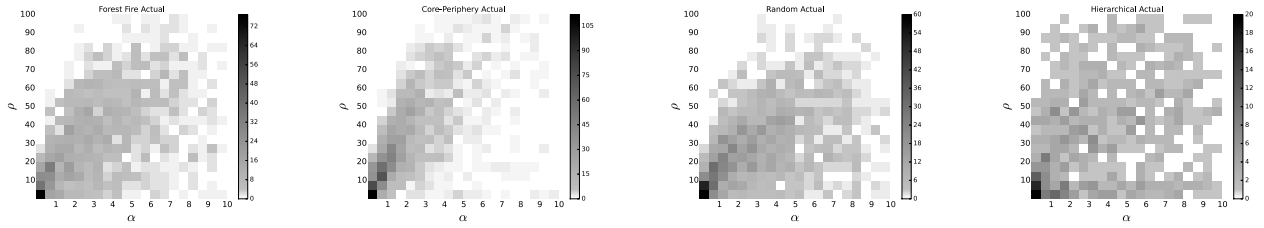
An example of such a property is the identity of 'tribe leaders' [10], who are well connected to a large tribe of nodes in the network, and whose tribe members follow their actions frequently in the cascades. Computing this notion of node influence is expensive even when the cascades are completely observed, but consider a simplification that is still interesting. Consider the out-degree of a node in the network, counting only those edges that are strong and also frequently used in the cascades. This influence score is much simpler to compute given completely observed networks and cascades, and yet is useful for marketers and epidemiologists.

In Fig. 1, we show the strength-frequency distribution of edges in four different synthetically-generated network diffusions. The x-axis shows edge strength (α) and the y-axis transmission frequency (ρ), which is the number of times an edge was used to transmit infections in the cascades. The distribution only considers actual edges used in the cascades. Fig. 1(a) through Fig. 1(d) correspond to Forest Fire, Core-Periphery, Random and Hierarchical graphs respectively, each with 1024 nodes and ~ 2000 edges. In each case, we generated 20 splitting, independent cascades [22] over these graphs with 2 randomly chosen seeds for each cascade. The plots can also be interpreted as the distribution of the simpler node influence score discussed above. The plots clearly show that these distributions look very different depending on the underlying network connections and possibly also the diffusion mechanism. Thus, given network diffusion data from some network with unknown structure

*The first two authors have contributed equally to this work

KDD'14, August 24–27, 2014, New York, NY, USA.

http://dx.doi.org/10.1145/2623330.2623693 .

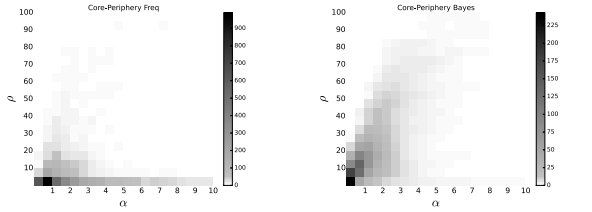


(a) Forest fire (FF) (b) Core-Periphery (CP) (c) Random (Rnd) (d) Hierarchical (HI)

Figure 1: Actual strength-frequency distribution for cascades from synthetic graphs

and diffusion mechanism, it is clearly of interest to construct and study such distributions.

In this paper, we investigate such joint properties of networks and cascades. The main difficulty in evaluating such properties for real-world network diffusions is that the connections strengths in the network are unknown. Additionally, the diffusion edges are also unknown — the cascades only record the catchers of the infections and the infection times, but not the actual path traced by specific infections. For example, in social information flows, the friends and followers are known, but not the extent of influence between them, and often users do not reveal the sources of information. Therefore, to evaluate the properties, these hidden variables need to be inferred from the observed cascades.



(a) Frequentist (b) Bayesian Expectation

Figure 2: Reconstructed distribution: CP

One way to evaluate a property is to take the 'frequentist plug-in approach', that finds point estimates of the network and the diffusion edges in the cascades, and then evaluates the property using these point estimates. The most popular point estimate used for network inference is the maximum likelihood estimate [6, 22]. This approach for evaluating properties has two drawbacks. The first is the well known problem of overfitting for a frequentist approach. More importantly, for properties that are not one-to-one functions of the network and the diffusion paths, the most likely value of the property need not correspond to the most likely network and diffusion paths. Fig. 2(a) shows the reconstructed edge distribution for the Core-Periphery diffusion data using this frequentist approach. It has failed to recover the signature shape of the distribution.

In this paper, we motivate and propose a Bayesian solution to this problem, where both the network and the diffusion paths are modeled as random variables. This makes network diffusion properties functions of random variables, and our problem becomes one of computing the expectation of the property under the posterior distribution of the hidden variables given the observed features of the cascade.

An obvious challenge for the Bayesian approach is the cost of computing expectations. This seems daunting for the network inference problem with its large number of coupled discrete and continuous hidden variables. However, our analysis shows that for the popular independent cascade model, many interesting network diffusion properties are 'nice', in that their expectations can be computed exactly and efficiently, at least in part. For parts of the expectations that are not 'nice', we propose a Gibbs Sampling technique for efficient Monte Carlo integration. Fig. 2(b) shows the reconstruction of the Core-Periphery strength-frequency distribution using our proposed approach. It has recovered the distinctive shape to a much better extent.

In detailed experiments using various network diffusion properties over multiple synthetic and real datasets, we demonstrate that the proposed approach is significantly more accurate than the MLE plug-in baseline, and is useful for traditional network inference as well. We show that the approach scales easily to very large datasets using a map-reduce implementation.

2. RELATED WORK

Different problems have been studied in the context of diffusion networks [2, 13]. The network inference problem [21, 7, 6, 8, 4, 9, 19, 22, 15] has been investigated in depth, starting with stationary discrete time models [19], to the more recent models that consider features [22] and time-varying networks [9]. The solutions have mostly been based on maximum-likelihood estimation.

Apart from inferring the complete network structure, there has been work on inferring summaries of the network, such as community structures [17, 1]. Other investigated properties are estimating influence of nodes [3], and subsequently selecting a subset of nodes that maximize influence [12]. Sadikov et. al. [20] study various properties of cascades assuming completely missing infections.

Milling et. al. [18] study the problem of deciding which of two given networks caused a specific diffusion with its path properties observed. Efficient algorithms have been designed for identifying leaders and tribes [10] and mining propagation summaries [16] from cascades, assuming the underlying network and the diffusion paths to be known. These problems may be seen as computing joint properties of networks and cascades, with all variables observed.

In summary, we are not aware of any general framework for estimating joint properties of networks and diffusion processes in the context of hidden network and diffusions paths. We are also not aware of any Bayesian framework for network diffusion analysis.

3. PROBLEM DEFINITION

In this section, we first review the network diffusion problem and the independent cascade model. Then we define network diffusion properties and their expectations.

Network Diffusion and Independent Cascade Model: We assume a *network* $G = (V, E)$ with nodes V and edges E. For $(u, v) \in E$, let $\alpha_{uv} \in \mathcal{R}_+$ denote the connection strength between nodes u and v. We have a set C of *cascades* corresponding to spreading infections over the network G. Each cascade $c \in C$ consists of a set of time-stamped infections: $c = \{(u_i, z_i, t_i)\}$. The i^{th} infection records that node u_i got infected at time t_i by its parent infection z_i. We will assume that $t_i < t_j$ for $i < j$. Let $\pi_i \subseteq \{1 \dots i-1\}$ denote the set of 'potential parents' for the i^{th} infection, so that $z_i \in \pi_i$. Observe that the infecting parent z_i provides edge information for reconstructing the infection paths over the network. We also know that the cascades were observed for time T, which is at least as large as the final infection time.

