Correcting for Missing Data in Information Cascades

Eldar Sadikov[†] Montserrat Medina[‡] Jure Leskovec[†] Hector Garcia-Molina[†]
Stanford University
[†]{eldar,jure,hector}@cs.stanford.edu, [‡]mmmedina@stanford.edu

ABSTRACT

Transmission of infectious diseases, propagation of information, and spread of ideas and influence through social networks are all examples of diffusion. In such cases we say that a contagion spreads through the network, a process that can be modeled by a cascade graph. Studying cascades and network diffusion is challenging due to missing data. Even a single missing observation in a sequence of propagation events can significantly alter our inferences about the diffusion process.

We address the problem of missing data in information cascades. Specifically, given only a fraction $\mathcal{C}'$ of the complete cascade $\mathcal{C}$, our goal is to estimate the properties of the complete cascade $\mathcal{C}$, such as its size or depth. To estimate the properties of $\mathcal{C}$, we first formulate *k-tree* model of cascades and analytically study its properties in the face of missing data. We then propose a numerical method that given a cascade model and observed cascade $\mathcal{C}'$ can estimate properties of the complete cascade $\mathcal{C}$. We evaluate our methodology using information propagation cascades in the Twitter network (70 million nodes and 2 billion edges), as well as information cascades arising in the blogosphere. Our experiments show that the k-tree model is an effective tool to study the effects of missing data in cascades. Most importantly, we show that our method (and the k-tree model) can accurately estimate properties of the complete cascade $\mathcal{C}$ even when 90% of the data is missing.

Categories and Subject Descriptors: H.2.8 **[Database Management]**: Database Applications – *Data mining*

General Terms: Algorithms, theory, experimentation.

1. INTRODUCTION

Social and information networks are a fundamental medium for the spread of information, ideas, viruses and behavior. A cascade graph can be used to represent the contagion across the network. For example, if Alice is connected to Bob in a social network and Bob participates in the "Fight Against Cancer" campaign, he may influence Alice to do the same. Or similarly, Bob may spread information to Alice, if Bob reads some article and shares it with Alice. As information or actions spread from a node to node through the social network, a *cascade* is formed. Nodes of the cascade are the nodes of the network that performed an action of interest and edges represent influence relations [3, 7, 8]. Thus, when Alice joins the campaign under Bob's influence, we observe a directed edge from Bob to Alice in the cascade. We define social networks and cascades formally in Section 2, but to illustrate now, Figure 1 gives a network and two types of cascades.

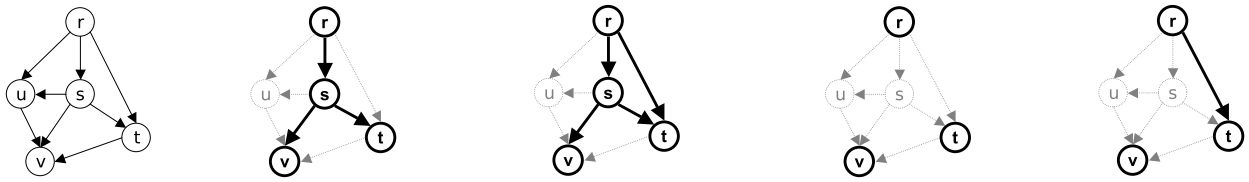
We may not observe all actions performed by the nodes of interest, and hence our cascades may be incomplete, i.e., have missing data. For example, Figures 1(d, e) show cascades when some of the data (i.e., actions of node s) is missing. The cascades with missing data may no longer have the same properties (e.g., depth, the number of edges) as the original cascade, and may not even be connected. Here, we address the problem of estimating properties of a complete cascade $\mathcal{C}$ from a small observed part $\mathcal{C}'$ of the complete cascade. Specifically, can we infer properties, like size and depth, of the complete cascade, when data is missing?

There are a number of reasons why cascades may have missing data. Most social networks do not provide full information about their user activity and thus we only observe a subset of users participating in the cascade. For example, Twitter does not provide public access to its full stream of tweets and most Facebook users keep their activity and profiles private. Furthermore, there have been growing concerns about Facebook's privacy policy, which indicates that users are generally reluctant to share their data. Finally, full information may not be available because of the costs of collecting it. Overall, the rapid growth of the social networks themselves, the increasing volume of their generated data, and the growing concerns of users over privacy will likely to only exacerbate the problem of missing data over time.

Why estimate properties of complete cascades? Processes that form cascades in a social network have been studied in a number of domains, including the diffusion of medical and technological innovations [22], adoption of strategies in game-theoretic settings [6], product adoption, promotion, and viral marketing [7, 14]. Diffusion and cascades have been studied in the context of Facebook [25], Twitter [12], Flickr [4], blogs [17], and email chain-letters [18]. To study diffusion processes underlying the cascades, one needs accurate knowledge of the cascade properties, such as node out-degree, in-degree, or cascade depth. However, observed properties may differ from the properties of the complete cascade which highly biases inferences about the diffusion processes.

Cascades are also essential for selecting trendsetters for viral marketing [21, 10], finding inoculation targets in epidemiology [20], and explaining trends in blogosphere [9]. Missing data in information cascades can have large effect on these applications. Consider, for example, the problem of influence maximization for viral marketing. The task here is to select a set of most influential nodes in the network where the influence of a node could be the average size

WSDM'11, February 9–12, 2011, Hong Kong, China.



(a) Network (b) Influence cascade $\mathcal{C}_i$ (c) Network cascade $\mathcal{C}_n$ (d) Sampled influence cascade $\mathcal{C}_i'$ (e) Sampled network cascade $\mathcal{C}_n'$

Figure 1: Missing data in cascades. (a) A social network. (b) Influence cascade: we observe edges over which the information propagated. (c) Network cascade: we only observe participating nodes, not propagation edges; edges are inferred from the network based on time order. (d, e) Influence and network cascades, respectively, with missing data (information about node s is missing).

of the cascades it creates. However, a cascade can become disconnected with missing data, so the size cannot be reliably estimated. Accordingly, the influence maximization algorithm will perform poorly and the targeted marketing campaign will likely fail.

Related work on missing data. Missing data in networks is a longstanding but relatively poorly understood problem. Related to our work here are the works that study the effects of missing data on measured properties of social networks [11] and the study of biases when obtaining a graph of the Internet based on measurements [13, 1]. Another related line of work is on sampling in large networks [15, 19, 24], where given a large network we would like to find some procedure to sample a small set of nodes such that important structural properties of the network are preserved.

In terms of the effects of missing data in information cascades prior work is practically nonexistent. The exception is the recent work by Choudhury et al. [5] that considers the effect of various sampling strategies on the measured properties of diffusion series (similar notion to cascades). While this work tries to find a sampling strategy that least distorts the observed properties, our work here differs. We work under consideration of uniform random sampling, where each node is missing independently with probability $1 - \sigma$. However, we are able not only to both *analytically* and empirically understand the distortion created by sampling (i.e., missing data) but also to *correct* for the distortion (i.e., infer properties of the complete cascade). To our knowledge this is the first attempt to analytically understand the distortions under missing data and, more importantly, to correct for them. This is especially challenging as cascades, tree-like graphs, are very fragile, easily disconnected even with a small fraction of missing nodes.

Outline. In the following we first propose a k-tree model of cascades and derive properties of the resulting cascades, such as size, number of edges, etc. Then, given an observed cascade $\mathcal{C}'$ with missing data, we show how to select a "proxy" k-tree model that best approximates $\mathcal{C}'$. The model can then be used to estimate the properties of the complete cascade $\mathcal{C}$. We experimentally show that the properties estimated via a proxy cascade are much closer to the true properties of $\mathcal{C}$ than the observed properties on $\mathcal{C}'$ for any sample ratio σ less than 0.7. Hence, we can effectively correct for missing data.

We evaluate our findings on a Twitter social network of 70 million nodes and 2 billion edges. We run our experiments on more than 1 billion tweets. In addition, we also study information diffusion cascades formed on the blogosphere. We show that our methodology can reliably infer structural properties of complete cascades with as much as 90% of missing data.

2. PROBLEM STATEMENT

We model a social network, over which cascades unfold, as a directed graph $G(V, E)$, where nodes V represent entities (e.g., people, web sites, blogs) and edges E represent directed interactions. For example, in network in Fig. 1(a), nodes r and s interact with t.

