Spheres of Influence for More Effective Viral Marketing

Yasir Mehmood
Pompeu Fabra University
Barcelona, Spain
yasir.mehmood@gmail.com

Francesco Bonchi
ISI Foundation
Turin, Italy
francesco.bonchi@isi.it

David García-Soriano
Eurecat
Barcelona, Spain
david.garcia@eurecat.org

ABSTRACT

What is the set of nodes of a social network that, under a probabilistic contagion model, would get infected if a given node s gets infected? We call this set the *sphere of influence* of s. Due to the stochastic nature of the contagion model we need to define a notion of "expected" or "typical" cascade: this is a set of nodes which is the closest to all the possible cascades starting from s.

We thus formalize the *Typical Cascade* problem which requires, for a given source node s, to find the set of nodes minimizing the expected Jaccard distance to all the possible cascades from s. The expected cost of a typical cascade also provides us a measure of the *stability* of cascade propagation, i.e., how much random cascades from a source node s deviate from the "typical" cascade. In this sense source nodes with lower expected costs are more reliable.

We show that, while computing the quality of a candidate solution is #**P**-hard, a method based on (1) sampling random cascades and (2) computing their Jaccard Median, can obtain a multiplicative approximation with just $O(1)$ samples. We then devise an index that allows to efficiently compute the sphere of influence for any node in the network.

Finally, we propose to approach the influence maximization problem as an instance of set cover on the spheres of influence. Through exhaustive evaluation using real-world networks and different methods of assigning the influence probability to each edge, we show that our approach *outperforms in quality* the theoretically optimal greedy algorithm.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: [Database Applications-Data Mining]; G.2.2 [**Discrete Mathematics**]: [Graph Theory-Graph Algorithms]

Keywords

uncertain graphs; typical cascade; sphere of influence; influence maximization

SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA

ISBN 978-1-4503-3531-7/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2882903.2915250

1. INTRODUCTION

The phenomenon of influence-driven cascades in social networks has received tremendous attention in the last years thanks to its applications, among which the most appealing is *viral marketing*. The idea of viral marketing is to exploit a pre-existing social network in order to increase brand awareness or to achieve other marketing objectives (such as product sales) through self-replicating viral processes, analogous to the spread of viruses. More concretely, the idea is to target a few "influentials", in the hope that, through word-of-mouth mechanism, they will be able to spread the marketing message to a large portion of the network, as it was a viral contagion.

This notion was formalized by Kempe et al. [24] in the *Influence Maximization* problem, i.e., the problem of finding the set of k influential nodes (usually named "seed set") such that activating them maximizes the expected number of nodes that eventually get activated in a social network where the contagion is governed by a stochastic propagation model. This problem has received a great deal of attention by the research community in the last decade.

However, Watts [39, 40, 41] challenges what he calls *"The Influentials Hypothesis"*, i.e., the assumption that a small set of super-star users can act as sparks to start a large forest fire. Watts states that influence processes are highly unpredictable and unreliable, and relying on a small seed set simply aggravates the unpredictability. Therefore, in order to implement viral marketing in the real world, Watts suggests we should target a large seed set of ordinary individuals who might trigger their small, but more reliable, *sphere of influence*. Even if each seed manages to activate a handful of other users, the large size of the seed set makes possible to reach a critical mass that can make the campaign go viral.

Inspired by this vision, in this paper we study how to compute the sphere of influence of each node s in the network, together with a measure of stability of such sphere of influence, representing how predictable the cascades generated from s are. We then devise an approach to influence maximization based on the spheres of influence and maximum coverage, which is shown to outperform in quality the theoretically optimal method for influence maximization when the number of seeds grows.

In order to better explain our contributions, we first need to provide some preliminary background on the influence maximization problem.

Influence maximization. Kempe et al. [24] modeled viral marketing as a discrete optimization problem, named *influence maximization*, and based on the concept of propagation model: i.e., a stochastic model that governs how users influence each other and thus how contagion happens. Given a propagation model and a set of nodes $S \subseteq V$, the expected number of nodes "infected" in the viral cascade started with S is called the *expected spread* of S, denoted by $\sigma(S)$. For a given $k \in \mathbb{N}$ the influence maximization problem asks for a set $S \subseteq V$, $|S| = k$, such that $\sigma(S)$ is maximum.

The most studied propagation model is the so called *Independent Cascade* (IC) model. We are given a directed probabilistic graph $\mathcal{G} = (V, E, p)$ where each arc $(u, v) \in E$ is labeled with a contagion (or influence) probability $p_{u,v} \in (0, 1]$, representing the strength of the influence of u over v. At a given time step, each node is either active (an adopter of product) or inactive. At time 0, a set S of seeds are activated. Time unfolds deterministically in discrete steps. When a node u first becomes active, say at time t, it has one chance to influence each inactive neighbor v with probability $p_{u,v}$, independently of the history thus far. If the attempt succeeds, v becomes active at time $t + 1$.

Influence maximization is generally **NP**-hard [24]. Kempe et al., however, show that the objective function $\sigma(S)$ is *monotone*[1] and *submodular*[2]. When equipped with such properties, the simple greedy algorithm that at each iteration greedily extends the current set of seeds S with the node w providing the largest marginal gain $\sigma(S \cup \{w\}) - \sigma(S)$, gives a $(1 - 1/e)$-approximation to the optimum [30, 24].

Another source of complexity is the fact that the computation of the expected spread which is itself #**P**-hard. Therefore in the work of Kempe et al. Monte Carlo simulations are run sufficiently many times to obtain an accurate estimate of the expected spread. In particular they show that for any $\phi > 0$, there is a $k = \text{poly}(n/\phi)$ such that by using k samples, we can obtain a $(1 - 1/e - \phi)$-approximate solution for influence maximization.

Finally, it is important to note that the $(1 - 1/e)$ approximation ratio for influence maximization cannot be further improved, at least under the IC propagation model. This is due to the fact that influence maximization under the IC model encodes max-k-cover as a special case, which has been shown to be not approximable within ratio $(1 - 1/e + \epsilon)$ unless **P**= **NP** [15]. For this reason, while a very literature has been produced on the efficiency and scalability of influence maximization, understandably very little attention has been devoted to improving the quality (at least in practice, given that in theory it is not possible).

Problem studied. The problem studied in this paper is, abstractly, to compute the set of nodes that, under a probabilistic contagion model, would get infected if a given node s get infected. This can be seen as a novel type of reachability query in *uncertain* or *probabilistic* graphs [31, 23]. More in details, our data is a probabilistic directed graph $\mathcal{G} = (V, E, p)$, where $p : E \to (0, 1]$ is the contagion probability, i.e., the probability that the arc will exist, or participate, in a contagion cascade. Our query is a source node $s \in V$, and the result is a set of nodes $C \subseteq V$, which we call the *sphere of influence* of s, i.e., the set of nodes that would get infected if the node s get infected.

This type of query can find application in many contexts besides viral marketing: from epidemics (*given an ebola case, which other individuals should we quarantine?*), to corporate workflows, computer and financial networks (*given a node failure, which is the typical cascade we can expect?*).

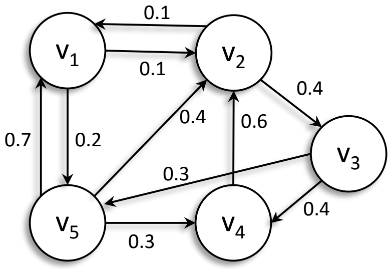


Figure 1: An example probabilistic graph.

As the contagion is a stochastic process, we need to define a way to identify a unique set of nodes C. In fact, each possible subset of V can be a possible cascade from s, each one with its own probability of materializing.

Example 1. *Consider the probabilistic graph in Figure 1 and suppose* v_5 *is our query node. The probabilities associated to the arcs define a probability distribution over the possible subsets of* $\{v_1, v_2, v_3, v_4\}$. *For instance the set* $\{v_1\}$ *is the cascade of* v_5 *with probability* $0.7 \cdot (1 - 0.4) \cdot (1 - 0.3) \cdot (1 - 0.1) = 0.2646$ *(i.e., the arc* (v_5, v_1) *succeeds transmitting the contagion, while* (v_5, v_2), (v_5, v_4), *and* (v_1, v_2) *all fail). Similarly the set* $\{v_2, v_4\}$ *is the cascade of* v_5 *with probability* $(1 - 0.7) \cdot 0.3 \cdot (1 - ((1 - 0.4) \cdot (1 - 0.6)) \cdot (1 - 0.1) \cdot (1 - 0.4) = 0.036936$ *(i.e., the arc* (v_5, v_1) *fails transmitting the contagion,* (v_5, v_4) *succeeds and at least one among* (v_5, v_2) *and* (v_4, v_2) *succeeds, finally* (v_2, v_1) *and* (v_2, v_3) *both fail).*

As a final example, the set $\{v_1, v_3, v_4\}$ *has null probability of being the cascade of* v_5 *as* v_3 *can only be infected by* v_2.

Thus, how to identify a unique set of nodes C from this probability distribution? One could think to select the most probable cascade, but this would not be a good choice as explained next. If we have $|V| = n$ nodes there are 2^n possible cascades and n is usually large. This means that we have a probability distribution over a very large discrete domain, with all the probabilities that are very small. As a consequence the most probable cascade still has a tiny probability, not much larger than many other cascades. Finally, the most probable cascade might be very different from many other equally probable cascades.

Instead in this paper we study the problem of computing set of nodes which is the closest (in expectation) to all the possible cascades of s. For our purpose we need a set-similarity measure: the Jaccard similarity is the most natural choice and it has the benefit of being a metric.

Paper contributions. The main contributions of this paper are summarized as follows:

- We formalize the *Typical Cascade* problem which requires, for a given source node s, to find the sets of nodes minimizing the expected Jaccard distance to all the possible cascades from s. Such expected cost also represents, for a given node s, a measure of the *stability* of its sphere of influence, i.e., how much a random cascades from a source node s deviate from the "typical" cascade. In this sense source nodes with lower expected cost are preferable: e.g., in the context of viral marketing they can be considered more reliable influencers.
- We show that for a given source node s, computing the expected cost for a candidate set of nodes is #**P**-hard.