The joint distribution $p(C|\alpha)$ on the cascades C given the network strengths is typically defined using a *generative process* that captures the dynamics of spreading infections. While many diffusion models have been proposed, we follow the popular *Continuous-Time Independent Cascade Model* [6]. Under this model, each cascade starts with an initial set of seed nodes getting infected. Each infected node proposes an infection times for each currently uninfected neighbor in the network. An uninfected node catches its infection from that infected neighbor who has proposed the earliest infection time for it. We consider the setting where nodes can get infected multiple times in the same cascade, and the *splitting model* for this [22], where all infections between the current and the previous infections of a node are considered as its potential parents.

The main building block of the model is the probability density function $f(t_i|u_i, u_j, t_j; \alpha_{u_j u_i})$, which models the conditional likelihood of node u_i getting infected at time t_i by node u_j which got infected at time t_j for $t_j < t_i$. [6, 22]:

$$p(c|\alpha) = \prod_i H(t_i|t_{z_i}; \alpha_{u_{z_i} u_i}) \prod_{j \in \pi_i} S(t_i|t_j; \alpha_{u_{z_j} u_i}) \prod_{v: l_v < t_i} S(T|t_i; \alpha_{u_i v})$$

where $S(t) = 1 - F(t)$ is the survival function, $H(t) = f(t)/S(t)$ is the hazard function corresponding to the CDF $F(t) = \int_0^t f(t)dt$, l_v is the time of last infection of node v. Assuming cascades to be generated *iid*, the likelihood of C becomes $p(C|\alpha) = \prod_{c \in C} p(c|\alpha)$.

In real-world network diffusions, many of the variables above are unobserved. We will assume that the observed trace $C^o = \{c^o\}$ of the cascade C only contains the infected node u_i and the infection time t_i: $c^o = \{(u_i, t_i)\}$. The infecting parent z_i is not observed. The posterior distribution $p(z|\{c^o\}, \alpha)$ over infection parents, conditioned on observed cascades $\{c^o\}$ and α, has the following form:

$$p(z \mid \{c^o\}, \alpha) = \prod_i \frac{H(t_i|t_{z_i}; \alpha_{u_{z_i} u_i})}{\sum_{j \in \pi_i} H(t_i|t_j; \alpha_{u_j u_i})} \quad (1)$$

Observe that this decouples into terms involving individual infection parents z_i. This will be a key property for efficient computation of network diffusion properties in Sec. 5.

The network connection strengths α_{uv} are also typically unobserved. Further, we assume that the set of network edges E is also not known. Therefore, we consider α to be a $|V| \times |V|$ matrix of unknown variables. The goal of the popular network inference problem is to reconstruct this α matrix using $\{c^o\}$ [6, 22]. The state-of-the-art approach is to obtain a maximum likelihood estimate:

$$\hat{\alpha} = \arg\max_\alpha \log p(\{c^o\}|\alpha) = \arg\max_\alpha \log \sum_z p(C|\alpha) \quad (2)$$

For modeling $f(t_i|u_i, u_j, t_j; \alpha)$, the Exponential, Powerlaw and Rayleigh distributions have been proposed [6, 22]. For the Exponential distribution,

$$f(t_i|t_j; \alpha) = \alpha e^{-\alpha(t_i - t_j)}$$
$$H(t_i|t_j) = \alpha; \quad S(t_i|t_j) = e^{-\alpha(t_i - t_j)}$$

and for the Rayleigh distribution,

$$f(t_i|t_j; \alpha) = \alpha(t_i - t_j) e^{-\frac{1}{2}\alpha(t_i - t_j)^2}$$
$$H(t_i|t_j) = \alpha(t_i - t_j); \quad S(t_i|t_j) = e^{-\frac{1}{2}\alpha(t_i - t_j)^2}$$

Network Diffusion Properties and Expectations: Given this background, we now define our problem. We are interested in computing properties $f(\alpha, C)$ of the cascades C and the network connections α. Two examples properties are the strength-frequency distribution of edges, and influence of leader nodes. We will see more examples in Sec. 5.

Since α and z are unobserved, the properties are not directly computable. We investigate a fully Bayesian solution to the problem, where we model both α and z to be random variables, so that the property $f(\alpha, z)$ is a function of random variables. Assuming a joint distribution $p(C, \alpha)$ to be defined on the cascade C and the network connections strengths α, we consider the posterior distribution $p(z, \alpha|\{c^o\})$ over the hidden variables z and α conditioned on the observed trace $c^o = \{(u_i, t_i)\}$ of the cascades. Then we evaluate the expectation $\bar{f}(C, \alpha)$ of $f(C, \alpha)$ under this posterior distribution:

$$\bar{f}(C, \alpha) = E_{p(z,\alpha|\{c^o\})}[f(C, \alpha)] \quad (3)$$

For properties that do not involve z, we consider the expectation under the marginal posterior distribution $p(\alpha|\{c^o\}) = \sum_z p(z, \alpha|\{c^o\})$. We similarly define expectations of properties that do not involve α.

Recall that existing approaches for network inference only model the conditional distribution $p(C|\alpha)$ assuming α to be given. In the rest of this paper, our goal is two fold: (a) augment this conditional using a prior $p(\alpha)$ to model the joint distribution $p(C, \alpha)$, (b) investigate tractability of this expectation for interesting network diffusion properties. We look at the first aspect in Sec. 4 and the second in Sec. 5.

4. A BAYESIAN FRAMEWORK

In this section, we introduce a Bayesian framework that will enable us to compute expectations of network diffusion properties. For a Bayesian analysis, we need to model α as a random variable, with a prior distribution. Assuming an *iid* prior $p(\alpha) = \prod_{uv} p(\alpha_{uv})$, the joint distribution is simply $p(C, \alpha) = p(C|\alpha) \prod_{uv} p(\alpha_{uv})$, and the posterior distribution $p(\alpha|\{c^o\}, z)$ looks as follows:

$$p(\alpha \mid \{c^o\}, z) = \prod_{uv} \frac{\bar{H}_{uv} \bar{S}_{uv} p(\alpha_{uv})}{\int_{\alpha_{uv}} \bar{H}_{uv} \bar{S}_{uv} p(\alpha_{uv}) d\alpha_{uv}} \quad (4)$$

where $\bar{H}_{uv} = \prod_{i \in A_{uv}} H(t_i|t_{z_i}; \alpha_{uv}, z)$ with $A_{uv}(z) = \{i : u_i = v, u_{z_i} = u\}$ denoting infections of v by u, $\bar{S}_{uv} = \prod_{i,j \in P_{uv}} S(t_i|t_j; \alpha_{uv}) \prod_{j \in T_{uv}} S(T|t_j; \alpha_{uv})$, with $P_{uv} = \{i, j : u_i = u, u_j = v; j \in \pi_i\}$ denoting potential infections of v by u and $T_{uv} = \{j : u_j = u, l_v < t_j\}$ denoting survivals of v from u. Observe that this decouples into terms involving individual network strengths α_{uv}. Efficient computation of network properties in Sec. 5 hinges critically on this, as on the decoupling in Eqn. 1.