We focus on nodes of G that have performed a particular type of action, e.g., joined the "Fight against cancer" campaign, participated in an online poll, or bought a camera. The process starts with an initially active node r (the root) and the decision to perform an action can be seen as an infection transmitted over the edges of G from a node to node as a result of their interaction. An *action sequence* A is a sequence of pairs (s, t), one pair for each node t that performed the action of interest, where s influenced t. For example, if s bought a camera under the influence of r, then (r, s) appears in the action sequence. The initially active node r is not influenced by anyone, denoted by $(\perp, r)$. The order of the pairs in A represent the order in which nodes performed the action of interest. For the scenario so far, we have $A = \langle(\perp, r), (r, s)\rangle$. We assume a node can be influenced by at most one other node, much like a disease is transmitted to a person from a specific individual in epidemic models [2]. If a node performs the action multiple times, we only consider the first action.

The subgraph of G defined by the influence relations in the action sequence forms an *influence cascade* $\mathcal{C}_i$. The nodes in $\mathcal{C}_i$ are the nodes in the action sequence and an edge (r, s) is in $\mathcal{C}_i$ if $(r, s) \in A$ (since actions only spread along the edges of G then $(r, s) \in E$). Figure 1(b) shows one possible influence cascade, where r is the initially active node, which then influenced node s, which in turn influenced nodes v and then t. Note, there is only one *root* node, which is not influenced by any other nodes – the first node in the action sequence. Influence cascades are trees because nodes cannot repeat in the action sequence and each non-root node s has one incoming edge (from the influencer of s). Such tree-like cascades are common in real data: we will show in Section 5, for example, that influence cascades arise in the blogosphere.

In some real-world scenarios, however, it may be hard to identify an influencing node. We may only observe action sequence pairs of the form $(\emptyset, u)$ where we know that node u performed an action but do not know which node influenced the action. In this case, we construct a *network cascade* $\mathcal{C}_n$. The nodes in $\mathcal{C}_n$ are the nodes in the action sequence and an edge (r, s) is in $\mathcal{C}_n$ if $(r, s) \in E$ and r appears before s in the action sequence. Intuitively, there is an edge between r and s in the network cascade if r performed the action before s and r is connected to s in the social network G. Network cascades, as we will see in Section 5, arise on Twitter.

For example, Figure 1(c) shows a network cascade. In particular, note that t is now connected to all nodes that could have possibly influenced it. We call the edges that are in the network cascade but not in the influence cascade *spurious*, e.g., edge (r, t) is spurious. Since each node may have more than one incoming edge, network cascades are not trees but rather directed acyclic graphs (DAGs).

As discussed in Section 1, we may not observe the complete action sequence, so we may have missing data in our cascades. In particular, say, we have a sample of the action sequence. Then if we use the sampled action sequence instead of the complete action sequence in the definitions above, we obtain a *sampled influence*

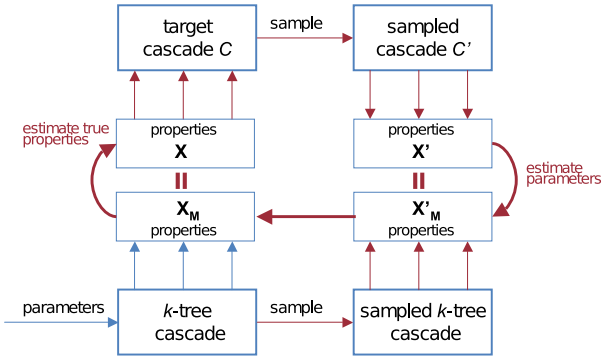


Figure 2: Methodology.

cascade or a *sampled network cascade*. For example, Figures 1(d) and (e) show the sampled influence and sampled network cascades for the sampled action sequence where information about node s is missing. Note how cascades become disconnected in both cases.

In this paper, we assume that missing data is a result of uniform random sampling. Specifically, each node in the complete action sequence is included in the sample at random with probability σ, independent of other nodes. We call σ the *sample ratio*. We consider uniform random sampling because this is the most common sampling strategy and is in fact used by Twitter for its public stream of tweets.

Methodology. Our goal is to obtain a set of properties $\mathbf{X}$ of a given complete (influence or network) cascade $\mathcal{C}$. For example, size and depth are two such properties. However, we do not have access to the cascade $\mathcal{C}$ itself but to a sample $\mathcal{C}'$. Thus, we can only compute the properties $\mathbf{X}'$ of the sample $\mathcal{C}'$. Note that the properties $\mathbf{X}'$ can be very different from the properties $\mathbf{X}$ of $\mathcal{C}$. For example, in Figure 1, the depth of the influence cascade is 2, while the depth of its sample is 0.

Figure 2 illustrates our approach. To estimate the properties $\mathbf{X}$ of $\mathcal{C}$, we first propose a *k-tree* model of cascades. The box labeled "k-tree cascade" represents a parameterized family of cascades. The samples of these cascades are represented by the box labeled "sampled k-tree cascade." We can compute the properties $\mathbf{X}_\mathbf{M}$ of the complete k-tree cascade and $\mathbf{X}'_\mathbf{M}$ of the sampled k-tree cascade.

Our strategy now is to find a sampled k-tree cascade with properties $\mathbf{X}'_\mathbf{M}$ similar to the properties $\mathbf{X}'$ of the sampled cascade $\mathcal{C}'$. For example, we can find a complete k-tree cascade (i.e., its parameters) by finding a sampled k-tree cascade with the expected number of edges equal to the number of edges in $\mathcal{C}'$. Once we find such k-tree cascade, we can approximate the properties $\mathbf{X}$ of the complete target cascade $\mathcal{C}$ by the properties $\mathbf{X}_\mathbf{M}$ of the complete k-tree cascade. For example, we estimate the size of $\mathcal{C}$ as the size of the complete k-tree cascade.

We start in Section 3 by defining our k-tree model of cascades and analytically derive their important properties. Then in Section 4, we discuss how to estimate model parameters so that $\mathbf{X}'_\mathbf{M}$ and $\mathbf{X}'$ are similar. Finally, in Section 5 we experimentally show the soundness of our approach. For our evaluation, we need complete target cascades in order to check whether $\mathbf{X}$ matches $\mathbf{X}_\mathbf{M}$. We consider two types of target cascades: (a) synthetic cascades obtained from a simulated action propagation process, (b) actual cascades obtained from Twitter and blogs. In addition, through our experiments, we show that the k-tree model is an effective tool to study the sensitivity of cascade properties to sampling.

3. CASCADE MODEL

Next we introduce k-tree model of cascades. The model allows for mathematical analysis of cascade properties without the need

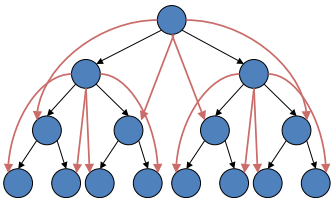


Figure 3: k-tree cascade with branching factor $b = 2$, number of parents $k = 2$, and depth $h = 3$.

Symbol	Description
σ	Fraction of $\mathcal{C}$ nodes observed in $\mathcal{C}'$ (sample ratio)
p	Probability of observing a node in the k-tree model
b	Number of children (out-degree) via influence edges
h	Height of the tree on influence edges
k	Number of parents (in-degree) of non-root nodes
n	Number of nodes in the complete k-tree, $n = \frac{b^{h+1}-1}{b-1}$
m	Number of nodes in the sampled k-tree, $m = p \cdot n$

Table 1: Table of symbols

for asymptotic analysis. We cannot assume cascades of infinite size or depth as real cascades are rather shallow. Obtaining precise constant factors in the expressions describing cascade properties is essential in order to be able to reconstruct the complete cascade.

A *k-tree* $\Gamma(b, h, k)$ is generated from a balanced tree of height h and branching factor b. We then augment each node of the tree with $k-1$ edges from its $k-1$ closest ancestors, starting from its grandparent. Thus nodes have k parents, except for nodes near the root which do not have enough ancestors. Figure 3 shows a k-tree with $b = 2$, $h = 3$, and $k = 2$. Original edges of the balanced tree model influence edges while the $k-1$ additional edges per node model spurious edges. In Figure 3, influence edges are darker and spurious edges are lighter.