[1] $\sigma(S) \leq \sigma(T)$ whenever $S \subseteq T$.

[2] $\sigma(S \cup \{w\}) - \sigma(S) \geq \sigma(T \cup \{w\}) - \sigma(T)$ whenever $S \subseteq T$.

- We then devise a solution based on sampling possible worlds and then computing the *Jaccard median* [11] of the obtained cascades.
- The next question we face is how many samples are needed in order to get a "good" approximation. We answer this question by providing theoretical bounds showing that, quite surprisingly, we can obtain a multiplicative approximation with a constant number samples, i.e., not dependent on the size of the network.
- Backed by our theoretical results, we turn our attention to the practical deployment of our algorithm and we devise an index that allows to efficiently compute the sphere of influence for any node in the network.
- Finally, we apply our framework to the influence maximization problem and propose a max-cover based solution over the spheres of influence. Trough exhaustive evaluation using real-world networks and different methods of assigning the influence probability to each arc, we show that our approach *outperforms in quality* the theoretically optimal greedy algorithm.

Our method based on spheres of influence has several interesting features that can explain its quality.

The first observation is that with our method we intuitively steer the attention of the greedy algorithm from the average size of cascades (i.e., the expected spread), to the size of the "average cascade". This gives us a more reliable approach to the influence maximization problem. In fact, as suggested by intuition, the typical cascade of a node gets larger when all the possible cascades from that node have a large common portion, or in other terms, are similar. Therefore, by picking nodes with large typical cascades, not only do we pick nodes that are influential, but we also implicitly favor influentials that are reliable. The connection between the size of the typical cascade and its cost is confirmed empirically in Section 6.3.

We also show empirically that, as the seed set size grows, at a certain point the standard influence maximization approach reaches a saturation point where it is no longer able to distinguish well among nodes to be added to the solution. Essentially, by focusing on the marginal gain w.r.t. the expected spread, the standard method finds itself choosing among many practically equivalent nodes. Instead our method, by focusing on the sets themselves, is still able to distinguish the next good candidate when the standard influence maximization has reached its saturation point. From this moment on (that is to say, for large seed sets) our method starts outperforming the theoretically optimal algorithm w.r.t. the expected spread objective function.

Our empirical findings are consistently confirmed by a thorough experimentation over several influence networks which are the typical benchmarks used in the influence maximization literature, and by using different ways of learning/assigning the influence probability to each link.

To the best of our knowledge our work is the first to show consistent improvement in terms of quality over the standard greedy algorithm for influence maximization.

For repeatability sake our software and datasets are publicly available at http://tinyurl.com/pxl9h89.

Roadmap. In Section 2 we first provide preliminary notions and notations, then we introduce the Typical Cascade problem, and study its hardness. In Section 3 we develop the theory behind our method based on sampling and Jaccard median, and we derive the bounds on the number of samples needed to have a good multiplicative approximation. In Section 4 we present the practical algorithm, while in Section 5 we show the application to influence maximization. Section 6 contains our experiments, Section 7 covers the related work, and Section 8 concludes the paper by summarizing the results and discussing future lines of investigation.

2. THE TYPICAL CASCADE PROBLEM

2.1 Preliminaries

Let $\mathcal{G} = (V, E, p)$ be a probabilistic directed graph, where $p : E \to (0, 1]$ is a function that assigns a probability of existence to each edge. Following the literature, we consider the edge probabilities independent [31, 22, 23, 6]. In this setting, the *possible-world* semantics [1, 13] is a principled way of defining the meaning of a query over uncertain data. Specifically, the possible-world semantics interprets $\mathcal{G}$ as a probability distribution over subgraphs of (V, E) defined by choosing every edge $e \in E$ independently at random with probability $p(e)$. That is, the probability of observing any possible world $G = (V, E_G) \sqsubseteq \mathcal{G}$ is:

$$\Pr(G) = \prod_{e \in E_G} p(e) \prod_{e \in E \setminus E_G} (1 - p(e)). \qquad (1)$$

Let $q(G)$ be a function that when applied to a deterministic graph G returns a value in $\mathbb{R}$. Following the possible-world semantics querying q over the probabilistic graph $\mathcal{G}$ is typically done by asking for its expected value:

$$q(\mathcal{G}) = \mathop{\mathbb{E}}_{G \sim \mathcal{G}} q(G) = \sum_{G \sqsubseteq \mathcal{G}} q(G) \Pr(G).$$

When $q(G)$ is a binary predicate then the expectation corresponds to the probability that the predicate is satisfied. For example, this is the case for instance of *reachability* query $r(u, v)$, which returns true if v is reachable trough a directed path from u. In the context of probabilistic graphs, the corresponding *reliability* query would ask for the probability of v being reachable from u.

In this paper we are interested in a type of query which returns neither a scalar nor a binary, but a set of nodes. Given a directed probabilistic graph $\mathcal{G} = (V, E, p)$ and a node $s \in V$ we are interested in the *cascade* originated from s, i.e., the set of nodes that would get infected if s get infected. In the case of a deterministic, graph that would be the set of nodes reachable from s trough directed paths. But how to determine the typical cascade in a probabilistic graph?

2.2 Problem statement

In a sense we want to define a notion of "expected" or "typical" cascade: this is a cascade which is *the closest to the set of possible cascades of s*. Towards formalizing this intuition, we need a metric to compute the distance among two possible cascades. As previously stated, a cascade simply corresponds to a set of nodes, we therefore use *Jaccard distance*. Given two sets of nodes $A, B \subseteq V$, their Jaccard distance is defined as

$$d_J(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} = \frac{|A \oplus B|}{|A \cup B|},$$

which is known to be a metric (see, e.g., [10]).

Given a deterministic graph $G \sqsubseteq \mathcal{G}$ and a node $s \in V$, we denote by $R_s(G)$ the set of nodes reachable from s in G. Given $\mathcal{G}$, $s \in V$ and a set of nodes $C \subseteq V$, we define the *expected cost*, $\rho_{\mathcal{G},s}(C)$, of C as the expected Jaccard distance between C and a random cascade generated from s:

$$\rho_{\mathcal{G},s}(C) = \mathbb{E}[d_J(R_s(G), C)] = \sum_{G \sqsubseteq \mathcal{G}} d_J(R_s(G), C) \Pr(G).$$

(We omit the dependence on $\mathcal{G}$ or s when appropriate.)

This represents a measure of the *stability*, i.e., how much random cascades from s deviate from C. It is therefore desirable to find the set $C^* \subseteq V$ that minimizes the expected cost $\rho_s(C^*)$. This set represents the *typical cascade* of the node s, or what we abstractly call its sphere of influence.

Problem 1 (Typical cascade). *Given a probabilistic graph $\mathcal{G} = (V, E, p)$ and a source node $s \in V$ find the set $C^* \subseteq V$ that minimizes the expected cost:*

$$C^* = \operatorname*{arg\,min}_{C \subseteq V} \rho_{\mathcal{G},s}(C).$$

We next study the complexity of our problem.

2.3 Complexity of Problem 1

The first source of complexity for the Typical Cascade problem is that for a given source, computing the expected cost of a set of nodes is #**P**-hard.

Theorem 1. *Given a probabilistic graph $\mathcal{G} = (V, E, p)$, a source node $s \in V$, and a set of nodes $C \subseteq V$, computing $\rho_{\mathcal{G},s}(C)$ is #**P**-hard.*

Proof. We employ a reduction from *s*-*t* reliability on graphs: given a directed probabilistic graph G and two nodes $s, t \in V(G)$, compute the probability that there is a path from s to t, denoted $rel(G, s, t)$. This problem was shown #**P**-hard by Valiant [38].

Consider an instance $\langle G, s, t\rangle$ of *s*-*t* reliability, where $G = (V, E)$. Let $n = |V|$ and consider the graph G' which is equal to G except that we add an arc from t to every node in the graph with existence probability 1. We calculate the cost of the two candidate medians $H_1 = V$ and $H_2 = V \setminus \{t\}$ and show that determining the reliability $rel(G, s, t)$ can be reduced in polynomial time to the computation of $\rho_{G',s}(H_1)$ and $\rho_{G',s}(H_2)$, proving the theorem.

Denote by $R(C)$ the event that a random cascade C from s in G reaches t, and let $U(C) = 1 - R(C)$ denote the complementary event. Note that there is a natural measure-preserving mapping ϕ between cascades in G and cascades in G': $\phi(C) = V$ if $R(C)$ holds, and $\phi(C) = C$ otherwise. In particular, for very $X \subseteq V$ it holds that

$$\rho_{G',s}(X) = \mathbb{E}_C[d_J(\phi(C), X)].$$

For any cascade C in G two cases may occur:

(a) $R(C) = 1$ (i.e., $t \in C$). Then $\phi(C) = V$ so $d_J(\phi(C), H_1) = 0$ and $d_J(\phi(C), H_2) = \frac{1}{n}$.

(b) $R(C) = 0$ (i.e., $t \notin C$). Then $\phi(C) = C$ so $d_J(\phi(C), H_1) = \frac{n-|C|}{n}$ and $d_J(\phi(C), H_2) = \frac{n-1-|C|}{n-1}$.