Another requirement for efficient computation is analytical integration of network properties with respect to α_{uv}. For this, we need *conjugate priors*. Both Rayleigh and Exponential are special cases of the Weibull distribution (corresponding to shape parameters 1 and 2) [3]. For likelihoods involving the Weibull distribution with given shape parameter, the conjugate distribution is the *Gamma distribution*:

$$Gamma(\alpha_{uv}; a, b) = \frac{b^a}{\Gamma(a)} \alpha_{uv}^{a-1} \exp\{-b\alpha_{uv}\} \tag{5}$$

Substitution into Eqn. 4 gives us the following:

$$p(\alpha|\{c^o\}, z) = \prod_{uv} Gamma(a + \rho_{uv}(z), b + \Delta_{uv}) \tag{6}$$

where $\rho_{uv}(z) = |A_{uv}(z)|$, $\Delta_{uv} = \sum_{i,j \in P_{uv}} \delta(t_i, t_j)$ $+ \sum_{j \in T_{uv}} \delta(T, t_j)$, and $\delta(t_i, t_j) = (t_i - t_j)$ for the Exponential distribution and $\frac{1}{2}(t_i - t_j)^2$ for the Rayleigh distribution. From now on we refer to $\rho_{uv}(z)$ as ρ_{uv}.

This posterior is suitable for the network inference problem. Consider $a < 1$. Then using a suitable b, for no transmissions across an edge, $\rho_{uv} = 0$, and the posterior distribution is peaked sharply at 0. This implies that in the absence of any transmission evidence in the cascade, there is very little belief in the existence of an edge. Once an observation is made and we have $\rho_{uv} \geq 1$, the posterior distribution is unimodal and peaked at $(a + \rho_{uv})/(b + \Delta_{uv})$.

When we have large volumes of data so that $\rho_{uv} \gg a$ and $\Delta_{uv} \gg b$, the mean of the posterior approaches the MLE. While the parameterization $a < 1$ models prior belief in sparse network connections, it is also possible to make the Gamma prior noninformative if necessary, using $a, b \ll 1$ [5].

5. NETWORK DIFFUSION PROPERTIES

In this section, we consider multiple types of network diffusion properties, and analyze the tractability of computing their expectations under the posterior distribution $p(z, \alpha|\{c^0\})$. Consider, as a motivation, the network diffusion property in the introduction that counts leaders of tribes. Computing this properties is hard even when all the network diffusion variables are observed, and we will see that computing the expectations with unobserved variables is not tractable. However, we will investigate simplifications of this properties that are interesting and useful, and at the same time allow their expectations to be computed efficiently.

We consider two different categories of network diffusion properties — network centric and cascade centric. In a network-centric property, the focus is on entities in the network, such as nodes, or edges, which satisfy some constraints in the network as well as in the cascade. The 'counting leaders' property is an example in this category, with nodes in the network being the focus. A cascade-centric property, on the other hand, is about entities in the cascade, such as individual infections, which satisfy certain cascade constraints and additionally some network constraints. Before discussing more about such properties in Sec 5.2 and Sec 5.3, we first investigate conditions under which expectations of network diffusion properties are efficiently computable.

5.1 Niceness of Properties

Given the large size of real-world network diffusion data, in all of the following discussion, we consider a computation to be efficient if it is linear in the size of the network and the size of the cascade. Computing the expectation involves marginalizing out two variables: an integration over possible network strengths α, and a summation over possible network paths defined by the infection parent variables z. We first analyze these two marginalizations separately, before looking at computing the complete expectation.

Integrating over α: First, we characterize properties for which the integration over α can be performed efficiently. We call such properties *nice-α*. Intuitively, a *nice-α* property decomposes into terms that involve the parent variables z, and individual connection strengths α_{uv}. Additionally, the functions involving α_{uv} should be amenable to analytical integration with $p(\alpha_{uv}|z, \{c^o\})$ which is in the Gamma form.

DEFINITION 5.1. *A property $f(\alpha, z)$ is nice-α if it can be written as $g(z) \prod_{u,v} h_{uv}(\alpha_{uv}, z)$ or as $g(z) \sum_{u,v} h_{uv}(\alpha_{uv}, z)$ where $\int h_{uv}(\alpha_{uv}, z)\, p(\alpha_{uv}|z, \{c^o\}) d\alpha_{uv}$ can be performed analytically $\forall\ u, v$.*

THEOREM 5.1. *Let $f(\alpha, z)$ be nice-α. Then computing the z-marginal $\bar{f}_z(z) = \int_\alpha f(\alpha, z) p(\alpha|z, \{c^o\}) d\alpha$ is $O(|\boldsymbol{V}|^2)$.*

The notion of *nice-α* can be extended to properties that depend only on α and not on z. Such properties $f(\alpha)$ need to be of the form $\prod_{u,v} h_{uv}(\alpha_{uv})$ or $\sum_{u,v} h_{uv}(\alpha_{uv})$, where $\int h_{uv}(\alpha_{uv}) p(\alpha_{uv}|z, \{c^o\}) d\alpha_{uv}$ can be performed analytically for *all* z. Note that the z-marginal $\bar{f}_z(z)$ is still a function of z through $p(\alpha|z, \{c^o\})$ Also, properties that are independent of α are trivially *nice-α*. Finally, this complexity corresponds to the scenario when no edge information is available. Given a set E of potential edges, the complexity is $O(|E|)$.

Summing over z: Now we characterize properties for which the summation over infection parents z can be performed efficiently. We call such properties *nice-z*. Recall from Eqn. 1 that the posterior distribution $p(z|\alpha, \{c^o\})$ decomposes into terms involving individual z_i variables. Intuitively, the summation over z can be performed efficiently if the property $f(\alpha, z)$ also decomposes over z.

DEFINITION 5.2. *A property $f(\alpha, z)$ is nice-z if it can be written either as $g(\alpha) \prod_i h_i(z_i, \alpha)$ or as $g(\alpha) \sum_i h_i(z_i, \alpha)$*

THEOREM 5.2. *Let $f(\alpha, z)$ be nice-z. Then the α-marginal $\bar{f}_\alpha(\alpha) = \sum_z f(\alpha, z) p(z|\alpha, \{c^o\})$ can be computed in $O(\pi|C|)$ time, where $\pi = \max_i \pi_i$ is the maximum number of potential parents over all infections.*

As for *nice-α*, the notion of *nice-z* can be extended to properties that involve only z and ignore α. Note that for such properties, the α-marginal $\bar{f}_\alpha(\alpha)$ still depends on α through the posterior distribution $p(z|\alpha, \{c^o\})$. Also, a function which is independent of z is trivially *nice-z*.

Marginalizing both α and z: For computing the complete expectation in Eqn. 3, both marginalizations need to be performed. We now investigate strategies for dong this.

Interestingly, it turns out that the complete expectation can be computed efficiently and exactly for some network diffusion properties, which we call *nice-z*, α.