As noted earlier, influence cascades are trees, so we model influence cascades by k-trees with $k = 1$, equivalent to regular balanced trees. Network cascades, on the other hand, are modeled by k-trees with $k > 1$, since each node of a network cascade can have more than one parent. Even though real cascades may be imbalanced, we have found that they are usually shallow and follow a monotonic growth pattern where the number of nodes at any depth i usually increases as i increases. Hence, k-trees are a good model of the real cascades. We will see in Section 5, that the effect of missing data on the real cascades is the same as it is on the k-trees.

When a k-tree cascade has missing data, we refer to it as *sampled k-tree*. We use $\Gamma(p, b, h, k)$ to refer to a k-tree $\Gamma(b, h, k)$ with p fraction of its nodes observed. Each node is included in the sampled k-tree with probability p, independently of other nodes.

In what follows, we derive structural properties $X1, \ldots, X6$ of sampled k-trees $\Gamma(p, b, h, k)$ as a function of the four parameters p, b, h, and k. Some of the properties we study are important in their own right. Others make it easier to match a k-tree cascade to the target cascade, as described in Section 2. Table 1 provides a reference for the symbols used in the theorems and proofs.

X1: Number of nodes. We first derive an expression for the expected number of nodes m in a sampled k-tree $\Gamma(p, b, h, k)$.

THEOREM 1. *The expected number of nodes m in a sampled k-tree $\Gamma(p, b, h, k)$ is $p\frac{b^{h+1}-1}{b-1}$.*

PROOF. Let n be the number of nodes in the complete k-tree. By summing the geometric series we get $n = \sum_{i=0}^{h} b^i = \frac{b^{h+1}-1}{b-1}$. We know that $m = n \cdot p$ (expectation of binomial random variable with parameters (n, p)). Then $m = p\frac{b^{h+1}-1}{b-1}$. □

X2: Number of edges. Observe that any regular tree (i.e., $k = 1$) has $(n-1)$ edges (one incoming edge per each non-root node), while any k-tree generally has close to $k(n-1)$ edges (k incoming edges per each non-root node). More formally, the number of observed edges is given by the following theorem:

THEOREM 2. *The expected number of edges in a sampled k-tree $\Gamma(p,b,h,k)$ is equal to:*

$$\frac{p^2}{b-1}\left(\frac{b(1-b^k)}{b-1} + kb^{h+1}\right)$$

PROOF. Let Z_i be the random variable representing the number of nodes at level i and W_i be the random variable representing the number of observed parents of a node at level i. Then the number of edges is equal to $\sum_{i=0}^{h} Z_i \cdot W_i$. By linearity of expectation, $E[\sum_{i=0}^{h} Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i \cdot W_i]$. Furthermore, since Z_i is independent of W_i (because each node is observed independently of other nodes), $\sum_{i=0}^{h} E[Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i]E[W_i]$. Since Z_i is a binomial random variable with parameters (b^i, p) and W_i is a binomial random variable with parameters $(\min(\{i,k\}), p)$, $E[\sum_{i=0}^{h} Z_i \cdot W_i] = \sum_{i=0}^{h} p^2 b^i \cdot \min(\{i,k\}) = \frac{p^2}{b-1}\left(\frac{b(1-b^k)}{b-1} + kb^{h+1}\right)$. □

X3: Number of isolated nodes. A node becomes isolated if and only if its parents and its children are not observed. Thus, to derive the number of isolated nodes, let's first derive the number of children each node has.

LEMMA 1. *The expected number of children of a node at level $i (i \le h)$ in a sampled k-tree $\Gamma(p,b,h,k)$ is $p\frac{b^{l+1}-b}{b-1}$, where $l = \min(k, h-i)$.*

PROOF. Any non-leaf node at level $i \le (h-k)$ has outgoing edges to all of its descendants at the next k levels. Hence, each non-leaf node at level $i \le (h-k)$ has the following number of outgoing edges in the complete tree: $\sum_{j=1}^{k} b^j = (\frac{b^{k+1}-1}{b-1} - 1) = \frac{b^{k+1}-b}{b-1}$. If $i > (h-k)$, then $k > (h-i)$, and accordingly, the node can only connect to $(h-i)$ levels of descendants. Hence, such node will have $\sum_{j=1}^{h-i} b^j = \frac{b^{h-i+1}-b}{b-1}$ children. Combining both cases, a node at level i in the complete tree has $\frac{b^{l+1}-b}{b-1}$ children, where $l = \min(k, h-i)$. Since each node is included in the sample independently of other nodes with probability p, the expected number of children in the sampled tree is $p\frac{b^{l+1}-b}{b-1}$. □

Now using Lemma 1, we can derive the expected number of isolated nodes in a sampled k-tree:

THEOREM 3. *In a sampled k-tree $\Gamma(p,b,h,k)$ the expected number of isolated nodes is equal to:*

$$\sum_{i=0}^{h} b^i p(1-p)^{l+\frac{b^{c+1}-b}{b-1}}$$

where $l = \min\{i,k\}$ and $c = \min\{h-i,k\}$.

PROOF. Let Z_i be the random variable representing the number of nodes at level i and W_i be the indicator random variable for any node at level i, equal to 1 if all of the node's parents and children are not observed and 0 otherwise. The number of isolated nodes is then $\sum_{i=0}^{h} Z_i \cdot W_i$ and, by linearity of expectation, $E[\sum_{i=0}^{h} Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i \cdot W_i]$.

For a node at level i, the probability that all of its parents are excluded from the sample is $(1-p)^l$ where $l = \min\{i,k\}$. On the other hand, for the same node the probability that all its children are excluded from the sample is given by $(1-p)^{\frac{b^{c+1}-b}{b-1}}$ where $c = \min\{h-i,k\}$ (since a node at level i has $\frac{b^{c+1}-b}{b-1}$ children). Hence, W_i is a Bernoulli random variable with success probability $(1-p)^{l+\frac{b^{c+1}-b}{b-1}}$. On the other hand, Z_i is a binomial random variable with parameters (b^i, p). Noting that Z_i and W_i are independent (because parents and children of a node are at different levels and each node is observed independently of other nodes): $\sum_{i=0}^{h} E[Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i]E[W_i] = \sum_{i=0}^{h} b^i p(1-p)^{l+\frac{b^{c+1}-b}{b-1}}$ □

X4: Number of weakly connected components. A new weakly connected component is formed in a sampled k-tree if and only if all parents of a given node are not observed. Hence, the number of weakly connected components of a sampled k-tree is equal to the number of roots of such tree, i.e., nodes with no incoming edges.

THEOREM 4. *The expected number of connected components of a sampled k-tree $\Gamma(p,b,h,k)$ is equal to:*

$$p\frac{[(1-p)b]^{a+1}-1}{(1-p)b-1} + \begin{cases} p(1-p)^k \frac{b^{h+1}-b^a}{b-1} & \text{if} \quad h > k \\ 0 & \text{if} \quad h \le k \end{cases}$$

where $a = \min(\{k,h\})$.

PROOF. Let Z_i be the random variable representing the number of nodes at level i and W_i be the indicator random variable for any node at level i, equal to 1 if all of the node's parents are not observed and 0 otherwise. The number of weakly connected components is then $\sum_{i=0}^{h} Z_i \cdot W_i$. By linearity of expectation, $E[\sum_{i=0}^{h} Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i \cdot W_i]$. Furthermore, since Z_i is independent of W_i (because parents of a given node are not among the nodes at the current level and each node is observed independently of other nodes), $\sum_{i=0}^{h} E[Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i]E[W_i]$. Now we know that Z_i is a binomial random variable with parameters (b^i, p) and W_i is a Bernoulli random variable with success probability $(1-p)^{\min(\{k,i\})}$. Hence, the number of weakly connected components is in expectation $\sum_{i=0}^{h} pb^i(1-p)^{\min(\{k,i\})}$. Simplifying this expression, we obtain:

$$p\frac{[(1-p)b]^{a+1}-1}{(1-p)b-1} + \begin{cases} p(1-p)^k \frac{b^{h+1}-b^a}{b-1} & \text{if} \quad h > k \\ 0 & \text{if} \quad h \le k \end{cases}$$

where $a = \min(\{k,h\})$ □

X5: Out-degree of a non-leaf node. The expected out-degree of a non-leaf node is equal to:

$$\frac{\text{number of edges}}{\text{number of nodes} - \text{number of leaves}} \tag{1}$$

By noticing that a node is a leaf if it has no children we derive the number of leaves in a sampled k-tree:

THEOREM 5. *In a sampled k-tree $\Gamma(p,b,h,k)$ the expected number of leaves is equal to:*

$$\sum_{i=0}^{h} b^i p(1-p)^{\frac{b^{c+1}-b}{b-1}}$$

where $c = \min\{h-i,k\}$.