Therefore

$$\begin{aligned}\rho_{G',s}(H_1) &= \mathbb{E}_C\left[R(C)\cdot 0 + U(C)\cdot\frac{n-|C|}{n}\right]\\ &= \mathbb{E}_C\left[U(C)\cdot\frac{n-|C|}{n}\right]\\ \rho_{G',s}(H_2) &= \mathbb{E}_C\left[R(C)\cdot\frac{1}{n} + U(C)\cdot\frac{n-|C|-1}{n-1}\right]\\ &= \mathbb{E}_C\left[\big(1-U(C)\big)\cdot\frac{1}{n} + U(C)\cdot\frac{n-|C|-1}{n-1}\right].\end{aligned}$$

Manipulating these expressions yields

$$\begin{aligned}&n\cdot\rho_{G',s}(H_1) - (n-1)\cdot\rho_{G',s}(H_2) =\\ &\mathbb{E}_C\left[U(C)\cdot(n-|C|)\right] -\\ &-\mathbb{E}_C\left[\big(1-U(C)\big)\cdot\frac{n-1}{n} + U(C)\cdot(n-|C|-1)\right] =\\ &\mathbb{E}_C\left[U(C)\cdot\left(2-\frac{1}{n}\right) - 1 + \frac{1}{n}\right] = q\cdot\left(2-\frac{1}{n}\right) - 1 + \frac{1}{n},\end{aligned}$$

where $q = \mathbb{E}_C[U(C)] = \Pr_C[t \notin C]$ is the unreliability probability of t from s. Therefore the reliability is

$$rel(G, s, t) = 1 - q = \frac{1 - \frac{1}{n} - n\rho_{G',s}(H_1) + (n-1)\rho_{G',s}(H_2)}{2 - \frac{1}{n}},$$

which shows that evaluating $\rho_{G',s}$ is #**P**-hard. □

A natural approach to deal with some #**P**-hard problems is by means of Monte-Carlo sampling: this means to sample a large enough number ℓ of independent cascades $\mathcal{S} = \{S_1, \ldots, S_\ell\}$ from s, and use them to compute an estimate $\bar{\rho}_s(C)$ as the empirical mean of $d_J(C, s_i)$ over the cascades sampled:

$$\bar{\rho}_s(C) = \frac{1}{\ell}\sum_{i\in[\ell]} d_J(C, \tau_i).$$

This is an unbiased estimator of the actual cost $\rho_s(C)$ as defined above, so one may hope to use $\bar{\rho}_s(C)$ as a proxy for the actual cost $\rho_s(C)$ and attempt to solve the following related optimization problem:

Problem 2 (Jaccard median). *Given a finite set V and a collection $\mathcal{S}$ of ℓ sets $S_1, \ldots, S_\ell \subseteq V$, find a set $\bar{C}^* \subseteq V$ that minimizes the average Jaccard distance of $\bar{C}^*$ from the elements of $\mathcal{S}$.*

$$\bar{C}^* = \operatorname*{arg\,min}_{C \subseteq V} \bar{\rho}_s(C).$$

Chierichetti et al. [11] show that Problem 2 is **NP**-hard, and present a polynomial-time approximation scheme.

The difference between Problems 1 and 2 is that in the latter we are given a list of sets, while the first one defines implicitly a distribution over exponentially many sets. Neither one seems easily reducible to the other, though: on the one hand, enumerating all subgraphs to apply Jaccard median to our problem would require exponential time; on the other hand, a solution to the typical cascade problem may not extend to a general solution to Jaccard median, since the set of possible cascades from a vertex in a graph has certain special properties (for example, closure under unions).

3. SAMPLING AND JACCARD MEDIAN

In the previous section we hinted at a possible approach to tackle the Typical Cascade problem:

1. sample ℓ random cascades from source nodes s;
2. compute their Jaccard median as the typical cascade.

An important question is how many deterministic graphs we need to sample in order to obtain a good estimate of the median quality, and to avoid overfitting in the scheme above. To fix notation, let $\mathcal{C}$ denote a distribution over non-empty subsets of $[n]$ (for example, $\mathcal{C}$ could be the reachability sets from a given vertex in an n-vertex uncertain graph). Let

$$\rho(X) = \mathop{\mathbb{E}}_{C \sim \mathcal{C}}[d_J(C, X)]$$

denote the cost of a candidate solution X, and

$$M^* = \mathop{\arg\min}_{M \subseteq [n]} \rho(M)$$

denote an optimal median with cost $\epsilon^* = \rho(M^*)$.

Recall that we cannot evaluate $\rho(X)$ efficiently, so we resort to sampling ℓ independent elements of $\mathcal{C}$ and using the empirical mean $\widetilde{\rho}(X)$ as an estimator for $\rho(X)$. Then we derive a median M that approximately minimizes $\widetilde{\rho}_s$ on the sample, with the hope that its actual cost $\rho(M)$ will be close to optimal.

While one can easily show that the cost of any particular set X is approximately preserved (with additive error) in the sample, this may not hold simultaneously for all sets. The situation is analogous with the problem of overfitting in learning theory: while the error of any given classifier can be accurately estimated from the training set, if the learner's hypothesis class is large enough it may happen that we find a hypothesis that does exceptionally well on the training set, but performs badly on the test set. (In our setting, the set $2^{[n]}$ of all "candidate medians" play the role of the hypothesis class.) In fact, as there are 2^n candidate medians, a naive estimate via the union bound would suggest that $\Theta(n)$ samples are needed, which is too large a sampling size to be practical (recall that n is the number of nodes of the graph in our application). Fortunately this is far from tight: our next result shows that these bounds can be substantially improved, as long as the cost of the optimal median is small: a *constant* number of samples suffice to get good *multiplicative* approximations.

Theorem 2. *Let $\widetilde{M}^* = \arg\min_{X \subseteq [n]} \widetilde{\rho}(X)$. For $\delta \geq \exp(-\ell/10)$, the following holds with probability at least $1-\delta$:*

$$\rho(\widetilde{M}^*) \leq \left(1 + O\left(\epsilon^* + \sqrt{\frac{\log(\ell/\delta)}{\ell}}\right)\right)\epsilon^*,$$

More generally, whenever $\widetilde{\rho}(\widetilde{Y}) \leq (1+\beta)\widetilde{\rho}(\widetilde{M}^*)$, we have

$$\rho(\widetilde{Y}) \leq \left(1 + O\left(\beta + \epsilon^* + \sqrt{\frac{\log(\ell/\delta)}{\ell}}\right)\right)\epsilon^*,$$

In particular, for any $\alpha > \epsilon^*$, a sample of size $\ell = \log(1/\alpha)/\alpha^2$ suffices to obtain an $(1 + O(\alpha))$-approximate median. This is remarkable, because the number of samples is independent of n and moreover, it does not suffice in general to estimate the cost of a candidate solution with small multiplicative error (this would require $\ell = \Omega(1/\epsilon^*)$). This means that the empirical and true costs may differ significantly, yet the cost of the *solution* found by solving the empirical problem is very close to the true optimal. The result also yields a sublinear-time randomized approximation algorithm for *standard* Jaccard median (Problem 2) when the number of input sets is large: sample $O(\log(1/\alpha)/\alpha^2)$ of the input sets and work on the smaller instance.

Roughly speaking, the proof of Theorem 2 proceeds as follows. We show that a) any nearly optimal median X gives rise to an easily manageable approximate cost function f_X with certain properties implying that no median can do much better than X (Lemma 1); and b) these properties of f_X are approximately preserved after sampling (Lemma 2). This trick allows us to convert the statement "for all candidate medians, their sample cost is not much smaller than $\rho(M^*)$" into a statement regarding the *single* function f_X (see Lemma 2), which can be proved directly. The proof of these intermediate lemmas may be found in Appendix A.

Lemma 1. *Let $X \subseteq [n]$ and define*

$$f_X(Y) = \mathop{\mathbb{E}}_{C \sim \mathcal{C}}\left[\frac{|Y \oplus C|}{|X \cup C|}\right].$$

The following hold for all $Y, Y' \subseteq [n]$:

(a) $d_J(Y, Y') \leq \min\left(\rho(Y) + \rho(Y'), 6(\rho(X) + f_X(Y) + f_X(Y'))\right)$.

(b) If $X \cap Y \neq \emptyset$, then $1 - d_J(X, Y) \leq \frac{\rho(Y)}{f_X(Y)} \leq \frac{1}{1-d_J(X,Y)}$.

(c) If $\rho(Y) \leq \rho(X)$, then $f_X(Y) \leq \frac{f_X(X)}{1-2f_X(X)}$.

Intuitively, in order to prove that X is a nearly optimal median, it suffices to show that X is an approximate minimizer of $f_X(Y)$.

Denote by $\mathcal{D}$ the uniform distribution over the sample $S_1, \ldots, S_\ell$, and observe that $\widetilde{\rho}(X) = \mathbb{E}_{D \sim \mathcal{D}}[d_J(X, D)]$.

Lemma 2. *Let $X \subseteq [n]$. Define $Z = \arg\min_{Y \subseteq [n]} f_X(Y)$ and $\widetilde{Z} = \arg\min_{Y \subseteq [n]} \widetilde{f}_X(Y)$, where $\widetilde{f}_X(Y) = \mathbb{E}_{D \sim \mathcal{D}}\left[\frac{|Y \oplus D|}{|X \cup D|}\right]$. With probability at least $1 - \delta$,*

$$f_X(\widetilde{Z}) \leq \left(1 + O\left(\sqrt{\frac{\log(\ell/\delta)}{\ell}}\right)\right) f_X(Z),$$

provided $\rho(X)$ is below some constant and $\delta \geq \exp(-\ell/10)$.

Moreover, whenever $\widetilde{f_X}(Y) \leq (1+\beta)\widetilde{f}_X(\widetilde{Z})$, we have

$$f_X(\widetilde{Y}) \leq \left(1 + O\left(\beta + \sqrt{\frac{\log(\ell/\delta)}{\ell}}\right)\right) f_X(Z).$$

We are now equipped with the tools needed to prove Theorem 2.