DEFINITION 5.3. *A property $f(\alpha, z)$ is nice-z, α if it can be written as $\prod_{u,v} g_{uv}(\alpha_{uv}) \prod_{i=1}^{|D|} \frac{h_i(z_i)}{\alpha_{u z_i u_i}}$ where analytical solutions exist for $\int g_{uv}(\alpha_{uv}) p(\alpha_{uv}|z, \{c^o\}) d\alpha_{uv}$ $\forall u, v$.*

LEMMA 5.3. *A property that is nice-z, α is both nice-α according to Defn. 5.1 and nice-z according to Defn 5.2.*

In addition to being *nice-*α and *nice-z*, *nice-z*, α properties require decoupling of α and z variables, not just in the property, but also in the posterior distribution $p(\alpha, z|\{c^o\})$. This is achieved by the $\alpha_{u_{z_i} u_i}$ terms in the property definition. These cancel out the corresponding terms in $p(\alpha, z|\{c^o\})$, which are responsible for the coupling.

THEOREM 5.4. *Let $f(\alpha, z)$ be nice-z, α. Then the expectation $\bar{f}(\alpha, z)$ can be computed in $O(\pi|\boldsymbol{C}|)) + O(|\boldsymbol{V}|^2)$ time, up to a multiplicative constant.*

The multiplicative constant in question is the inverse of the likelihood $p(\{c^o\})$ of the observed variables in the cascades. This implies that we may not be able to compute the exact value of any *nice-z*, α efficiently, but we may efficiently *compare* two different *nice-z*, α properties.

In general, there will be properties for which any one or both marginalizations cannot be performed analytically or efficiently. In such cases, we resort to Monte Carlo techniques. Here, we will assume that it is possible to draw *iid* samples $(\alpha^{(s)}, z^{(s)})$ from the joint distribution $p(\alpha, z|\{c^o\})$, and similarly $(\alpha^{(s)}) \sim p(\alpha|\{c^o\})$ and $(z^{(s)}) \sim p(z|\{c^o\})$ from the marginal distributions. In Sec 6, we describe a Gibbs Sampling algorithm for drawing such samples.

First consider properties which are *nice-*α but for which the subsequent marginalization $\sum_z f_z(z) p(z|\{c^o\})$ over z cannot be performed efficiently. For such properties, we first obtain the z-marginal $\bar{f}_z(z)$ efficiently, and then use Monte Carlo summation for z:

$$\bar{f}(\alpha, z) \approx \frac{1}{S} \sum_s \bar{f}_z(z^{(s)}), \text{ where } z^{(s)} \sim p(z|\{c^o\}),\ s = 1 \ldots S$$

Similarly, consider properties which are *nice-z* but for which the subsequent marginalization $\int f_\alpha(\alpha) p(\alpha|\{c^o\}) d\alpha$ over α cannot be performed efficiently. For such properties, we first obtain the α-marginal $\bar{f}_\alpha(\alpha)$ efficiently, and then use Monte Carlo integration for α:

$$\bar{f}(\alpha, z) \approx \frac{1}{S} \sum_s \bar{f}_\alpha(\alpha^{(s)}), \text{ where } \alpha^{(s)} \sim p(\alpha|\{c^o\}),\ s = 1 \ldots S$$

Finally, for properties where neither of the two marginalizations can be performed efficiently, we use Monte Carlo integration for both α and z:

$$\bar{f}(\alpha, z) \approx \frac{1}{S} \sum_s f(\alpha^{(s)}, z^{(s)}),$$

$$\text{where } (\alpha^{(s)}, z^{(s)}) \sim p(\alpha, z|\{c^o\}),\ s = 1 \ldots S$$

Having characterized the notion of niceness for network diffusion properties in terms of computing expectations, we now return to our motivating properties, and analyze them in this light.

5.2 Network-centric Properties

We first discuss network-centric properties, which involve computing scores for specific entities in the network, such as nodes, edges, etc. These scores are functions of the network connection strengths α and also of the cascade C. Recall that the network and the cascades are connected through the node index u_i in the individual infections.

The basic building block for network scores of network entities is the *direct* connection strength α_{uv} between nodes u and v. Using this, we can define the *second-order* network connection strength $\alpha^{(2)}_{uv}$ between u and v as $\sum_w \alpha_{uw}\alpha_{wv}$ or its approximation $\max_w \min(\alpha_{uw}, \alpha_{wv})$. Generalizing further, the r^{th}-order connection strength $\alpha^{(r)}_{uv}$ between them is $\sum_w \alpha^{(r-1)}_{uw}\alpha_{wv}$, and $\alpha^*_{uv} = \sum_{r=1}^{R} \alpha^{(r)}_{uv}$.

On the other hand, the building block for cascade scores of network entities is the *direct* transmission frequency ρ_{uv} between u and v in the cascades, defined as $\sum_{ij} I(u_i = v, z_i = j, u_j = u)$. This can be similarly generalized to define the *second-order* transmission frequency $\rho^{(2)}_{uv}$ between u and v in the cascades as $\sum_w \rho_{uw}\rho_{wv}$ or its approximation $\max_w \min(\rho_{uw}, \rho_{wv})$. The interpretation is that u frequently infects some node w, who in turn frequently infects v in the cascades. This can be similarly generalized further to define the r^{th}-order transmission frequency $\rho^{(r)}_{uv}$, and finally ρ^*_{uv}.

Node-centric Properties: We now formally define our first motivating network diffusion property, that of finding influential nodes considering both network strengths α^*_{uv} and transmission frequencies ρ^*_{uv}.

Node influence score: Intuitively, a node's influence score $f_u(\alpha, z)$ is high if it has many 'followers' v with high α^*_{uv} and high ρ^*_{uv}. One way to capture this is to define

$$f_u(\alpha, z; a, r) = \sum_v I(\alpha^*_{uv} > a) I(\rho^*_{uv}(z) \geq r) \tag{7}$$

Alternatively, we may couple α^*_{uv} and ρ^*_{uv} as $\sum_v \alpha^*_{uv}\rho^*_{uv}$ or $\sum_v {\alpha^*_{uv}}^{\rho^*_{uv}}$ Unfortunately, all of these forms are neither *nice-*α nor *nice-z* even when we consider only first and second order infections ($R = 2$). So their computation requires sampling over both α and z. But it turns out that the definition for $R = 1$ is more tractable, as we see next.

Node influence score for direct infections: This is the special case of node influence score considering only directly connected nodes in the network who are also directly infected in the cascades:

$$f_u(\alpha, z; a, r) = \sum_v I(\alpha_{uv} > a) I(\rho_{uv}(z) \geq r) \tag{8}$$

It turns out that this property is *nice-*α, so that the expectation can be calculated efficiently and exactly, in part. Though $\rho_{uv}(z)$ is itself *nice-z*, Eqn. 8 is not *nice-z* since discretization of $\rho_{uv}(z)$ through $I(\rho_{uv}(z) \geq r)$ leads to coupling across z_i variables. This implies that the alternatives $f_u(\alpha, z; a) = \sum_v I(\alpha_{uv} > a)\rho_{uv}(z)$ and $f_u(\alpha, z) = \sum_v {\alpha_{uv}}^{\rho_{uv}(z)}$ are both *nice-*α and *nice-z*, though not *nice-z*, α, providing two different routes for partly approximating their expectations.

As a further simplification of Eqn. 8, we may restrict the node influence score to consider only the network connections and ignore the cascade:

$$f(\alpha; a)_u = \sum_v I(\alpha_{uv} > a) \tag{9}$$

Interestingly this still remains *nice-*α and (trivially) *nice-z*, but not *nice-*z, α. Alternatively, we could consider only the transmission frequencies:

$$f(z, r)_u = \sum_v I(\rho_{uv}(z) \geq r) \tag{10}$$

This is (trivially) *nice-*α but not *nice-z* because of the discretization.