PROOF. Let Z_i be the random variable representing the number of nodes at level i and W_i be the indicator random variable for any node at level i, equal to 1 if all of the node's children are not observed and 0 otherwise. The number of leaves is then $\sum_{i=0}^{h} Z_i \cdot W_i$ and, by linearity of expectation, $E[\sum_{i=0}^{h} Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i \cdot W_i]$. The number of children a node at level i has is in the general form: $\frac{b^{c+1}-b}{b-1}$ where $c = \min\{h-i,k\}$. Hence, W_i is a Bernoulli random variable with success probability $(1-p)^{\frac{b^{c+1}-b}{b-1}}$. On the other hand, Z_i is a binomial random variable with parameters (b^i, p). Since Z_i and W_i are independent (because children of a node are not among the nodes at the current level and each node

is observed independently of other nodes), we have: $\sum_{i=0}^{h} E[Z_i \cdot W_i] = \sum_{i=0}^{h} E[Z_i]E[W_i] = \sum_{i=0}^{h} b^i p(1-p)^{\frac{b^{c+1}-b}{b-1}}$ □

Now, using Theorems 1, 2, 5, and assuming independence between the number of nodes, the number of edges, and the number of leaves, we find an approximation for the out-degree of non-leaves. The approximation for an arbitrary k follows from expression (1) above. However, in the theorem that follows, we consider the case of $k = 1$ in particular, because it yields an expression for b that does not depend on h, as further discussed in Section 4.

THEOREM 6. *The expected out-degree of a non-leaf node in a sampled k-tree $\Gamma(p, b, h, k)$ for $k = 1$ is approximately:*

$$\frac{pb}{1-(1-p)^b}$$

PROOF. Assuming independence between the number of nodes, the number of edges, and the number of leaves, using Theorems 1, 2, 5, the expected out-degree of non-leaves is equal to:

$$\frac{p^2 \frac{b^{h+1}-b}{b-1}}{p\frac{b^{h+1}-1}{b-1} - (p(1-p)^b \frac{b^h-1}{b-1} + pb^h)} = \frac{pb}{1-(1-p)^b}$$

□

X6: Average node degree. Assuming independence between the number of nodes and the number of edges, we can derive an approximation for the average node degree:

THEOREM 7. *Average node degree in a sampled k-tree $\Gamma(p, b, h, k)$ with $h \gg k$, is approximately pk.*

PROOF. The average degree of a node in a k-tree can be approximated by making independence assumption between the number of nodes and the number of edges. Then using Theorem 1 and 2, we get:

$$\frac{\frac{p^2}{b-1}\left(\frac{b(1-b^k)}{b-1} + kb^{h+1}\right)}{p\frac{b^{h+1}-1}{b-1}} = \frac{p(\frac{b(1-b^k)}{b-1} + kb^{h+1})}{b^{h+1}-1}$$

If $h \gg k$, the above expression approaches pk. □

When $k = 1$ and $h \gg 1$, the theorem above shows that the average node degree is proportional to p.

Although expressions in both Theorems 6 and 7 give approximate results, we have experimentally observed that both expressions are accurate in practice.

4. MODEL ESTIMATION

Recall from Figure 2 that we observe a sample $\mathcal{C}'$ of the target cascade $\mathcal{C}$. We aim to estimate parameters of the sampled k-tree $\Gamma(p, b, h, k)$, such that its properties $\mathbf{X'_M}$ closely resemble the properties $\mathbf{X'}$ of the sampled cascade $\mathcal{C}'$. The premise is that if $\mathbf{X'_M}$ matches $\mathbf{X'}$, then $\mathbf{X_M}$ will match $\mathbf{X}$. Here we show how we estimate k-tree parameters from $\mathbf{X'}$ using the expressions for X1–X6 (i.e., $\mathbf{X'_M}$) we derived in Section 3. We estimate model parameters in two steps. We first obtain p and then obtain b, h, and k.

Obtaining p. For influence cascades, we use property X6, and specifically Theorem 7, to obtain an estimate for p, $\hat{p}$. Recall that for $k = 1$ (which is always the case for influence cascades), average node degree is equal to the fraction of observed nodes p. Accordingly, $\hat{p}$ equals to the average node degree measured on $\mathcal{C}'$.

For network cascades, on the other hand, we cannot solve analytically for p. Thus, we obtain $\hat{p}$ by other means. If the sample ratio σ used to obtain $\mathcal{C}'$ is known, $\hat{p} = \sigma$. If σ is unknown, we estimate σ and set $\hat{p}$ to the estimated σ. For example, if we have multiple cascades $\mathcal{C}'$, all of which were obtained with the same σ, one can estimate σ as the fraction of cascades where the root node is observed (granted that we can identify the root of each cascade).

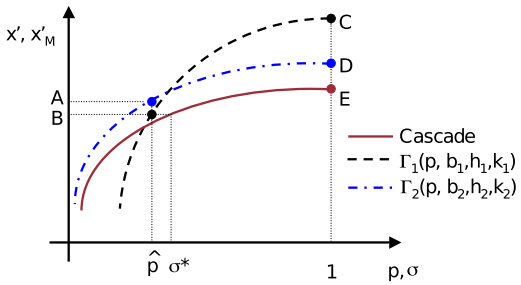


Figure 4: Fitting a k-tree model. Two alternative k-tree models with respect to the observed cascade property.

Obtaining b, h and k. Setting p to $\hat{p}$ from the previous step and equating analytical expressions for X1–X4 to the measured X1–X4 on $\mathcal{C}'$ yields a set of equations with 3 unknowns: b, h, k. For example, if m' is the number of observed nodes (property X1), then equating the expression for m from Theorem 1 yields the following equation: $\hat{p}\frac{b^{h+1}-1}{b-1} = m'$.

Since the derived equations are non-linear, we cannot directly solve this system. Instead, we solve for the unknowns by finding the set of parameters with the minimum sum of the squares of the errors made in solving every single equation. Specifically, if x' is the measured value of one property on $\mathcal{C}'$ and x'_M is the corresponding value predicted by the model $\Gamma(\hat{p}, b, h, k)$, then the squared error for that value is $(x' - x'_M)^2$.

We have experimentally observed that minimizing the errors in this fashion gives poor results. To explain why, consider Figure 4. The solid curve corresponds to the value $x'(\sigma)$ of some property x, e.g., the number of nodes in the cascade, as a function of the sample ratio σ. Of course, in reality we do not see this curve; we only see the value $x'(\sigma^*)$ at the sample ratio σ^* used to obtain $\mathcal{C}'$ (and in many cases we do not even know the value of σ^*). Our goal is to estimate the value of $x'(\sigma)$ at $\sigma = 1$. The two dashed lines illustrate the value $x'_M(p)$ of the same property x for two k-tree models Γ_1 and Γ_2. Even though at $p = \hat{p}$ model Γ_1 fits better than Γ_2 (point B is closer to the solid line than A), model Γ_2 may be preferred. At $\sigma = 1$ point D of Γ_2 is closer to E than C of Γ_1.

In order to prefer models like Γ_2 over Γ_1, we should look for models that match the $x'(\sigma)$ curve better – more than just at the point $\sigma = \sigma^*$. But how can we do this fitting when we do not know the shape of $x'(\sigma)$? We can discover other points on the $x'(\sigma)$ curve by further subsampling the sampled cascade $\mathcal{C}'$.