Proof. Consider the optimal median M^* with cost $\rho(M^*) = \epsilon^*$, and the optimal solution to the empirical median $\widetilde{M}^* = \arg\min_X \widetilde{\rho}(X)$. Since $f_X \leq 1$, we may assume ϵ^* is bounded above by a suitable constant. By Lemma 2, with probability at least $1 - \delta$ it holds that $\widetilde{\rho}(M^*) \leq \rho(M^*)(1 + O(\sqrt{\log(\ell/\delta)/\ell})) \triangleq \lambda$. By definition, $\rho(M^*) \leq \rho(\widetilde{M}^*)$ and $\widetilde{\rho}(\widetilde{M}^*) \leq \widetilde{\rho}(M^*)$, so by applying Lemma 1 to $\mathcal{C}$ and $\mathcal{D}$,

$$2d_J(M^*, \widetilde{M}^*) \leq (\rho(M^*) + \rho(\widetilde{M}^*)) + (\widetilde{\rho}(M^*) + \widetilde{\rho}(\widetilde{M}^*)),$$

i.e., $d_J(M^*, \widetilde{M}^*) \leq \epsilon^* + \lambda$.

We introduce the following shorthand notation for comparing costs: $x \preceq y$ if $x \le y/(1-O(\epsilon^*+\lambda))^{O(1)}$, $x \succeq y$ if $y \preceq x$, and $x \approx y$ if both $x \succeq y$ and $x \preceq y$ hold. Let Z minimize $f_{M^*}(Y)$ and $\widetilde{Z}$ minimize $\widetilde{f}_{M^*}(Y)$. Then we have $d_J(M^*, Z) \le \rho(M^*) + f_{M^*}(M^*) + f_{M^*}(Z) = 2f_{M^*}(M^*) + f_{M^*}(Z) \le 3\epsilon^*$, and $\rho(Z) \le \epsilon^*/(1-3\epsilon^*) = O(\epsilon^*)$ for small enough ϵ^*. Likewise, $d_J(\widetilde{M}^*, Z) \le \rho(\widetilde{M}^*)+\rho(Z) \le O(\lambda+\epsilon^*)$ and $d_J(\widetilde{M}^*, \widetilde{Z}) \le \widetilde{\rho}(\widetilde{M})+f_{\widetilde{M}}(\widetilde{Z}) \le \lambda+2\widetilde{f}_M(\widetilde{M}) \le O(\lambda+\epsilon^*)$.

By definition, $f_{M^*}(Z) \le f_{M^*}(M^*)$ and $\widetilde{f}_{M^*}(\widetilde{Z}) \le \widetilde{f}_{M^*}(Z)$. By Lemma 1, $f_{M^*}(M^*) \preceq f_{M^*}(Z)$, so $f_{M^*}(Z) \approx f_{M^*}(M^*) = \rho(M^*)$. The same lemma applied to $\widetilde{f}_{M^*}$ yields $\widetilde{f}_{M^*}(\widetilde{M}^*) \approx \widetilde{f}_{M^*}(\widetilde{Z}) \le \widetilde{f}_{M^*}(Z)$, thus $f_{M^*}(\widetilde{Z}) \preceq f_{M^*}(Z)$ and so $f_{M^*}(\widetilde{Z}) \approx f_{M^*}(Z)$. Hence, by Lemma 2, with probability at least $1 - O(\delta)$,

$$f_{M^*}(\widetilde{M}^*) \approx f_{M^*}(\widetilde{Z}) \approx f_{M^*}(Z).$$

If we divide δ by a constant factor and repeat the argument, we obtain the first part Theorem 2. The second part is proved similarly by using the "moreover" part of Lemma 2. □

4. PRACTICAL ALGORITHMS

Next we put together the pieces from the theoretical insights introduced in the previous section, and discuss practical efficiency considerations. First, we describe an indexing scheme enabling efficient simulations of the cascades needed. Then we present the main algorithm to compute a typical cascade for every node of $\mathcal{G}$.

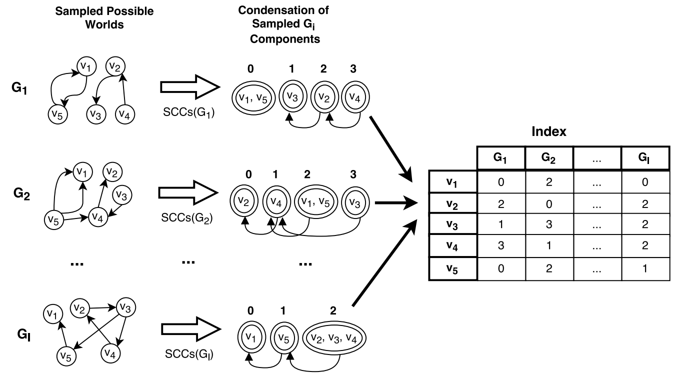
In order to obtain the typical cascade for a given vertex v, we first need to produce a certain number ℓ of cascades from v. As we saw before, taking $\ell = O(\log(1/\alpha))$ samples is enough to obtain a $(1+\alpha)$-approximation provided that the cost is $\Omega(\alpha)$; if we wish this guarantee to hold simultaneously for all vertices, we may take $\ell = O(\log(n/\alpha)/\alpha^2)$. Rather than sampling separately for each vertex, we sample ℓ possible worlds $G_1, \ldots, G_\ell$ from $\mathcal{G}$, each of which implicitly defines a sample cascade from each vertex $v \in V(G)$, which may be obtained by performing a DFS traversal of G rooted at v.

A key observation that we exploit to speed up this process is that all the vertices in the same *strongly connected component* (SCC) have the same reachability set: since any two vertices u, v in the same SCC are reachable from each other, any vertex reachable by u is also reachable by v, and viceversa. Therefore we can represent each sampled possible world G_i by its SCC structure. Representing G_i in terms of its SCCs yields savings in both space usage and computational runtime, because of the compactness of representation and because a single DFS is sufficient to identify the reachability set of all vertices in the same component.

Based on these observations we build an index that contains for all the sample possible worlds $G_1, \ldots, G_\ell$:

1. the *condensation* C_i of G_i, that is, the directed acyclic graph of links between SCCs, obtained by contracting [42] each component of G_i to a single vertex;
2. for each vertex v and index i, the identifier of the connected component of v in G_i (see Figure 2);

Computing the SCCs and performing their contraction can be performed in time linear in the total number of vertices and edges of the graphs sampled [36].



	G$_1$	G$_2$	...	G$_l$
v$_1$	0	2	...	0
v$_2$	2	0	...	2
v$_3$	1	3	...	2
v$_4$	3	1	...	2
v$_5$	0	2	...	1

Figure 2: Cascade index: for each sampled possible world G_i it is stored the structure made of the condensation of the SCCs and a matrix indicating for each vertex v and each possible world G_i, the index of the component to which v belongs in G_i.

```
Algorithm 1: Index construction
  Input  : Input graph $\mathcal{G}$ and number of samples $\ell$.
  Output: Index $\mathcal{I}$, Component Pointers $\mathcal{P}$
  $\mathcal{I} \leftarrow [|V| \times \ell]$
  $\mathcal{P} \leftarrow [1 \times \ell]$
  for $i \leftarrow 1$ to $k$ do
      Sample $G_i$ from $\mathcal{G}$
      $SCCs \leftarrow StronglyConnectedComponents(G_i)$
      $\mathcal{P}[i] \leftarrow transitiveReduction(SCCs)$
      foreach $v \in V$ do
          $\mathcal{I}[v, i] \leftarrow nodeComponentIndex(v,\ SCCs)$
      end
  end
  return ($\mathcal{I}$, $\mathcal{P}$)
```

To further reduce the space consumption, we perform the transitive reduction [3] of the condensation of C_i, i.e., find the unique graph T_i (not necessarily a subgraph of C_i) with vertex set $V(C_i)$ that preserves the reachability/non-reachability between every pairs of vertices of C_i and has the smallest number of edges. While the worst-case computational complexity of this task is theoretically equivalent to that of Boolean matrix multiplication [3], for which the best algorithms known run in time $O(n^{2.373})$, in the practical instances arising in our experiments the algorithm from [3] proved adequate.

The procedure to construct the index is summarized in Algorithm 1.

Given a node v and $i \in [\ell]$, the cascade of v in G_i can be obtained as follows: look at the identifier of the SCC of v in G_i; recursively follow the links from the associated condensed vertex in C_i to find all the reachable components; and output the union of the elements in the reachable components. The time to perform this computation is linear in the number of nodes of the output and the number of edges of the condensation C_i, which is typically much smaller than the number of edges of G_i.

For the computation of the typical cascade C_v of a node v, we need to compute an approximate Jaccard median of the collection of ℓ cascades $S_1, \ldots, S_\ell$ from v. To this end, we use the work of Chierichetti et al. [11]. Their PTAS to achieve arbitrarily good approximations is mostly of theoretical interest, so we use the algorithm described in Section 3.2 of [11], which achieves an $1 + O(\epsilon)$ factor approximation (where ϵ is the cost of the optimal median of the instance) and runs in time $\widetilde{O}(k + \sum_i |S_i|)$.

Algorithm 2: All Typical Cascades

```
Input  : Input graph $\mathcal{G}$, number of samples $\ell$.
Output: The typical cascades for each $v \in G$.
$(\mathcal{I}, \mathcal{P}) \leftarrow$ Index$(\mathcal{G}, \ell)$
for $vinG$ do
    $S \leftarrow [1 \times \ell]$ (list of cascade sets)
    for $i \leftarrow 1$ to $\ell$ do
        $c \leftarrow \mathcal{I}[v, i]$
        $cG \leftarrow reachable\_components(\mathcal{P}[i], c)$
        $S[i] \leftarrow \bigcup\{nodes(c) \mid c \in cG\}$
        $C_v \leftarrow JaccardMedian(S)$ (by Chierichetti et al. [11])
    end
end
return $\{(v, C_v) \mid v \in V(\mathcal{G})\}$
```

5. INFLUENCE MAXIMIZATION

In this section we present, as an application of the typical cascade computation, a novel approach to influence maximization. While our approach is heuristic in nature, it is motivated by the observations below.

1. Given a seed set $\mathbb{S}$, we can define its stability as the expected cost of its typical cascade C^* exactly as we have done for singleton nodes in Section 2. If this cost is small we say that the seed set is reliable.