Edge-centric Properties: Edge-centric properties compute scores for an edge (u, v) in the network. As before, we will focus on scores involving connection strengths α_{uv} and transmission frequencies ρ_{uv}.

Edge Distribution: Given a range (r_1, r_2) for the transmission frequency, and a range (a_1, a_2) for the connection strength, this counts the number of edges (u, v) in the network whose connection strength α_{uv} and transmission frequency ρ_{uv} lie in this range.

$$f(\alpha, z) = \sum_{u,v} I(a_1 < \alpha_{uv} < a_2) I(r_1 \leq \rho_{uv}(z) < r_2) \tag{11}$$

The resultant distribution of the edges over the (α, ρ) space can be suggestive of the effectiveness of viral marketing strategies for this network, or the susceptibility of this network to epidemics. Eqn. 11 can also be viewed as the distribution of the summed (or averaged) direct node influence scores. Recall that the plots in the introduction correspond to this property.

Marginals or projections of this distribution along the ρ and α dimensions can also be useful.

$$f(z) = \sum_{u,v} I(r_1 \leq \rho_{uv}(z) < r_2); \quad f(\alpha) = \sum_{u,v} I(a_1 < \alpha_{uv} < a_2)$$

The first two edge-centric properties are *nice-*α *but not nice-z. The last property is both nice-*α *and nice-z but not nice-*z, α.

Observe that removing the binning for the α-projection recovers the well studied network inference problem.

$$f(\alpha, z) = \alpha \tag{12}$$

Computing the expectation of this score is the Bayesian formulation of the network inference problem, where we are seeking the expected network connection strengths given the cascades. This is again *nice-*α, and *nice-z*.

All the network-centric properties introduced so far are at best partly *nice*. We conclude this discussion by presenting an interesting property that is *nice-*z, α. Imagine that we are interested in finding strong edges that are not frequent, and weak edges that are frequent. For this, the following score is useful:

$$f_{uv}(\alpha, z) = \alpha_{uv}^{-\rho_{uv}(z)} \tag{13}$$

This function satisfies Def. 5.3, and therefore the complete expectation can be computed efficiently upto a multiplicative constant.

5.3 Cascade-centric Properties

For cascade centric properties, the focus is on entities in the cascade, such as individual infections, for which we compute some score based on the network as well as the cascade. We illustrate such properties using individual infections.

Infections due to Strongest Neighbor: The strongest neighbor of a node v in the network is the one with the maximum connection strength α_{uv}. We may count the number of infections for which the infecting parent u_{z_i} is the strongest neighbor for u_i in the network:

$$f(\alpha, z) = \sum_i I(u_{z_i} = \arg\max_v \alpha_{vu_i}) \tag{14}$$

We can similarly count number of infections by the n^{th}-strongest neighbor, for $n > 1$. Both properties are *nice-z*, but not *nice-*α.

As an even simpler example of a cascade property, we can consider checking the parents nodes for individual infections.

Infection parent identification: This indicates if node u is the parent of infection i.

$$f(\alpha, z)_{iu} = 1 \text{ if } z_i = u; \quad = 0 \text{ otherwise} \tag{15}$$

This is equivalent to computing the diffusion paths for a cascade. This infection-centric property is *nice-z* and also trivially *nice-*α.

It is worth observing that the *complete likelihood* $p(\{c^o\}, z \mid \alpha)$ of a cascade C given network strengths α can be seen as a cascade-centric property, where the entity of interest is the entire cascade.

$$f(\alpha, z) = p(\{c^o\}, z \mid \alpha) = \prod_{u,v} e^{-\alpha_{uv}\Delta_{uv}} \prod_i \alpha_{u_{z_i} u_i} \tag{16}$$

The *likelihood* can be viewed similarly as a cascade-centric property. Unlike complete likelihood, the likelihood is a function of only the observed infection variables, and the parent variables z are summed out.

$$f(\alpha) = p(\{c^o\} \mid \alpha) = \prod_{u,v} e^{-\alpha_{uv}\Delta_{uv}} \prod_i \sum_{j \in \pi_i} \alpha_{u_j u_i} \tag{17}$$

Both of these properties are *nice-z* (the likelihood trivially so), but not *nice-*α. In the context of test cascades, the expectation of this property can be interpreted as considering the entire posterior distribution over α, learnt from the training cascades, to explain the test cascades. In contrast, the frequentist strategy uses only a point estimate.

6. INFERENCE

We have seen in Sec. 5 that computing the expectation for network diffusion properties that are not completely *nice* requires drawing samples from the posterior distribution $p(\alpha, z|\{c^o\})$ over network strengths α and infection parents z conditioned on the observed cascades $\{c^o\}$. In this section, we propose a Gibbs Sampling framework for this. In this framework, we iterate over all latent variables, sampling a new value for it from its conditional distribution, given the current values of all other variables. Asymptotically, the samples are from the joint posterior distribution over all latent variables. For our problem, we need to draw samples from $p(z_i|\{c^o\}, \alpha, z_{-i})$ and from $p(\alpha_{uv}|\{c^o\}, z, \alpha_{-uv})$, where z_{-i} and α_{-uv} denote variables other than z_i and α_{uv}. All expressions below are for the Exponential Distribution. Expressions for Rayleigh can be derived in a similar manner.

First, the posterior distribution $p(z, \alpha \mid \{c^o\})$ over both z and α looks as follows:

$$p(z, \alpha \mid \{c^o\}) \propto \prod_{uv} \alpha_{uv}^{\rho_{uv}(z)+a-1} e^{-\alpha_{uv}(\Delta_{uv}+b)}$$

Given this, the conditional distribution for the i^{th} infection parent z_i turns out to have a very simple form:

$$p(z_i = j \mid z_{-i}, \alpha, \{c^o\}) \sim \alpha_{ji}$$

The conditional distribution for individual network strengths α_{uv} also has a simple *Gamma* density form:

$$p(\alpha_{uv} \mid \{c^o\}, z, \alpha_{-uv}) \sim Gamma(\rho_{uv} + a, \Delta_{uv} + b)$$

For network properties that are *nice-α*, only samples of z are required. In such cases, an alternative is to perform *collapsed Gibbs Sampling*, by analytically integrating out α:

$$p(z|\{c^o\}) \propto \int_\alpha p(z, \{c^o\} \mid \alpha) p(\alpha) d\alpha \propto \prod_{uv} \frac{\Gamma(\rho_{uv}(z) + a)}{(\Delta_{uv} + b)^{(\rho_{uv}(z)+a)}}$$

Given this conditional, the conditional distribution for individual infection parents z_i looks as follows:

$$p(z_i = j \mid z_{-i}, \{c^o\}) \propto \frac{(\rho^{-i}_{u_j u_i}(z) + a)}{\Delta_{u_j u_i} + b}$$

where $\rho^{-i}_{u_j u_i}(z)$ ignores the i^{th} infection for pair-wise infection counts over nodes. The collapsed Gibbs Sampling algorithm is remarkably simple. It repeatedly samples the parents of the individual infections from Multinomial distributions and updating infection counts for pairs of nodes.

The posterior over α_{uv} when $\rho_{uv} = 0$ is given by Gamma(a,b + Δ_{uv}). Since this is peaked sharply at 0, we sample α_{uv} only when $\rho_{uv} > 0$. This is significantly smaller than $|V|^2$.