Say we re-sample the cascade $\mathcal{C}'$ with rate α ($0 < \alpha \leq 1$) and evaluate the properties. The effect is the same as if we had sampled the original cascade $\mathcal{C}$ with sample rate $\alpha \cdot \sigma^*$. Thus by using multiple values of α we obtain multiple points along the $x'(\sigma)$ curve. Each α yields a new error term of the form: $(x'(\alpha\sigma^*) - x'_M)^2$, where x'_M is the value predicted by a model when $p = \alpha \cdot \hat{p}$. Minimizing the sum of the error terms for all α values, we fit the model not only at a single point σ^*, but along the whole interval $(0, \sigma^*)$. We found it best to generate several samples at the same α, and then average the measured x' values.

Figure 5 summarizes the parameter estimation procedure:

1. Subsample observed cascade $\mathcal{C}'$ for multiple values of $\alpha \in (0, 1]$. For each α, generate multiple subsamples of $\mathcal{C}'$ and average measured properties X1 through X4.
2. For each α (and for $\alpha = 1$), generate an error term: the squared difference between the measured (averaged) value

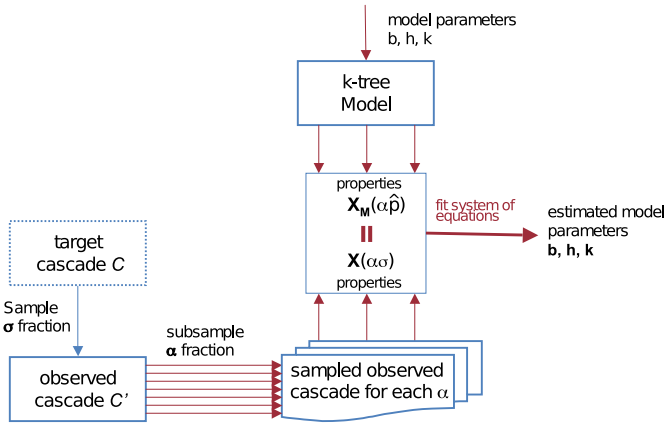


Figure 5: Parameter estimation of k-tree model. We subsample the observed cascade to obtain more accurate parameters.

and the value predicted by the k-tree model (function of $\alpha\hat{p}$, b, h, and k).

3. Apply the least squares method (we use grid-based search) to find parameters $\hat{b}$, $\hat{k}$, $\hat{h}$ that minimize the sum of the errors.

Influence cascades. In-degree of non-roots in influence cascades is always 1. So when estimating parameters for influence cascades, k is explicitly set to 1 and we only have to solve for b and h.

Furthermore, we found that better k-tree models can be found by first solving for b using X5. Specifically, we measure the out-degree of non-leaves on $\mathcal{C}'$, say it is x', and equate it to the expression from Theorem 6 to obtain: $\hat{p}b/(1-(1-\hat{p})^b) = x'$. Then we solve this equation for b numerically using bisection. Having found $\hat{b}$ estimate this way, we then estimate h using X1–X4 and subsampling as described above (X1 alone would also suffice).

Integer-valued vs. real-valued parameters. Our k-tree model assumes integer values for all three parameters. However, in real cascades nodes have varying branching factor b and in-degree k, and different leaf nodes are at different heights. Hence, we allow real valued parameters for k-trees.

Real-valued parameters have natural interpretation in our k-tree model. Real-valued b can be interpreted as an average number of direct children (not counting children attached via spurious edges), e.g., if b is 2.5, half of the nodes have 2 children and half of the nodes have 3 children. Similarly, real-valued k is interpreted as an average in-degree of non-roots, e.g. if k is 2.5, half of the nodes have connections from 2 closest ancestors while half have connections from 3 closest ancestors. Finally, modulo of h can be interpreted as the fraction of nodes with children at level $\lfloor h \rfloor$, e.g. if h is 3.5, half of the nodes at level 3 have children, while half do not.

However, the expressions for X1–X5 do not allow non-integer values for h and k (because bounds of summations need to be integer-valued). To address this, we linearly interpolate the function value between two integer values. For example, if $y_0 = f(x_0)$ for integer x_0 and $y_1 = f(x_0+1)$, then the value of $f(x)$ for $x \in [x_0, x_0+1]$ is $y_0 + (x-x_0)(y_1-y_0)$. In our case we linearly interpolate between $f(h,k)$ and $f(h+1,k+1)$ in 2 dimensions.

5. EXPERIMENTS

In this section, we evaluate our cascade model and our method for correcting for missing data in the cascades. We first evaluate whether k-tree cascades' properties are affected by the missing data in the same way as the properties of the target cascades. In other words, is there a k-tree for each target cascade such that the properties of the target cascade are similar to the properties of the k-tree at each sample ratio? Next, we evaluate the soundness of the method. Specifically, do the properties $\mathbf{X}_\mathrm{M}$ of the complete k-tree parameterized based on $\mathcal{C}'$ match the properties $\mathbf{X}$ of the complete target cascade $\mathcal{C}$? Finally, we study the parameters of the model itself. b, k, and h can themselves be viewed as properties of the original cascade. Hence, we look at how well k-tree parameters match the corresponding properties of the target cascade. If k-tree parameters match the corresponding cascade properties then each parameter indeed has an intuitive meaning.

5.1 Experimental Setup

For our evaluation, we need complete target cascades. We consider two types of target cascades: (a) *synthetic cascades* generated on real and synthetic networks, (b) actual cascades obtained from Twitter and blogs which we refer to as *real cascades*. Each complete cascade, in our experiments, can actually be sampled at several ratios, not just at one ratio σ^*, as it is the case for the target cascade of Figure 2. Hence, in this section we refer to the sample ratio of the observed cascade using variable σ, and not σ^*.

Synthetic cascades. Synthetic cascades are generated using an action propagation model simulated on a given network. The model takes as input a network and action sequence size, set to 127 in our simulations, and generates an action sequence which specifies influences. We use both synthetic networks and the real network of Twitter users to simulate our action propagation model.

We use a variant of the Susceptible-Infected (SI) model [2], commonly used in studies of virus and information diffusion. Here are the steps of the simulation. First, we select at random a root node r with non-zero out-degree. Then, r is added to the initially empty list of infected nodes I and all of its outgoing edges (r,s) are added to the initially empty FIFO queue of infected-susceptible node pairs S. After that, we repeat the following steps until 127 elements of the action sequence A are produced:

1. Remove the first pair (r,s) from the queue S.
2. With probability β, output an action sequence element (r,s). Then add s to I and add all edges (s,u) where $u \notin I$ to S. Otherwise (i.e., with the remaining probability $1-\beta$), push (r,s) back into S.

Our action propagation model requires a network as input. We used two types of networks: synthetic networks and real social network of "who follows whom" of Twitter.

Synthetic networks were generated using three network models: Erdős-Rényi random graph, Scale Free random graph and Forest Fire model [16]. We do not present results for all of them in this paper, but complete results for all the models can be found in the extended version of the paper [23]. All networks were generated with 10^6 nodes. Erdős-Rényi graph was generated with an average degree of 10, Forest fire network was generated with parameters $p_f = 0.36$ and $p_b = 0.315$ (yielding average degree of 10). And Scale Free graph was generated with power law degree distribution exponent $\alpha = 2.0$ (roughly corresponding to the power law degree exponent of the Twitter network).