2. If $\mathbb{S}$ is a highly reliable seed set, the size of its typical cascade C^* is very close to the mean size of cascades from $\mathbb{S}$ (see [11]). In other words, by optimizing for size of the typical cascade we are also indirectly optimizing for expected spread, unless the optimal solution is unreliable (which is ruled out by the next item).

3. It is an empirically observable phenomenon (see the stability analysis in Sec. 6) that the expected cost of the typical cascade of $\mathbb{S}$ tends to decrease as $\mathbb{S}$ grows. Intuitively, this means that the cascading process becomes more and more deterministic (or predictable) as the size of the seed set increases. We want to leverage this fact by acting as if the cascade from $\mathbb{S}$ were effectively C^*. However, for sake of efficiency, in our method we will not use the typical cascade of $\mathbb{S}$, but instead we will use the union of the typical cascades of all the seed nodes in $\mathbb{S}$. This is justified by the next point.

4. It can be shown that some nearly optimal typical cascade from seed set $\mathbb{S}$ is a superset of the typical cascades for the cascades induced by the individual elements of $\mathbb{S}$. This is because if the typical cascade has cost ϵ, then simply selecting all elements that are present with probability at least 1/2 can be proved to be a solution with cost at most $\epsilon + O(\epsilon^{3/2})$ [11]. But note that the probability that a given vertex is reachable from $\mathbb{S}$ is monotonically increasing with $\mathbb{S}$, so the set of elements reachable with probability at least 1/2 is monotonically increasing as well. Consequently the nearly optimal typical cascade for $\mathbb{S}$ can be assumed to contain the typical cascades for all its elements. (It may contain further elements, which would increase the solution size and decrease its cost, so we are being conservative by ignoring them.)

These observations motivate us to approach the influence maximization problem as max-cover problem over the typical cascades of the singleton nodes. Let $\mathbb{S} \subseteq V$ denote a set of nodes and let C_v be the median cascade of each $v \in V$. Write $\Phi(\mathbb{S}) = \bigcup_{v \in \mathbb{S}} C_v$ for the elements covered by the typical cascades of all the nodes in $\mathbb{S}$, Now, given an finite integer $k \leq |V|$, our goal is to find a set $\mathbb{S}^*$ such that the coverage $\Phi(\mathbb{S}^*)$ is maximized for $|\mathbb{S}^*| = k$.

Formally,

$$\mathbb{S}^* = \underset{\substack{\mathbb{S} \subseteq V: \\ |\mathbb{S}| = k}}{\arg\max}\, \Phi(\mathbb{S})$$

This is an instance of the maximum coverage problem, which can be approximated by the standard greedy algorithm that runs for k iterations and at each iteration it selects a node v whose addition increases the value of Φ the most. This approach, whose pseudocode is shown in Algorithm3, is named InfMax_TC: *Influence Maximization using Typical Cascades.*

Algorithm 3: InfMax_TC

```
Input  : Typical cascades for all $v \in V$: $\{C_1, C_2, \cdots, C_{|V|}\}$
Output: Seed set of $k$ nodes: $\mathbb{S}^*$
$\mathbb{S}^* \leftarrow \emptyset$
for $i \leftarrow 1$ to $k$ do
    $u = \arg\max_{u \in V \setminus \mathbb{S}^*} \Phi(\mathbb{S}^* \cup u)$
    $\mathbb{S}^* \leftarrow \mathbb{S}^* \cup u$
end
return $\mathbb{S}^*$
```

In the next section we compare this method for influence maximization with the standard method, w.r.t. the objective function of the influence maximization problem: the expected spread.

6. EXPERIMENTS

In our experiments we first report basic statistics about the spheres of influence (or typical cascades) and their computation, such as their size, cost and the running time of our procedure. We then focus on the main goal of our experimental assessment which is to show the performance, in terms of quality, of our method for influence maximization.

The majority of the literature on influence maximization uses a set of benchmark social graphs, where the influence probability for each edge is artificially assigned according to some certain standard methods. Perhaps more appropriately, some authors [16, 33] have started to use influence probabilities learnt from a log of past user activity. In our experiments, for sake of exhaustiveness, we follow both approaches. We use six datasets which are often used as benchmarks in the literature for influence maximization: in three of these the edge probabilities are *learnt*, whereas for the other three the probabilities are *assigned* artificially as described later. Moreover, we use two different methods for learning the contagion probabilities and two different methods for assigning them. This gives us a total of 12 datasets to work with.

Datasets	$\lvert V \rvert$	$\lvert E \rvert$	Type	Probabilities
Digg	$68K$	$875K$	directed	learnt
Flixster	$137K$	$1.2M$	undirected	learnt
Twitter	$23K$	$650K$	undirected	learnt
NetHEPT	$15K$	$31K$	undirected	assigned
Epinions	$76K$	$509K$	directed	assigned
Slashdot	$77K$	905	directed	assigned

Table 1: Dataset characteristics.

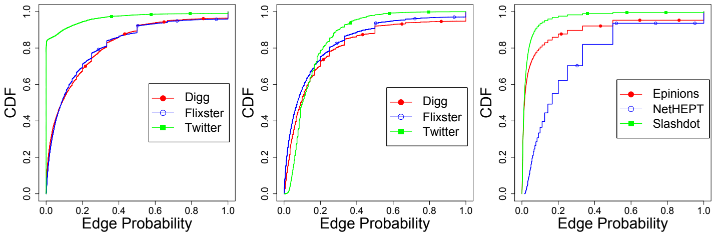


Figure 3: CDF of edge probabilities learnt from Saito (left), Goyal (center), and WC model (right). We do not report the distribution for the fixed probability method as this is not meaningful.

6.1 Dataset description

The three datasets that come with a log of users activity and can thus be used for learning the influence probabilities are Digg, Flixster and Twitter.

Digg is a news portal that allows users to submit news stories, as well as rate the posted stories by means of *voting*. The ratings are then used to promote stories on the front page of Digg portal. The data snapshot we use is related to the the voting history of all the stories *promoted* during June, 2009 [20]. It has $3M$ votes for $3.5K$ stories. The data also provides a *fan network* from which a directed social graph is induced.

Flixster is an online social networking service (https://flixster.com/) enabling its users to rate and review movies. Here, we have user ratings from November 2005 to November 2009 [21]. This amounts to $8.2M$ ratings for $49K$ items/movies.

Twitter. The final dataset in this category is a snapshot of Twitter, obtained by crawling its public timeline[3]. The items in Twitter represent the URLs propagating across the network. Unlike the previous two datasets, in Twitter the user activity corresponds to sharing/resharing of the URLs instead of rating items. The data contains $6K$ items and $383K$ user activities.

The other three datasets are from the SNAP dataset collection [28]. These include NetHEPT, Epinions, and Slashdot. The first is a network of citations, whereas the other two are social networks. These datasets are widely used in the study of social networks and influence maximization [35, 9]. Table 1 reports basic statistics on the datasets used. When a graph is undirected, we just consider the edges existing in both directions.

Datasets	$avg(\lvert\widetilde{C^*}\rvert)$	$sd(\lvert\widetilde{C^*}\rvert)$	$max(\lvert\widetilde{C^*}\rvert)$
Digg-S	9.0	22.2	263
Flixster-S	4.3	12.0	439
Twitter-S	17.0	86.4	1459
Digg-G	7.3	17.0	130
Flixster-G	999.5	822.7	2589
Twitter-G	24.9	58.4	1727
NetHEPT-W	3.0	1.2	13
Epinions-W	3.6	8.6	684
Slashdot-W	4.8	19.4	420
NetHEPT-F	1067.5	915.5	4138
Epinions-F	4774.5	1574.4	6345
Slashdot-F	1337.0	841.5	5574

Table 2: In the table $\lvert\widetilde{C^*}\rvert$ denotes the size of the approximated typical cascade computed and we report its average, standard deviation, and maximum over all nodes in the graph.

6.2 Edge probabilities

Below we describe the two different ways of learning the influence probabilities, and the two different ways of artificially assigning the influence probabilities, that we use in our experiments.

Learning from real-world cascades. The datasets in the first category (Digg, Flixster, and Twitter) provide us two key elements: (i) a social network (ii) log of user activities (for different items) with the corresponding timestamps. Both methods we use exploit these two pieces of input to learn the edge probabilities. The first method is by Saito et al. [33], which model the learning of the influence probabilities as a likelihood maximization problem and devise an EM algorithm to solve it.

The second method by Goyal et al. [16] follows a frequentist approach. Among the various models they propose we use the simplest one: the probability assigned to an edge (u, v) is simply the number of times in the propagation log in which v performs an action after u, divided by the number of actions performed by u [16].

In the following we will use a suffix -S and a suffix -G to denote the datasets with the probabilities learnt by following Saito et al. [33] and Goyal et al. [16] respectively.

Artificial assignments. For the second group of datasets (NetHEPT, Epinions, and Slashdot), we use two different methods for artificially assigning probability to each edge. The first the methods is the weighted cascade (WC) model [9], which sets the probability $p_{u,v}$ over an edge (u, v) as: $p_{u,v} = \frac{1}{inDeg(v)}$. Here, $inDeg(v)$ is the in-degree of node v. In the second method, we assign a fixed probability $p_{u,v} = 0.1$ to each edge (u, v).

In the following we will use a suffix -W and a suffix -F to denote the datasets with the probabilities assigned by weighted cascade method and fixed respectively.

Using the methods of learning/assigning edge probabilities explained above, we have 12 datasets in total for our experiments detailed in this section. Figure 3 reports the CDFs of the edge probabilities in all datasets.