Recently, the independent cascade model has been extended to handle features of individual infections [22], which is useful to capture contents of social media posts when inferring influences. Our approach can be extended in a straightforward manner to incorporate features in this way. The analysis in Sec. 5 remains unchanged since the decoupling in Eqns. 1 and 6 still hold. The Gibbs Sampling conditional distributions acquire an additional feature term. We omit further details due to space constraints.

Map-Reduce Implementation: For computing the Δ_{uv} values, each cascade can be processed in parallel. The final value for each for each u, v pair can be obtained by adding across cascades. This is exploited by the Mapper. Each Mapper computes Δ_{uv} for the set of cascades given to it and emits (v:Δ_{uv}) pairs. Each Mapper also generates a list of possible parents for each infection and emits (v:[i,π_i]) pairs.

Sampling a parent for an infection of node v requires ρ_{*v} and Δ_{*v} in case of collapsed sampler, and α_{*v} in the case of uncollapsed sampler. Sampling α_{*v} requires only ρ_{*v} and Δ_{*v}. Moreover, after sampling only ρ_{*v} and α_{*v} are needed to be updated. As a result, the sampler for each node v can be run in parallel if the set of possible parents of each infection is known. The sampling is performed in the Reducer, which exploits this parallelism. Each Reducer performs sampling for a subset of nodes. For each node v assigned, a Reducer combines information from different mappers to compute the final Δ_{uv}, and the complete list of infections of node v. It performs sampling for these infections and generates the samples. The samples from all Reducers are combined to get the complete set of samples.

7. EXPERIMENTS

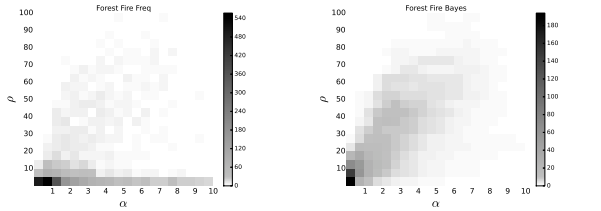
In this section, we present experimental evaluations of various network diffusion properties defined in Sec. 5 using our Bayesian approach on synthetic and real world datasets. We focus on the accuracy of the properties computations and on the scalability of our algorithms for large datasets.

Baseline: We note at the outset that this general problem has not been studied before, and that there exists no baseline for comparison. However, one potential strategy is to first recover a point estimate $\hat{\alpha}$ (e.g. MLE) of the network strengths α using a state-of-the-art approach, consider the most likely infection parents $\hat{z} = \arg\max_z p(z|\hat{\alpha}, \{c^o\})$ for $\hat{\alpha}$, and then evaluate the property $f(\hat{\alpha}, \hat{z})$ at $(\hat{\alpha}, \hat{z})$. While this suffers from deficiencies outlined in Sec. 3, this is the best possible baseline using existing network inference techniques. As the state-of-the-art network inference approach for the continuous time independent cascade model, we used the featureless version of MONET [22]. We do not use NETRATE [6], since it does not support multiple infections of a node in a cascade. In the rest of this section, we will refer to this approach as the frequentist plug-in approach (**FP**), and to our proposed approach of computing expectations as the Bayesian Expectation approach (**BE**). We have used the Exponential distribution for all experiments. For the Gamma prior we use a = 0.00001 and b = 0.1.

Synthetic data experiments: We first conducted experiments on multiple synthetic datasets. These experiments allowed us to evaluate accuracy against a gold-standard, which is unavailable for most real-world network diffusion datasets. Secondly, these helped us analyze our proposed approach for different families of graphs. Following earlier experiments on network inference [7, 6], we created synthetic graphs with 1024 nodes using Forest Fire (FF), Random (Rnd), Hierarchical (HI) and Core-Periphery (CP) Graph models, the last three being instances of Kronecker Graph models. We used parameter values [0.5, 0.5; 0.5, 0.5] for Rnd, [0.9,0.1;0.1,0.9] for HI, [0.9,0.5;0.5,0.9] for CP, following Gomez-Rodriguez et. al. [6]. To generate weights α_{uv} for each edge (u, v), we sampled uniformly from $(0.01, 10)$ [3]. We then generated 20 *splitting* cascades over these graphs with 2 randomly chosen seeds for each cascade. Finally, we obtained 2046 edges and 48947 infections for the Random graph, 1496 and 38046 for the Hierarchical, 2042 and 58062 for the Core-Periphery, and 2023 and 55274 for the Forest Fire graph.

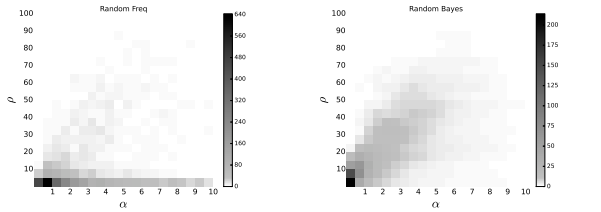
Recall that one of the reasons behind the synthetic data experiments is to be able to evaluate accuracy. For the infection parents z, we considered the true parents as the gold-standard. However, for the real-valued network connections α_{uv}, the true values are not always possible to recover from finite length cascades. For example, it is impossible to recover the strength for any edge that has no transmission in the cascade. Therefore, we considered as our gold-standard the best achievable α_{uv} given the true infection parents in the cascades: $\alpha^* = \arg\max_\alpha f(\{c^o\}, z^*; \alpha)$. For accuracy in case of a scalar property, we used absolute error between the gold-standard $f(\alpha^*, z^*)$ and the computed value of the property, and for vectors and matrices the root mean squares of the individual errors.

(A) Network-centric Properties: In this category, we first evaluate the edge-distribution (Eqn. 11) as an example of a **property on edges**. Evaluating accuracy for this property is challenging because of the threshold parameters a and r. We discretized the α and the ρ ranges, and within each region of the (α, ρ) space, computed the actual, BE and FP values of these properties, and the errors for BE and FP.



(a) Frequentist (b) Bayesian Expectation

Figure 3: Reconstructed edge distribution: FF



(a) Frequentist (b) Bayesian Expectation

Figure 4: Reconstructed edge distribution: Rnd

The actual plots for the four networks were introduced in Fig. 1. The FE and BE reconstructions for the Core-Periphery graph were also introduced earlier in Fig. 2. The reconstructions for the other three graphs are shown in Figs. 3, 4, and 5. It can be seen quite clearly that while BE is able to reconstruct the actual distributions to a reasonable extent for all 4 graphs, FP does quite poorly. In fact, the FP reconstruction looks similar in all 4 cases, and fails to pick up the signatures for the different graphs.

We also calculated the actual errors for the two approaches over the (α, ρ) space. Since it is difficult to visualize the plots in 2D, we evaluated the projections on the α-dimension and z-dimension (Eqn. 5.2) for the edge distribution for the 4 graphs.