The Twitter "who follows whom" network was collected via the Twitter API in a breadth-first manner from June through December 2009 with the set of seed user IDs taken from the public stream of tweets (Twitter status updates) monitored in that period. In other words, for every user u for which we observed a tweet, we collected friends of u followed by their friends of friends, etc. in a breadth-first manner. The network we obtained has 71,804,410 nodes and 2,040,072,198 directed edges (average degree of 28.4), where each

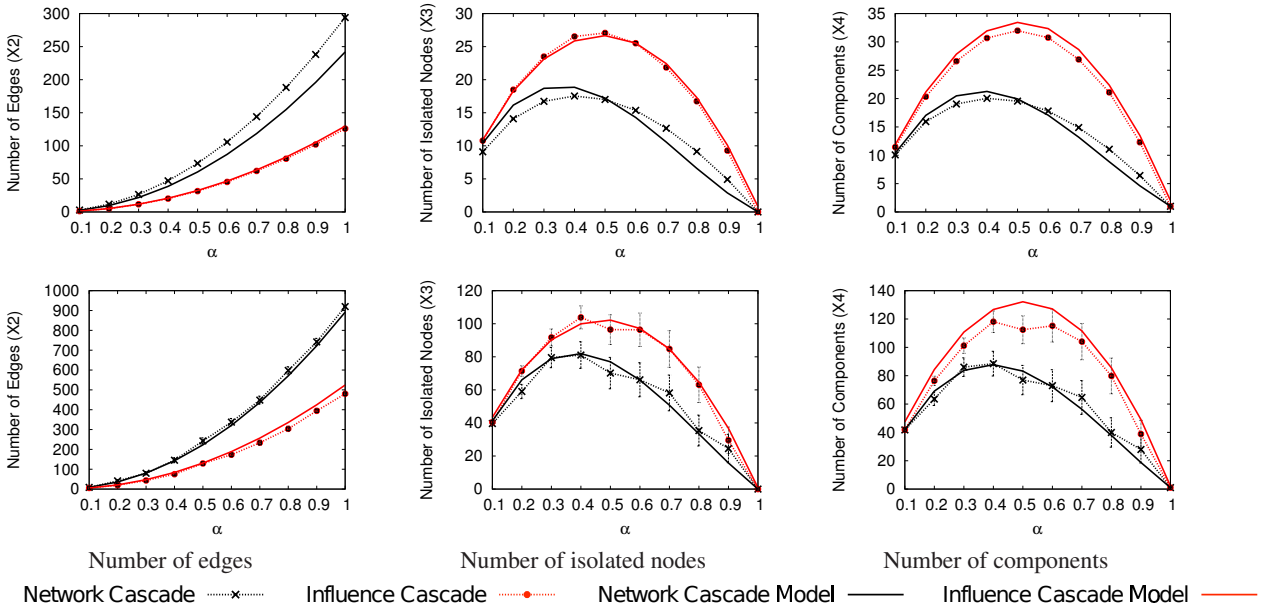


Figure 6: X2-X4 properties on estimated k-tree and observed cascades. First row: synthetic cascades on Twitter network. Second row: a single Twitter retweet cascade. Error bars correspond to 95% confidence interval. Note agreement between the properties of the target cascade (solid line) and the properties of the k-tree cascade (dashed line) as we vary the fraction of missing data.

edge corresponds to the "follows" relationship among users (if A follows B, A receives B's tweets).

We also consider real cascades that are constructed from action sequences extracted from traces of human activity. We use Twitter retweets as natural action sequences that when combined with the Twitter network form network cascades. In addition, we also use the action sequences of link creation between blog posts that naturally form influence cascades.

Retweet Cascades. Tweets are Twitter status update messages and retweets are re-postings of the previously posted tweets. We focus on how a given URL x propagates through the Twitter social network by people reposting (i.e., forwarding) the original Tweet.

We took a *complete* set of tweets, collected by Topsy, for the most popular URLs posted on Twitter between June and December 2009. From the set of tweets with URLs, we then extracted retweets. If user u posts a tweet with URL x, any tweet of the form "RT @u t", where t contains URL x, is a retweet of x. For example, suppose user A posts a tweet with URL x, then user B who follows A posts "RT @A x" and another user C who follows B (but may not follow A) posts: "RT @A x". This sequence of tweets forms an action sequence $\langle(\perp, A), (\emptyset, B), (\emptyset, C)\rangle$. This action sequence, combined with the network of who follows whom, can then be used to construct a network cascade (if a node retweets more than once, we consider only the first retweet).

Note that there is no way to tell which node influenced which other node from retweets. Using the same example, if another node D retweets A's x and D follows both B and C (but does not follow A), then both B and C could have influenced D. Thus to obtain influence cascades from retweet network cascades, we select a single incoming edge for each node giving credit to the last neighbor to retweet. In our example, if C retweeted after B, we say (C, D).

Although we had to drop some of the tweets due to changed and deleted usernames, our final cascades were nearly 95% complete with the experiments performed on their largest connected components. We considered only cascades of more than 100 nodes with the total of 250 such cascades.

Blog Cascades. Blog posts and links in them to other blog posts provide action sequences with explicit influence relations. Specifically, each post with no outgoing links starts an action sequence. Suppose blog A makes a post a and blog B links to post a in one of its posts b, then we can infer action sequence (A, B). If blog C then makes a post c linking to b, our action sequence becomes $\langle(\perp, A), (A, B), (B, C)\rangle$. Accordingly, we can then construct an influence cascade. If C has links to say both x and y, we arbitrarily pick one of the links. In our dataset, blog posts linking to more than one post in the same cascade were extremely rare (less than 0.01%), validating our model of influence cascades as trees.

For our experiments, we extracted influence cascades from the set of blog posts collected by Spinn3r between August and November 2008. This data set includes essentially a complete snapshot of the English blogosphere. We considered only cascades of 100 and more nodes with the total of 100 such cascades. We did not consider network cascades in the context of blogs because there is no explicit blog network (although implicitly created blog networks have been studied before [17]).

5.2 Soundness of the k-tree Model

One of the assumptions underlying our methodology is that sampling (i.e., missing data) has the same effect on k-trees as on target cascades. We test this assumption to validate our method and demonstrate that k-tree is a useful model for real cascades.

In this experiment we work with complete target cascades so $\sigma = 1$ (i.e., $\mathcal{C}' = \mathcal{C}$). We sample each target cascade at rates $\alpha = 0.1, 0.2, \ldots, 1.0$ and measure properties X1–X4 at each rate. Then we estimate a k-tree model $\Gamma(\hat{b}, \hat{h}, \hat{k})$ as discussed in Section 2. Estimated parameters $\hat{b}, \hat{h}, \hat{k}$ are the same for all four properties.

Figure 6 shows a grid with 6 graphs. The top row corresponds to synthetic cascades on the Twitter network and the bottom row corresponds to a retweet cascade[1]. Each column of the grid corresponds to one of the X2–X4 properties. Consider the bottom

[1]We estimate model parameters for each real cascade individually, so we are showing results for only one cascade as an example.

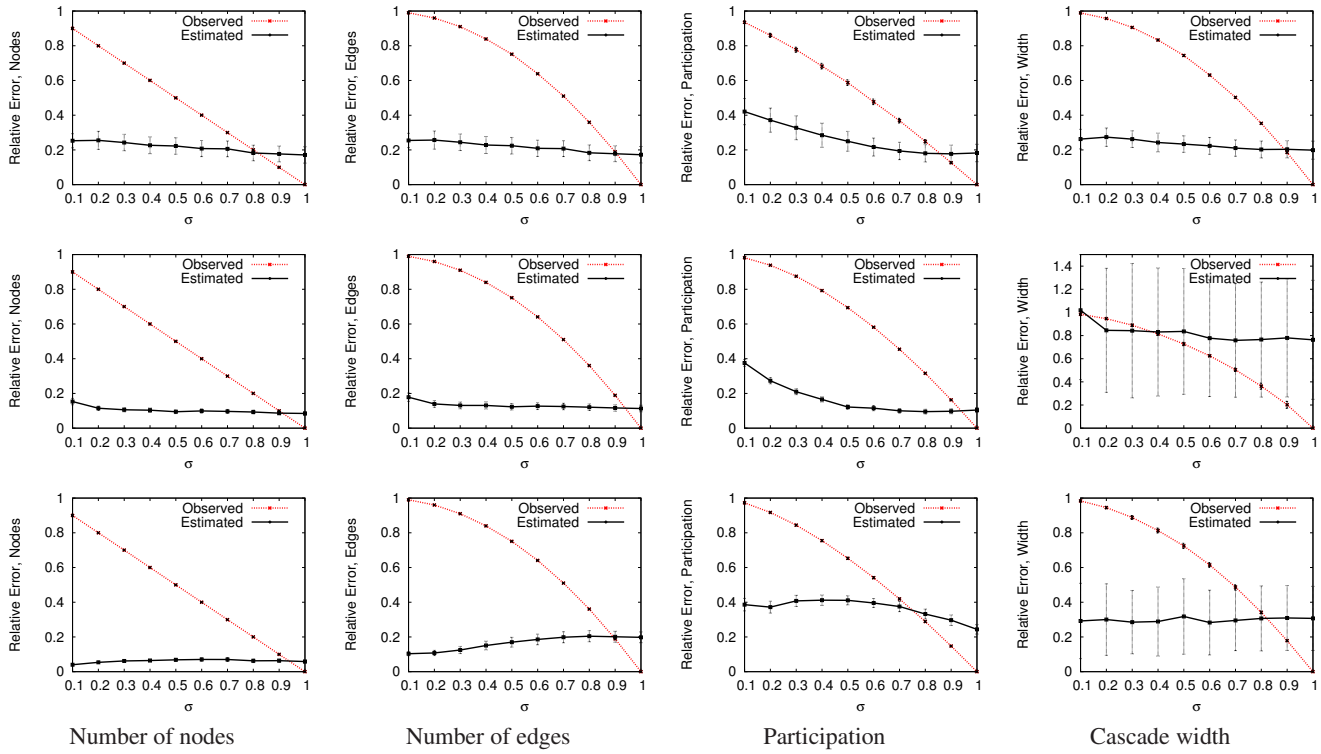


Figure 7: Observed vs. estimated properties. First row: blog influence cascades. Second row: retweet influence cascades. Third row: retweet network cascades. All errors are averaged over a set of cascades, error bars correspond to 95% confidence interval.