[3] https://dev.twitter.com/rest/reference/get/statuses/user_timeline

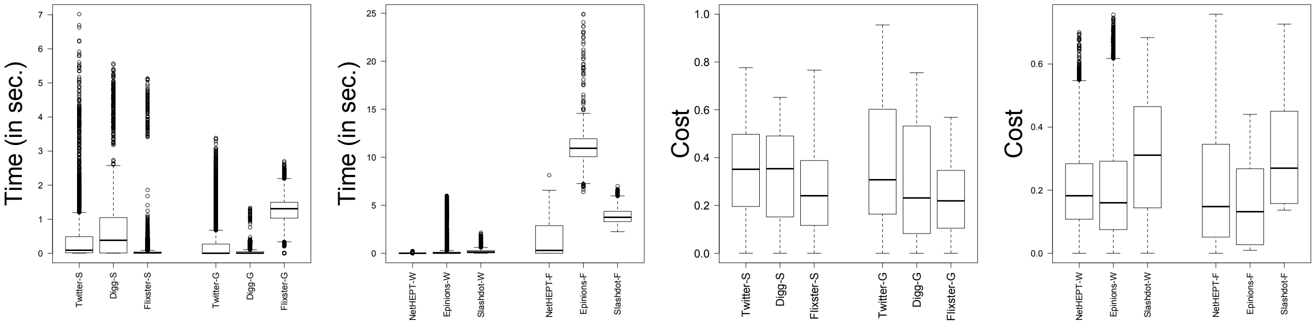


Figure 4: Time taken to compute the typical cascade $\widetilde{C^*}$ (two left-most plots) and its expected cost $\rho_{\mathcal{G},s}(\widetilde{C^*})$ (two right-most plots).

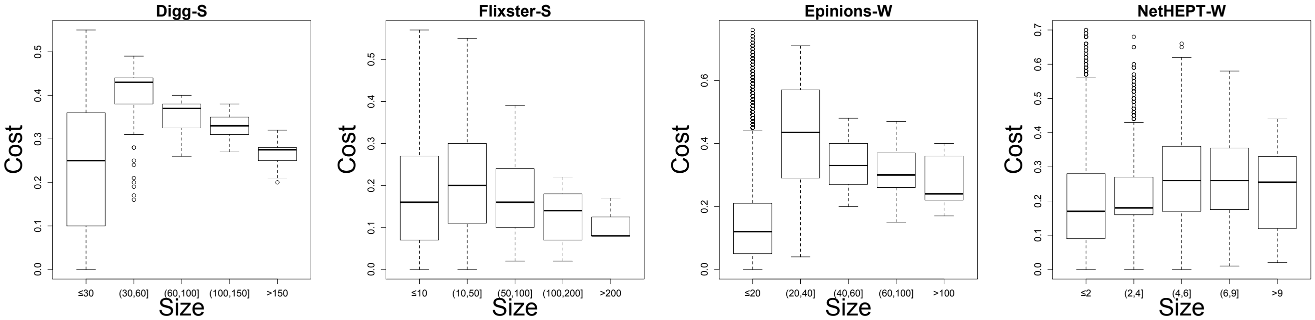


Figure 5: Distribution of the expected cost $\rho_{\mathcal{G},s}(\widetilde{C^*})$ of the typical cascade w.r.t. its size.

6.3 Computing the typical cascades

Table 2 reports basics statistics on the size of the sampled cascades (S_i) and the typical cascade computed from the samples ($\widetilde{C^*}$). Given that the edge probabilities learnt using the method of Goyal et al. [16] are larger than the probabilities learnt by means of Saito et al. [33] method (See Figure 3), not surprisingly also the average size of typical cascades is larger for the former than for the latter.

The pattern is even more evident in Flixster, which showcases the fact that different strategies used to assign probabilities may greatly impact the size of the samples, and thereby, the size of the corresponding typical cascade. We also note that when artificially assigned, the probabilities set fixed to 0.1 also result in larger sampled cascades, and thus larger typical cascade, than produced when assigning probabilities by means of the WC model.

In Figure 4 we report the time taken to compute the typical cascade $\widetilde{C^*}$ and its expected cost $\rho_{\mathcal{G},s}(\widetilde{C^*})$, excluding the index construction. That is, this is the time to extract the cascades from the index and run the Jaccard median approximation on this instance. (Recall that we are using 1000 samples so the number of elements to process per vertex is often in the hundreds of thousands.) The times reported use a Python implementation, on a Intel Xeon 2.2 Ghz with 6 cores and 16 GB memory. As depicted, the time remains almost always well under 1 second except for a small number of nodes. As regards the expected costs they rarely exceed 0.4, and in most of the dataset the average is around 0.2.

Figure 5 reports the distributions of the expected cost w.r.t. the size of the typical cascade, in order to assess whether the quality (or reliability) of the solution also depends on its size. In every plot, if we disregard the bucket of very small cascades, which is in any case not very interesting for applications, we can observe that the larger is the typical cascade, the more reliable it is (smaller cost). This becomes even more evident when observing the maximum cost observed: it is practically impossible to find a large typical cascade with large cost. This matches the intuition discussed in Section 1.

6.4 Influence maximization

We next present our main practical result: the fact that our method for influence maximization based on spheres of influence (presented in Section 5) outperforms the standard influence maximization method for what concerns quality, i.e., the expected spread achieved.

Quality of influence maximization. In the following, the standard greedy (theoretically optimal) algorithm for influence maximization [24] is denoted InfMax_std, and our greedy algorithm for maximum coverage using the sphere of influence (typical cascade) of each node is denoted InfMax_TC. In all the experiments we use $k = 200$ for the seed set size and we use the same number of sampled cascades (1000) for both methods: to estimate the expected spread for InfMax_std, and for computing the typical cascades for InfMax_TC. The expected spread $\sigma(S)$ is reported in each iteration of the two greedy algorithms from $|S| = 1$ to 200. For the standard greedy algorithm for influence maximization InfMax_std we use the implementation provided by [18].

The results for all 12 combinations of datasets and ways of assigning edge probabilities are reported in Figure 6: on the X-axis we report the size of the seed set $|S|$, on the Y-axis we report the expected spread $\sigma(S)$. We can observe the same pattern emerging in all settings: InfMax_std outperforms InfMax_TC in the selection of the first several seeds, but at a certain point, as the seed set size grows, the two curves cross and InfMax_TC starts outperforming the standard method.

Point of saturation analysis. In order to gain more insight into why this happens we study the "point of saturation": i.e., when the greedy algorithm starts seeing all the nodes as indistinguishable w.r.t. the marginal gain. Recall that the general strategy of greedy algorithms is to select the next seed that provides the maximum gain w.r.t. the objective function (expected spread InfMax_std and maximum coverage for InfMax_TC).

More in detail, at an arbitrary iteration j, let MG_i^j denote the node that is ranked i^{th} position for what concerns the marginal gain it will add to the current solution. Any greedy algorithm, so both InfMax_std and InfMax_TC would select MG_1^j. In this test we check the ratio MG_{10}^j/MG_1^j, i.e., we compare how much the marginal gain of the selected node is larger than that of the node which is ranked 10^{th}. This ratio is by definition in $[0,1]$: a ratio closer to 0 means that the selected node is much better than the 10^{th} node in the ranking, while a ratio closer to 1, means that the improvement provided by the selected node is no different than other potential candidates. A ratio close to 1 means that the greedy algorithm can no longer distinguish well among candidates and thus its choice becomes essentially random. At this point we say that the saturation has likely arrived.

Before presenting the results, it is worth noting that running this test is costly. In fact we need to run the standard greedy algorithm with no optimization at all (for instance we cannot use the optimizations in [18]). For this reason we cannot scale. Therefore we report experiments only on the smaller datasets, Twitter-S and NetHEPT-F. Moreover, we start from the 50^{th} iteration of the greedy algorithm and compute the marginal gain ratio for a little more than 30 iterations.

The results of this experiment are shown in Figure 7, where we can observe that, as expected the ratios grow with the iterations, but also that InfMax_std has a ratio already much larger than InfMax_TC at the 50^{th} iteration, and very close to 1 already at the 65^{th} iteration. At this point InfMax_std is already unable to distinguish between the top-10 nodes with the largest marginal gain.

From this perspective, our method has more power to discriminate among interesting nodes, as it reaches its saturation point much later.

Stability analysis. As we already discussed in Section 5, we can define the stability of a seed set $\mathbb{S}$, as the expected cost of its typical cascade C^* exactly as we have done for singleton nodes: the smaller this expected cost the more stable (or reliable) is the seed set. In this last experiment we compare the expected cost of the seed sets selected by InfMax_std and InfMax_TC. These costs are reported in Figure 8 for six datasets.

We can observe that the seed sets selected by InfMax_TC are consistently more stable than the seed sets selected by InfMax_std. In some of the cases the two methods have similar stability at the beginning of the greedy process (small seed sets), but then they start diverging quite early. In other cases (e.g., NetHEPT-W and Slashdot-W) InfMax_std starts by selecting very unstable influential nodes. This confirms one of the motivations behind our work (previously described in Example 2 in Section 1): nodes that have a very high expected spread are not necessarily reliable.

In conclusion, our method not only constantly outperforms the classic (and theoretically optimal) greedy algorithm in terms of expected spread, but the seed sets it produces are also more reliable than those produced by the standard greedy. This could be an important feature, when it comes to real-world deployment.

7. RELATED WORK

To the best of our knowledge, no prior formulation of the problem finding the *sphere of influence* of a node in terms of its typical cascade exists in the literature.

A related line of research studies *reliability* in uncertain graphs [44, 23, 2]. This, for instance, includes finding the probability of connection between two nodes, also known as 2-terminal reliability [8]. Other variants of this problem ask to compute the probability that all nodes are pairwise connected or all nodes in a subset are pairwise connected [19, 34]. However, it has been shown that even the basic problem of computing 2-terminal reliability is #**P**-complete [38]. For this reason, several approximation schemes have been proposed mainly exploiting Monte Carlo sampling methods to estimate connection probabilities. Quite recently, these ideas have been developed to formulate *reliability search* problem in order to efficiently find all nodes reachable from a set of source nodes greater than a probability threshold [25].