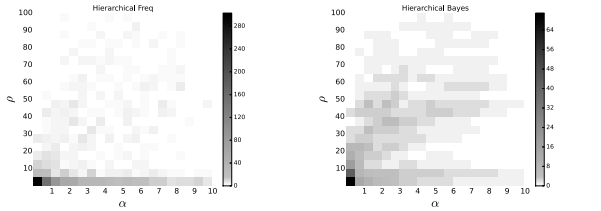
Table 1: α-proj. for edge distribution: abs. error

NW	CP		HI		Rnd		FF	
α	BE	FE	BE	FE	BE	FE	BE	FE
0 1	591	6514	156	1501	470	4004	432	4008
1 2	169	1587	5	287	34	1085	30	1039
2 3	13	618	18	166	6	497	8	379
3 4	9	340	1	87	6	265	15	196
4 5	8	159	2	57	32	139	9	101
5 6	6	138	1	51	14	127	11	61
6 7	3	75	1	34	7	86	13	62
7 8	1	82	**14**	**9**	9	70	4	64
8 9	2	48	8	17	13	45	1	38
9 10	1	46	1	16	4	44	4	36

Table 2: ρ-proj. for edge distribution: abs. error

NW	CP		HI		Rnd		FF	
ρ	BE	FE	BE	FE	BE	FE	BE	FE
0 10	524	4373	69	1032	344	3339	287	2903
10 20	**348**	**228**	14	15	**108**	**8**	89	21
20 30	20	82	3	0	40	87	20	51
30 40	59	91	1	14	28	64	29	73
40 50	47	100	3	19	13	44	10	43
50 60	43	59	5	16	6	16	8	34
60 70	19	27	0	3	1	13	9	12
70 80	1	1	1	0	2	6	2	3
80 90	4	4	2	0	0	4	2	12
90 100	3	5	2	1	1	2	2	8

In Tab. 1, we record the errors for BE α-projection and the FP α-projection for different α-intervals. We can see

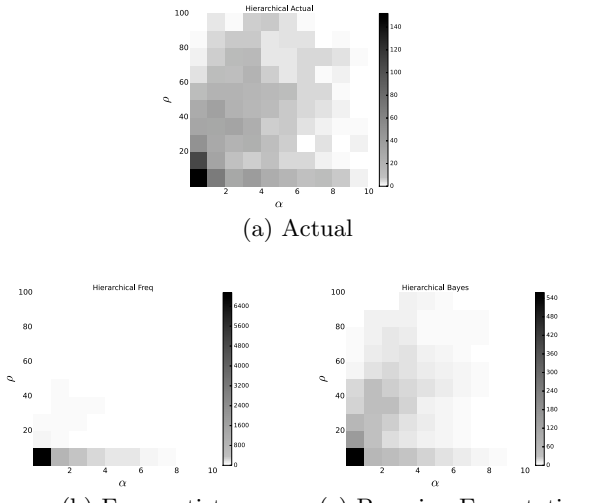


(a) Frequentist (b) Bayesian Expectation

Figure 5: Reconstructed edge distribution: HI

that for the α-projection, the FP errors are an order of magnitude bigger for all intervals, except for $\alpha \in (7, 8)$ for Hierarchical. Similarly, in Tab. 2, we record the errors for BE ρ-projection and the FP ρ-projection for different ρ-intervals. In this case as well, FP error is significantly lower only for the $(10, 20)$ interval for CP and Rnd.

Finally, we come to **properties of nodes**. We evaluated direct node influence (Eqn. 8), and indirect node influence for 2^{nd}-order neighbors (Eqn. 7) for the 4 graphs. Again, we partitioned the (α, ρ)-space into regions. However, reporting detailed results is even harder here, since we have actual, BE and FP scores *for each node* in each α, ρ-region. We consider the sum (or average) of the influence scores over all nodes. Recall that one interpretation of the edge-distribution is the distribution of the sum of direct influence scores over all nodes. So the edge-distribution evaluation above additionally serves as an evaluation of the *direct node influence scores* at an aggregate level.



(a) Actual

(b) Frequentist (c) Bayesian Expectation

Figure 6: Indirect node influence distribution: HI

For *indirect node influence*, due to space constraints, we show the (averaged) indirect node influence distribution only for the Hierarchical graph in Fig. 6. Again, we see that BE is able to pick up the signature of the distribution to a reasonable extent, whereas FP has failed completely.

In Tab. 3, we report the aggregated errors over all nodes and over all (α, ρ) regions for both direct and indirect influence scores. We have scaled down the values by the total number of nodes, which is 1024. We again see that the BE

Table 3: Node influence scores: agg. error

NW	CP		HI		Rnd		FF	
	BE	FP	BE	FP	BE	FP	BE	FP
Dir	29	124	22	46	30	104	30	98
InDir	62	417	15	98	27	288	27	273

errors are significant smaller than the FP errors across the board.

(B) Cascade-centric Properties: Under cascade-centric properties, we evaluate infections due to n^{th} strongest neighbor (Eqn. 14) for $n = 1, 2, 3$.

Table 4: Infections by n^{th}-strongest nbr: abs. error

NW	CP		HI		Rnd		FF	
n	BE	FP	BE	FP	BE	FP	BE	FP
1	3711	22090	631	13537	578	21613	127	23476
2	2374	6171	691	2506	1352	3533	1284	**712**
3	191	**59**	194	3924	152	4825	388	6828

Tab. 4 records the absolute error for the number of infections by the n^{th}-strongest neighbor for $n = 1, 2, 3$. Notice that FP has very high errors for $n = 1$. There are just two instances where FP works better than BE: for $n = 2$ in Forest Fire and for $n = 1$ in Core-Periphery, where the values are comparable. In all other cases, FP has significantly higher error than BE.

Likelihood, Network Inference and Parent Identification: The performance improvement of BE over FP in the experiments so far is attributable to two reasons: (a) the Bayesian approach of using the full posterior distribution instead of the frequentist point estimate, and (b) many-to-one mapping of network variables to property values. We now look at experiments using the basic inference problems for network analysis, and generalization ability on held-out data, which are one-to-one network diffusion properties in our formulation.

Table 5: Loglikelihood for synthetic data

NW	CP	HI	Rnd	FF
	BE, FP	BE, FP	BE, FP	BE, FP
Test	1.0e4, 0.6e4	6.5e3, 2.4e3	1.1e4, -1.5e4	1.2e4, 926
Train	2.8e4, 3.6e4	2.0e4, 2.2e4	2.3e4, 2.9e4	2.8e4, 3.3e4

In Tab. 5, we record the train and test likehoods for the 4 synthetic datasets. We see that BE consistently has higher test likelihood, while the train likelihood is higher for FP, suggesting overfitting.

Table 6: Network Inf. (NI) & Parent Id. (PI)

NW	CP		HI		Rnd		FF	
	BE	FP	BE	FP	BE	FP	BE	FP
NI	0.116	2.553	0.884	3.210	0.147	17.483	0.329	736.821
PI	0.533	0.406	0.861	0.783	0.757	0.646	0.770	0.674

In Tab. 6, we record the errors in recovery of α for BE and FP. Observe that the errors are consistently lower for BE across the 4 datasets. In fact, the FP errors are very high for the Random and Forest Fire datasets. In Tab. 6, we also see that parent identification accuracy of BE is consistently around 10% more than that of FP. These three experiments demonstrate that the Bayesian approach for network diffusion analysis has advantages even for one-to-one properties.