rightmost graph for a single retweet network cascade and its corresponding influence cascade. The dark dashed curve shows X4 (number of weakly connected components) measured on the network cascade as a function of subsample rate α. The dark solid curve shows analytically calculated X4 for the network cascade's k-tree model as a function of α. Similarly, the light dashed curve shows X4 measured on the influence cascade and the light solid curve shows analytically calculated X4 for its k-tree model. Observe that the model predicts the actual values fairly well.

Now consider the top row. Here, for the target influence and network cascades, each measured property value at each sample rate is an average across 1000 synthetic cascades simulated on the Twitter network (all of the same size). Accordingly, the k-tree models for both influence and network cascades are fitted to the average of measured values. Again, the fits are fairly close.

Note that the model curves in each row correspond to the same k-tree (estimated for either network or influence target cascade), so the curves in each graph were not fitted individually. Yet the analytical model values match closely the measured values for all the properties. Although we do not present results here, the same close fit for all properties, including X1, was observed for synthetic cascades simulated on synthetic graphs and real cascades constructed from blogs. More importantly, we performed the same experiment with cascade properties that are not explicitly fitted (e.g., size of the largest weakly connected component) and found similar close fits. Overall, we conclude that *for every target cascade there is a k-tree model such that their properties are alike at each sample rate*.

5.3 Estimating Target Cascade Properties

Next, we evaluate to what degree do the properties of the estimated k-tree model approximate the properties of the complete target cascade.

We reconstruct the following cascade properties: (1) number of nodes, (2) number of edges, (3) width, which is defined as the maximum number of nodes at any depth/level [5], and (4) participation, which is defined as the number of non-leaf nodes [5].

Figure 7 shows a grid of 12 plots. The top row corresponds to blog cascades, the middle row corresponds to retweet influence cascades, and the bottom row corresponds to retweet network cascades. Each column corresponds to one of the four properties. The results are averaged across 250 retweet and 100 blog cascades (the results for synthetic cascades are similar to those presented here).

For example, consider the top leftmost graph for the number of nodes measured on blog cascades. Say x is the number of nodes in $\mathcal{C}$, $x'(\sigma)$ is the number of nodes in $\mathcal{C}'$ obtained with sample ratio σ and x_M is the number of nodes in the k-tree model estimated from $\mathcal{C}'$. The relative error of the k-tree model estimate at sample ratio σ is then $\hat{e} = \frac{|x_M - x|}{x}$, shown by a dark curve in the plots (k-tree model may differ at each σ). Similarly, the relative error of the observed value at sample ratio σ is $e' = \frac{|x'(\sigma) - x|}{x}$, shown by a light curve in the plots. As expected, the light curve is a straight line because the observed cascade size, and its relative error, is linear with σ.

In all of the plots of Figure 7, except for the width on retweet influence cascades, the error of the estimated properties is better than the error of the observed properties for almost all σ values. In general, with 70% or less of the target cascade $\mathcal{C}$, our method provides a significantly better estimate of $\mathcal{C}$'s properties than what is observed on $\mathcal{C}'$. However, for $\sigma > 0.9$, our method does worse than working with $\mathcal{C}'$ directly and ignoring the missing data. This suggests that if estimated sample ratio $\hat{p}$ is high, one is better off measuring properties on $\mathcal{C}'$ directly. But, of course, most properties are not perturbed by missing data at such such high σ values and one would not bother to correct for missing data in such case.

	Network Cascade		Influence Cascade	
	estimated error ($\hat{e}$)	observed error (e')	estimated error ($\hat{e}$)	observed error (e')
p	–	–	0.02	–
b	0.03	0.29	0.03	0.32
k	0.14	0.21	–	–
h	0.00	0.39	0.00	0.46

Table 2: Relative errors for estimated and observed parameters averaged over synthetic cascades on Twitter network, $\sigma^* = 0.5$

	Spurious Edges		No Spurious Edges	
	estimated error ($\hat{e}$)	observed error (e')	estimated error ($\hat{e}$)	observed error (e')
p	–	–	0.02	–
b	0.03	0.86	0.02	0.31
k	0.02	0.43	–	–
h	0.05	0.66	0.10	0.43

Table 3: Relative errors for estimated and observed parameters for k-trees with 127 nodes, $b \sim$ Normal$(2,1)$, $k = 3.5$, $\sigma^* = 0.5$

Distortions due to missing data become a bigger issue at lower σ values and this is where our method is most effective and significantly outperforms measurements made on $\mathcal{C}'$. 20-30% relative error is especially encouraging for such low values of σ as 0.1.

As seen in the rightmost plot of the second row of Figure 7, our method performs poorly on the width of retweet influence cascades. We found that while blog influence cascades are mostly shallow balanced trees (star-shaped), retweet influence cascades resemble more imbalanced trees, possibly because they were artificially generated from the network cascades. This is the reason we believe we are unable to estimate well the width of retweet influence cascades.

Finally, observe that while the number of nodes and edges are explicitly fitted during parameter estimation, participation and width are not fitted. Yet, our model predicts these properties fairly well.

5.4 Estimated vs. Observed Parameters

Parameters of the k-tree model can themselves be viewed as cascade properties. For example, parameter k naturally maps to the average node in-degree in a cascade. We next evaluate how well the model parameters match the corresponding cascade properties.

As described in Section 3, k accounts for the spurious edges ($k-1$ spurious edges per node), while b and h are branching factor and height of the tree, respectively, without the spurious edges. A network cascade is essentially an influence cascade with spurious edges. Accordingly, if a network cascade has a corresponding influence cascade (which is always the case, given our experimental settings), p, b and h values must be the same for both cascades. Because k is trivially 1 for influence cascades, the estimated parameter k is specific to a network cascade.

Now let's define how model parameters map to cascade properties. p corresponds to the sample ratio σ of $\mathcal{C}'$, so we say the *true* value of p is σ. For b, h, and k, the true value of each will be its corresponding property value measured on $\mathcal{C}$. The true value of b is the average out-degree of non-leaves in the influence cascade. The true value of h is the average weighted depth over all leaves of the influence cascade. The weight of each leaf is the number of its descendants at the max level of the tree had the tree been balanced (consistent with real-valued h in Section 4). Finally, the true value of k is the average in-degree of non-roots. Recall that we only consider k for network cascades.

To have a baseline when comparing estimated parameters to the true ones, we also measure properties corresponding to b, h, and k on $\mathcal{C}'$. We will refer to these measured properties as to *observed* parameters. Observed parameters are defined similarly to the true ones but unlike the true parameters these are measured on $\mathcal{C}'$ as opposed to $\mathcal{C}$ [2]. For example, observed b is defined as the average out-degree of non-leaves measured on $\mathcal{C}'$.

In this experiment, the goal is to compare estimated model parameters to the true parameters. As a baseline, we also compare the observed parameters to the true parameters. To make comparison direct, as in the previous experiment, we use relative errors $\hat{e}$ and e'. For example, if b^* is the true value of branching factor, $\hat{b}$ is its estimate, and $b'(\sigma)$ is the observed value at sample ratio σ, then $\hat{e} = \frac{|\hat{b}-b^*|}{b^*}$ and $e' = \frac{|b'(\sigma)-b^*|}{b^*}$.