Another related line of research is the data-driven vaccination problem: Given a set of already infected people in a population, what are the healthy people who should be immediately given vaccines to best control the epidemic? [43]

For what concerns influence maximization we have already covered the main main background information in the Introduction. The first algorithmic treatment of the problem was provided by Domingos and Richardson [14, 32], who modeled the diffusion process in terms of Markov random fields, and proposed heuristic solutions to the problem. Later, Kempe *et al.* [24] introduced influence maximization as a discrete optimization problem: this is the definition that has been followed in the subsequent literature. As explained previously most of the effort in this area has been devoted to improve the efficiency and scalability of influence maximization [27, 9, 17, 18].

Recently, Borgs et al. [7] proposed a near-linear time randomized algorithm based on the idea of sampling *"reverse-reachable"* (RR) sets in the graph. These ideas were extended to obtain a more practical algorithm – *Two-phase Influence Maximization (TIM)* – by Tang et al. [35]. Cohen et al. [12] proposed a sketch-based design for fast computation of influence spread, achieving efficiency and effectiveness comparable to TIM.

Most of this literature on efficient algorithms for influence maximization assumes the weighted social graph given, and do not address how the link influence probabilities $p_{u,v}$ can be obtained. Saito et al. [33] were the first to study the problem of learning the probabilities for the independent cascade model from a set of past observations, formalizing it as likelihood maximization and applying Expectation Maximization (EM) to solve it. Later Goyal et al. [16] provided a simpler and more scalable frequentist definition.

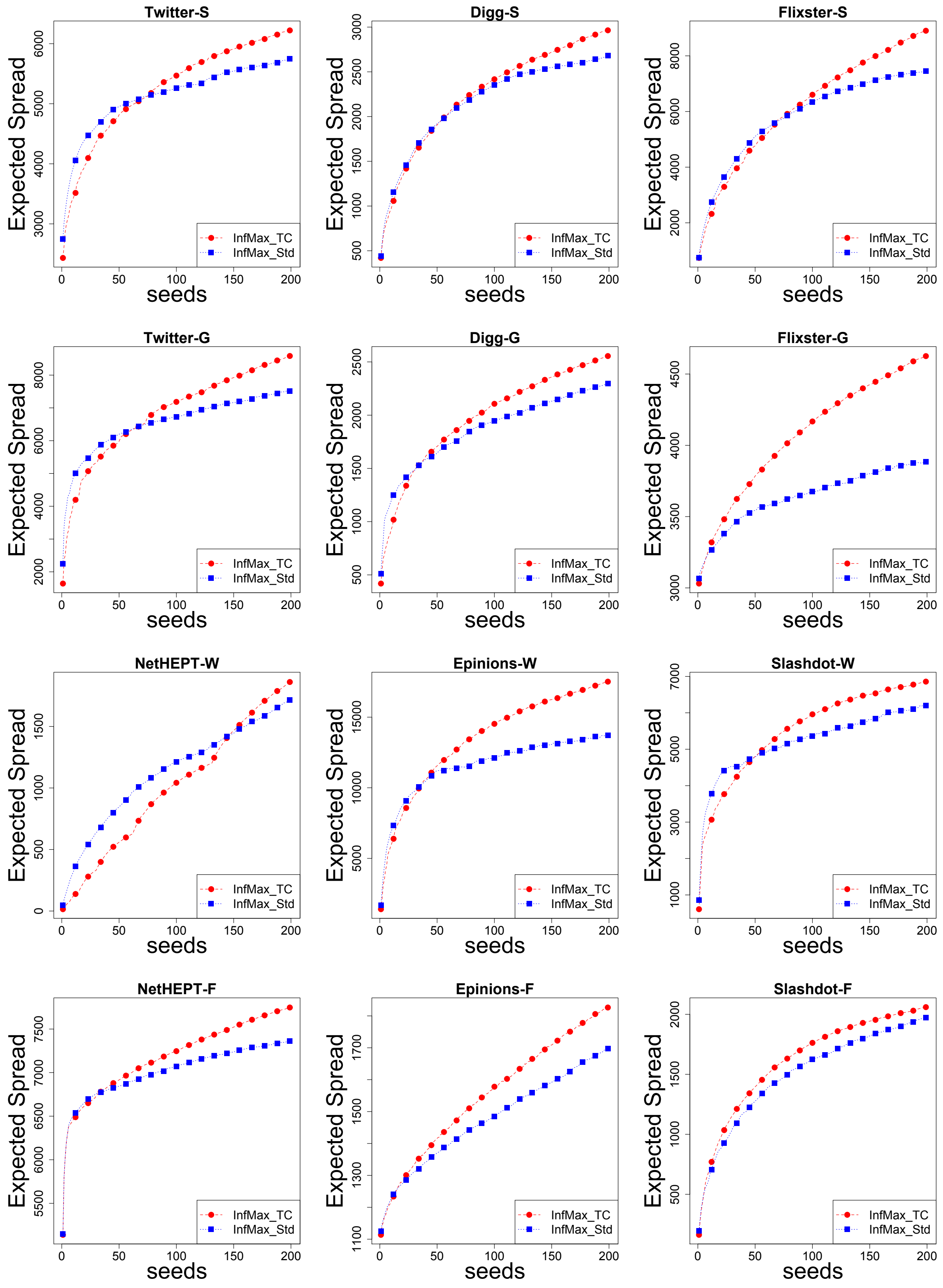


Figure 6: Influence maximization experiments in all 12 settings: on the X-axis we report the size of the seed set $|S|$, on the Y-axis we report the expected spread $\sigma(S)$.

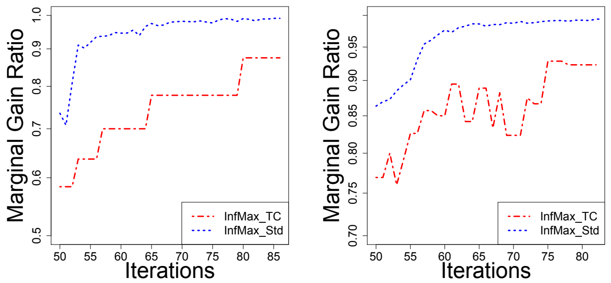


Figure 7: Marginal gain ration provided by seeds in NetHEPT-F (left) and Twitter-S (right).

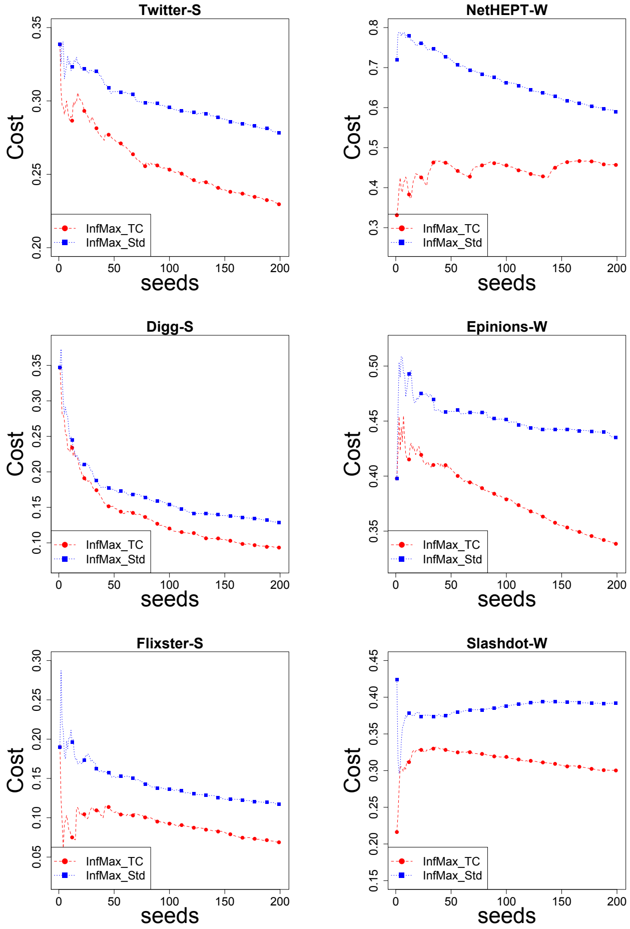


Figure 8: Stability analysis: expected cost of the seed sets extracted by InfMax_std and InfMax_TC over six datasets. By expected cost we mean the expected Jaccard distance between the typical cascade generated by the seed set and 1000 random cascades generated by the same seed set. The smaller this value, the more reliable the behaviour of the seed set.

Following the likelihood maximization of [33], Mathioudakis et al. [29] studied the problem of sparsifying the influence network to a prefixed extent while maximizing the likelihood of generating the propagation traces in the given log. Barbieri et al. [5] use a similar EM approach of [33] to learn *topic-aware* influence probabilities.

Following the frequentist definition of [16] Kutzkov et al. [26] studied the problem of learning influence probabilities in a big data scenario, where the network topology might not fit in memory and there is a continuous stream of actions (e.g., a stream of tweets in Twitter). Following the same definition of Goyal et al. [16], Tassa and Bonchi [37] studied the privacy and security implications of learning influence strength in a social network, and devised secure multiparty protocols.

8. CONCLUSIONS AND FUTURE WORK

In this paper we study the problem of computing the sphere of influence for each node in a social influence network. We formalize this as the Typical Cascade problem over a probabilistic directed graph where each directed edge (u, v) has associated a probability representing the contagion probability, or the strength of influence of u over v. We devise a method based on sampling and computing the Jaccard median of the samples. Then we propose a novel approach to influence maximization based on max-cover applied to the sphere of influence of all nodes in the network.

Our main theoretical result is a bound showing that we can obtain a multiplicative approximation to our problem with a constant number samples, i.e., not dependent on the size of the network (Theorem 2).

Our main practical contribution is the first method for influence maximization outperforming the theoretically optimal greedy algorithm for influence maximization, for large seed sets (Figure 6).

To the best of our knowledge *our work is the first to show consistent improvement in terms of quality over the standard greedy algorithm for influence maximization*, as confirmed by our thorough experimentation using several different benchmark networks and different ways of assigning the influence probabilities to the edges.