(C) Iterations vs Error: Before moving on to experiments on real-world data, we make a note about Gibbs Sampling iterations. Gibbs Sampling algorithms often take thousands of iterations to converge, which can be a serious problem for large real-world datasets. The number of iterations required for the Gibbs Sampler to converge can be greatly reduced by choosing a good initialization. We use $\hat{z} = \arg\max_z p(z \mid \{c^o\}; \hat{\alpha})$ as the initial z, where $\hat{\alpha_{uv}} = |P(u,v)|/\Delta_{uv}$. $\hat{\alpha}$ is an approximation of α_{MLE} where we use $|P(u,v)|$ which is purely a function of that data instead of ρ_{uv} in the numerator. In our experiments, we observe that the accuracy increases very sharply in the initial iterations, and is close to the best value within 100-200 iterations.

Experiments on real-world data: We now report experiments on real-world data, where the graph structures are more complex than the synthetic settings. The underlying diffusion process is also likely to be different from the Independent Cascade model, which our models assume, and which we used for generating the synthetic cascades.

The **Meme Tracker** dataset[1] records the diffusion of "memes" or catch-phrases across 5000 most active blogs and news sites between March 2011 and February 2012. The flow of each meme corresponds to one cascade. Related memes are grouped into one topics. For our experiments, we selected 5 topics, 2 of which have been used in earlier experiments involving non-stationary networks [9], and 3 others for which the networks are likely to be stationary. Basketball has 1187 Nodes, 130317 Infections and 2663 cascades, Alcohol has 1549 nodes, 151136 infections and 3018 cascades, Technology has 1797 nodes, 338931 infections and 6658 cascades, NBA has 1891 nodes, 179864 infections and 3637 cascades, and Occupy has 1601 nodes, 147135 infections and 2851 cascades. In each topic, we consider all sufficiently long cascades (length > 30). We split the cascades randomly to generate the training and test cascades(80-20), and then prune infections of those users in test cascades who are not present in the training cascades.

Table 7: Loglikelihood for Meme Tracker

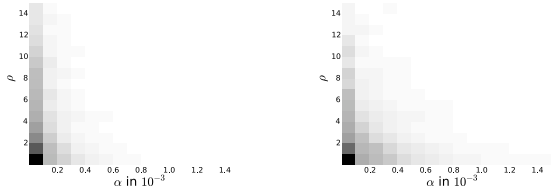
	Bball	Alcohol	Tech.	NBA	Occupy
BE	-1.5e6	-5.8e5	-6.6e5	-8.9e5	-5.1e5
FP	-3.5e6	-8.9e5	-2.6e6	-1.1e7	-1.2e6

Since no gold-standard is available for α or z for this dataset, we compared BE and FP using loglikelihood on held-out test data, with the knowledge of α learnt from training data. In Tab. 7, we report the loglikelihood values for the 5 selected topics. We can see that the BE values are significantly better than the FP values. Recall that this is the best scenario for the baseline since likelihood, as a network diffusion property, is a one-to-one function of α for this problem. This suggests that BE is likely to outperform FP to a larger extent for other properties on real-world datasets.

We also computed the proposed network-centric and cascade-centric properties for Meme Tracker using BE. In Fig. 7, we plot the edge distribution for two of the topics. Recall that there is no gold-standard for this. We can see that the nature of the plots is different from all of the synthetic datasets. The mass is more concentrated towards weaker, infrequent edges. We suspect that this is because of the way users were sampled for this dataset.

Scaling experiments: We also experimented with larger volumes of the Meme Tracker data using our map-reduce implementation. We created increasingly larger dataset sizes by randomly sampling cascades and checking the execution

[1]http://snap.stanford.edu/infopath//data.html



(a) Tech (b) Alcohol

Figure 7: Edge distribution for Meme Tracker topics

Table 8: Millions of Infections vs time (secs)

# Infections	Time (12 nodes)	Time (1 node)
15	552	3635
31	888	6873
43	1311	10277
63	1948	14783

time for 100 iterations of Gibbs Sampling. We performed experiments on a Intel Xeon server with 100GB RAM, which supports 12 mapper/ reducer tasks in parallel.

In Tab. 8, we record execution time with increasing data size on 12 processor nodes and compare against the time taken on a single node. We can see that the map-reduce implementation allows us to scale our analysis by providing a (roughly) linear speed-up in terms of number or processor nodes.

In summary, the experiments clearly demonstrate that computing expectations under the posterior distribution leads to significantly better reconstruction of a wide variety of network diffusion properties. The proposed Bayesian framework that combines exact efficient computation with Gibbs Sampling based approximations outperforms state-of-the-art algorithms even for the well-studied network inference and parent identification problems, and in generalizing to held-out test data. The map-reduce implementation is promising in terms of scaling up the analysis to study properties of large network diffusion datasets.

8. CONCLUSIONS

In this paper, we have investigated the novel problem of computing expectations of properties of network diffusions involving hidden variables. We have proposed a Bayesian framework for computing such expectations, and proposed and characterized network diffusion properties that can be handled efficiently in this framework. In experiments over synthetic and real world datasets, we have shown that we are able to reconstruct network properties significantly more accurately than a frequentist baseline.

9. REFERENCES

[1] N. Barbieri, F. Bonchi, and G. Manco. Influence-based network-oblivious community detection. In *ICDM*, 2013.

[2] F. Bonchi. Influence propagation in social networks: A data mining perspective. *IEEE Intelligent Informatics Bulletin*, 12(1), 2011.

[3] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *NIPS*, 2013.

[4] N. Du, L. Song, A. Smola, and M. Yuan. Learning networks of heterogeneous influence. In *NIPS*, 2012.

[5] A. Elfessi and D. Reineke. A bayesian look at classical estimation: The exponential distribution. *Journal of Statistics Education*, 9(1), 2001.

[6] M. Gomez-Rodriguez, D. Balduzzi, and B. Schölkopf. Uncovering the temporal dynamics of diffusion networks. In *ICML*, 2011.

[7] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *KDD*, 2010.

[8] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Modeling information propagation with survival theory. In *ICML*, 2013.

[9] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf. Structure and dynamics of information pathways in online media. In *WSDM*, 2013.

[10] A. Goyal, F. Bonchi, and L. Lakshmanan. Discovering leaders from community actions. In *CIKM*, 2008.

[11] A. Goyal, F. Bonchi, and L. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, 2010.

[12] A. Goyal, F. Bonchi, and L. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 2011.

[13] A. Guille, H. Hacid, C. Favre, and D. A. Zighed. Information diffusion in online social networks: A survey. *SIGMOD Rec.*, 42(2), July 2013.

[14] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD*, 2003.

[15] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis. Strip: Stream learning of influence probabilities. In *KDD*, 2013.

[16] L. Macchia, F. Bonchi, F. Gullo, and L. Chiarandini. Mining summaries of propagations. In *ICDM*, 2013.

[17] Y. Mehmood, N. Barbieri, F. Bonchi, and A. Ukkonen. Csi: Community-level social influence analysis. In *ECML PKDD*, 2013.

[18] C. Milling, C. Caramanis, M. S., and S. Shakkottai. Network forensics: Random infection vs. spreading epidemic. In *ACM Sigmetrics*, 2012.

[19] P. Netrapalli and S. Sanghavi. Finding the graph of epidemic cascades. In *ACM SIGMETRICS*, 2012.

[20] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for missing data in information cascades. In *WSDM*, 2011.

[21] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Learning continuous-time information diffusion model for social behavioral data analysis. In *Adv. in Mach. Learning*, 2009.

[22] L. Wang, S. Ermon, and J. Hopcroft. Feature-enhanced probabilistic models for diffusion network inference. In *ECML-PKDD*, 2012.