Performance on synthetic cascades. We generated synthetic cascades on synthetic networks and Twitter network. Due to space constraints we show results here only for Twitter network; the results for other networks are similar to the ones we present here. Synthetic cascades were generated with 127 nodes each with a total of 1000 cascades simulated on each network.

Table 2 shows results averaged over all simulations on Twitter network with sampled cascades $\mathcal{C}'$ at $\sigma = 0.5$. Each row corresponds to one of the four parameters. There are two sets of columns, one for the network cascades and one for the influence cascades. Each set has two columns: the left one corresponds to the estimated error $\hat{e}$ (with respect to one of the four parameters), the right one corresponds to the observed error e'. Note that we are not showing the observed error for p because this parameter cannot be directly measured from observed data. Also we cannot analytically estimate p for network cascades, so we omit values for p's estimated error for network cascades.

Observe that the errors of the estimated parameters are very low: almost all below 5%. Contrast them with 20-40% errors for observed parameters. Finally, note how accurately we are able to infer p for influence cascades. This demonstrates the effectiveness of X6 as an estimate of σ.

Recall that both influence and network cascades, in our setting, are constructed from the same action sequence. Accordingly, b and h are the same for the network and influence cascades, as discussed above. Interestingly enough, although model parameters for network and influence cascades are estimated independently, the estimated errors for b and h are low in both cases. Hence, similar b and h values are predicted for both influence and network cascades. This is yet another strength of our method: we are able to detect close similarity between the equivalent network and influence cascades.

Robustness of parameter estimation. Finally, we test the robustness of parameter estimation. By generating and sampling a k-tree cascade with constant integer b, h, k parameters and σ sample ratio, we experimentally verified that we can match all parameters precisely ($\hat{e} = 0$ for all parameters). So we added variance to parameter b and used real-valued k. Specifically, we generated cascades with 127 nodes, $b \sim$ Gaussian$(2,1)$ and $k = 3.5$ (since b is stochastic, the actual height varied). All cascades $\mathcal{C}$ were generated 1000 times and sampled to obtain $\mathcal{C}'$ with $\sigma = 0.5$.

Table 3 shows the results. The table is similar to Table 2 with the two sets of columns referring to cascades with and without the spurious edges instead of network and influence cascades, respectively. Although we added variance to the branching factor and used a real-valued k, the estimated parameters have a very small error with respect to the true values. This result demonstrates the robustness of our parameter estimation.

[2]We cannot assume to have access to both influence and network cascades, so the parameters are measured on whatever $\mathcal{C}'$ we are given (either network or influence).

6. DISCUSSION AND CONCLUSION

In this paper, we addressed the problem of estimating properties of a target cascade $\mathcal{C}$, given only its fraction $\mathcal{C}'$ obtained by uniform sampling of $\mathcal{C}$ nodes. This is the first attempt to our knowledge to analytically study the effect of missing data on cascade properties and, most importantly, the first attempt to correct for missing data in cascades. In summary, our contributions are as follows:

- Proposed an analytical k-tree model of cascades and rigorously derived a number of their important properties.
- Experimentally showed that the k-tree model is an effective proxy to study the effect of missing data on the observed properties of $\mathcal{C}'$.
- Proposed a method that, given a cascade model (k-tree, in our case), estimates properties of $\mathcal{C}$ given $\mathcal{C}'$.
- Experimentally demonstrated that the estimated properties of $\mathcal{C}$ using our method are significantly more accurate than the observed properties on $\mathcal{C}'$ (effectively correcting for the property distortions due to missing data in $\mathcal{C}'$).
- Experimentally showed that the k-tree model parameters have an intuitive meaning: they roughly correspond to the properties of the target cascade $\mathcal{C}$.

Our methodology for estimating cascade properties in the face of missing data is a practical necessity. For instance, Twitter provides public access to only up to 10% of its stream of tweets, whereas Facebook users are becoming more concerned about privacy of their data. Current algorithms and methods for finding influential nodes, designing viral marketing campaigns, or studying information diffusion processes assume access to complete cascade data. These algorithms fail given the distorted cascade properties caused by incomplete data. The method we propose can address these issues. For example, using our method one could estimate how many people influenced their followers to retweet a given tweet (equivalent to node participation) even when 90% of the tweets are missing. Furthermore, one could even estimate how much data is missing in the first place. For example, if one was studying cascades formed by links between blogs posts, one could use our techniques to estimate what fraction of posts are missing from the data set.

There are a number of future directions for this work. The k-tree model for cascades, although simple and thus relatively easy to analyze, may not work for all types of cascades. As we have seen for retweet influence cascades, cascade trees which are severely imbalanced, may create challenges for our model. Hence, we may need more sophisticated models of cascades, possibly stochastic in nature based on Galton-Watson trees [26]. Given a different model, however, our method to correct for missing data could still be applied. Finally, in this paper we worked in the regime of sampled action sequences but complete knowledge of the underlying network. One interesting venue for future work could be studying the effect of missing data in both action sequences and the network.

Acknowledgments

We thank Topsy and Spinn3r for providing us with the data that facilitated the research. Research was in-part supported by NSF CNS-1010921, NSF IIS-1016909, AFRL FA8650-10-C-7058.

7. REFERENCES

[1] D. Achlioptas, A. Clauset, D. Kempe, and C. Moore. On the bias of traceroute sampling: Or, power-law degree distributions in regular graphs. JACM, vol 56: 1–28, 2009.

[2] N. Bailey. *The Mathematical Theory of Infectious Diseases*. Griffin, London, 1975.

[3] S. Bikhchandani, D. Hirshleifer, and I. Welch. A theory of fads, fashion, custom, and cultural change in informational cascades. *J. of Polit. Econ.*, 100(5):992–1026, 1992.

[4] M. Cha, A. Mislove, and K. P. Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *WWW '09*, pages 721–730.

[5] M. de Choudhury, Y.-R. Lin, H. Sundaram, K. S. Candan, L. Xie, and A. Kelliher. How does the data sampling strategy impact the discovery of information diffusion in social media? In *ICWSM '10*.

[6] G. Ellison. Learning, local interaction, and coordination. *Econometrica*, 61(5):1047–71, 1993.

[7] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 3(12):211–223, 2001.

[8] M. S. Granovetter. Threshold models of collective behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[9] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW '04*.

[10] D. Kempe, J. M. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD '03*, pages 137–146.

[11] G. Kossinets. Effects of missing data in social networks. *Social Networks*, 28:247–268, 2006.

[12] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a Social Network or a News Media? In *WWW'10*.

[13] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling biases in ip topology measurements. In *INFOCOM*, 2003.

[14] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM TWEB*, 1(1):2, 2007.

[15] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *KDD '06*, pages 631–636.

[16] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM TKDD*, 2007.

[17] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *SDM '07: SIAM Conference on Data Mining*.

[18] D. Liben-Nowell and J. Kleinberg. Tracing information flow on a global scale using Internet chain-letter data. *PNAS*, 105(12):4633–4638, 2008.

[19] A. Maiya and T. Berger-Wolf. Sampling community structure. In *WWW '10*.

[20] M. E. J. Newman. The spread of epidemic disease on networks. *Phys. Rev. E*, 66:016128, 2002.

[21] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD '02*, pages 61–70.

[22] E. M. Rogers. *Diffusion of Innovations*. Free Press, New York, fourth edition, 1995.

[23] E. Sadikov, M. Medina, J. Leskovec, and H. Garcia-Molina. Correcting for Missing Data in Information Cascades. InfoLab, Stanford University, Tech. Report, 2010 `http://ilpubs.stanford.edu:8090/980/`

[24] D. Stutzbach, R. Rejaie, N. G. Duffield, S. Sen, and W. Willinger. Sampling techniques for large, dynamic graphs. In *INFOCOM*, 2006.

[25] E. Sun, I. Rosenn, C. Marlow, and T. Lento. Gesundheit! modeling contagion through facebook news feed. In *ICWSM '09*.

[26] H. W. Watson and F. Galton. On the probability of extinction of families. *Journal of the Anthropological Institute of Great Britain and Ireland*, (4):138–144, 1875.