Given that the classic greedy algorithm is essentially optimal in terms of quality under standard complexity assumptions, a very interesting line of research for future investigation is to characterize which graph properties of real-world datasets allow the greedy algorithm, either the classic one or our variant based on spheres of influence, to provide better approximations. One may also exploit the gadget for the max-cover greedy algorithm developed in [4] to obtain tight bounds on the optimality gap of the solutions found.

The computations of the sphere of influence might find applications in other contexts, that we plan to investigate in our future research. First of all, remaining in the viral marketing context, having the spheres of influence precomputed and stored in an index might provide a direct solution to several variants of influence maximization. Consider for instance the case where different segments of market (set of users) have different values for a viral marketing campaign. In our setting this is directly achieved by means of a weighted max-cover using the available spheres of influence. Then when the next campaign is run, and the users have different values, we can again reuse the same spheres of influence. Other examples might include viral marketing campaigns under different types of constraints, such as, e.g., when different nodes have different costs to become a seed.

Outside of viral marketing, we can consider application of spheres of influence in contagion problems, or in the vaccination problem [43].

9. REFERENCES

[1] S. Abiteboul, P. Kanellakis, and G. Grahne. On the representation and querying of sets of possible worlds. In *SIGMOD*, 1987.

[2] K. Aggarwal, K. Misra, and J. Gupta. Reliability evaluation a comparative study of different techniques. *Microelectronics Reliability*, 14(1):49–56, 1975.

[3] A. V. Aho, M. R. Garey, and J. D. Ullman. The transitive reduction of a directed graph. *SIAM J. Comput.*, 1(2):131–137, 1972.

[4] R. A. Baeza-Yates, P. Boldi, and F. Chierichetti. Essential web pages are easy to find. In *WWW*, 2015.

[5] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *ICDM*, 2012.

[6] F. Bonchi, F. Gullo, A. Kaltenbrunner, and Y. Volkovich. Core decomposition of uncertain graphs. In *KDD*, 2014.

[7] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, 2014.

[8] T. B. Brecht and C. J. Colbourn. Lower bounds on two-terminal network reliability. *Discrete Applied Mathematics*, 21(3):185–198, 1988.

[9] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, 2010.

[10] F. Chierichetti and R. Kumar. LSH-preserving functions and their applications. *Journal of the ACM*, 62(5):33, 2015.

[11] F. Chierichetti, R. Kumar, S. Pandey, and S. Vassilvitskii. Finding the jaccard median. In *SODA*, 2010.

[12] E. Cohen, D. Delling, T. Pajor, , and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*, 2014.

[13] N. Dalvi and D. Suciu. Efficient query evaluation on probabilistic databases. In *VLDB*, 2004.

[14] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD*, 2001.

[15] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

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

[17] A. Goyal, F. Bonchi, and L. V. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[18] A. Goyal, W. Lu, and L. V. Lakshmanan. CELF++: optimizing the greedy algorithm for influence maximization in social networks. In *WWW*, 2011.

[19] G. Hardy, C. Lucet, and N. Limnios. K-terminal network reliability measures with binary decision diagrams. *IEEE Transactions on Reliability*, 56(3):506–515, 2007.

[20] T. Hogg and K. Lerman. Social dynamics of digg. *EPJ Data Science*, 1(1):1–26, 2012.

[21] M. Jamali. Flixster data set. http://www.cs.ubc.ca/~jamalim/datasets/.

[22] R. Jin, L. Liu, and C. C. Aggarwal. Discovering Highly Reliable Subgraphs in Uncertain Graphs. In *KDD*, 2011.

[23] R. Jin, L. Liu, B. Ding, and H. Wang. Distance-Constraint Reachability Computation in Uncertain Graphs. *PVLDB*, 4(9):551–562, 2011.

[24] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the Spread of Influence through a Social Network. In *KDD*, 2003.

[25] A. Khan, F. Bonchi, A. Gionis, and F. Gullo. Fast reliability search in uncertain graphs. In *EDBT*, 2014.

[26] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis. STRIP: stream learning of influence probabilities. In *KDD*, 2013.

[27] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. M. VanBriesen, and N. S. Glance. Cost-effective outbreak detection in networks. In *KDD*, 2007.

[28] J. Leskovec and A. Krevl. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data.

[29] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. Sparsification of influence networks. In *KDD*, 2011.

[30] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Mathematical Programming*, 14(1):265–294, 1978.

[31] M. Potamias, F. Bonchi, A. Gionis, and G. Kollios. k-Nearest Neighbors in Uncertain Graphs. *PVLDB*, 3(1):997–1008, 2010.

[32] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, 2002.

[33] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *KES*, 2008.

[34] A. R. Sharafat and O. R. Ma'rouzi. All-terminal network reliability using recursive truncation algorithm. *IEEE Transactions on Reliability*, 58(2):338–347, 2009.

[35] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*, 2014.

[36] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM J. Comput.*, 1(2):146–160, 1972.

[37] T. Tassa and F. Bonchi. Privacy preserving estimation of social influence. In *EDBT*, 2014.

[38] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM J. Comput.*, 8(3):410–421, 1979.

[39] D. Watts. Challenging the influentials hypothesis. *WOMMA Measuring Word of Mouth, Volume 3*, pages 201–211, 2007.

[40] D. Watts and P. Dodds. Influential, networks, and public opinion formation. *Journal of Consumer Research*, 34(4):441–458, 2007.

[41] D. Watts and J. Peretti. Viral marketing for the real world. *Harvard Business Review*, pages 22–23, May 2007.

[42] D. B. West. *Introduction to Graph Theory (2nd Edition)*. Prentice Hall, Aug. 2000.

[43] Y. Zhang and B. A. Prakash. Dava: Distributing vaccines over networks under prior information. In *SDM*, 2014.

[44] K. Zhu, W. Zhang, G. Zhu, Y. Zhang, and X. Lin. Bmc: an efficient method to evaluate probabilistic reachability queries. In *Database Systems for Advanced Applications*, pages 434–449. Springer, 2011.

APPENDIX

A. MISSING PROOFS FROM SECTION 3

Before showing the required proofs, we need to establish a couple of technical lemmas. The first gives a relationship between Jaccard distance and size of the union of two sets; the second gives a tight bound on the expectation of the inverse denominators of $d_J(X, C)$.

LEMMA 3. *If $A \cap B \neq \emptyset$, then $|A \cup B| \le \frac{\min(|A|,|B|)}{1-d_J(A,B)}$.*

PROOF. Compute the Jaccard similarity of A and B:

$$J(A,B) = \frac{|A\cap B|}{|A\cup B|} \le \frac{\min(|A|,|B|)}{|A\cup B|} \le \frac{|A|}{|A\cup B|} \le \frac{1}{1+\alpha},$$

so their Jaccard distance satisfies

$$d_J(A,B) = 1 - J(A,B) \ge \frac{\alpha}{1+\alpha}.$$

□

LEMMA 4. *For all X,*

$$\frac{1}{|X|} \ge \mathop{\mathbb{E}}_{C\sim\mathcal{C}}\left[\frac{1}{|X\cup C|}\right] \ge \frac{1-2\sqrt{\rho(X)}}{|X|}.$$

PROOF. The first inequality is obvious.

To show the second, we first prove that, for all $\alpha \in (0,1)$,

$$\Pr\left[\frac{|C|}{|X|} \notin [1-\alpha, 1+\alpha]\right] \le \frac{1+\alpha}{\alpha}\rho(X).$$

To see this, $B(C)$ denote the "bad event" $\frac{|C|}{|X|} \notin [1-\alpha, 1+\alpha]$. Note that $1/(1-\alpha) > 1+\alpha$, so by Lemma 3, $B(C)$ implies $d_J(C,X) \ge \frac{\alpha}{1+\alpha}$. Then

$$\mathbb{E}[d_J(C,X)] \ge \Pr[B(C)] \cdot d_J(C,X),$$

so

$$\Pr[B(C)] \le \frac{\mathbb{E}[d_J(C,X)]}{\alpha/(1+\alpha)} = \frac{1+\alpha}{\alpha}\rho(X).$$

Setting $\epsilon \triangleq \rho(X)$ and $\alpha \triangleq \sqrt{\epsilon}$, we obtain

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{|X\cup C|}\right] &\ge \frac{\Pr[|X\cup C| \le (1+\alpha)|X|]}{(1+\alpha)|X|} \\
&\ge \frac{1-\frac{1+\alpha}{\alpha}\epsilon}{(1+\alpha)|X|} \\
&= \frac{1}{|X|}\left(\frac{1}{1+\alpha} - \frac{\epsilon}{\alpha}\right) \\
&\ge \frac{1}{|X|}\left(1-\alpha-\frac{\epsilon}{\alpha}\right) \\
&= \frac{1}{|X|}(1-2\sqrt{\epsilon}).
\end{aligned}$$

□

A.1 Proof of Lemma 1

PROOF. (a) Since d_J is a metric and the support of $\mathcal{C}$ is nonempty,

$$\begin{aligned}
d_J(Y,Y') &= \mathop{\mathbb{E}}_{C\in\mathcal{C}}[d_J(Y,Y')] \\
&\le \mathop{\mathbb{E}}_{C\in\mathcal{C}}[d_J(Y,C) + d_J(C,Y')] \\
&= \rho(Y) + \rho(Y').
\end{aligned}$$

On the other hand, observe that $|Y \oplus C| + |X \oplus C| \ge |X \oplus Y|$ (the triangle inequality for the Hamming metric), so by Lemma 4,

$$\begin{aligned}
f_X(Y) + \rho(X) &= f_X(Y) + f_X(X) \\
&= \mathbb{E}\left[\frac{|Y\oplus C| + |X \oplus C|}{|X\cup C|}\right] \\
&\ge \mathbb{E}\left[\frac{|X\oplus Y|}{|X\cup C|}\right] \\
&\ge \frac{|X\oplus Y|}{|X|}\left(1-2\sqrt{\rho(X)}\right) \\
&\ge \frac{|X\oplus Y|}{|X\cup Y|}\left(1-2\sqrt{\rho(X)}\right) \\
&= d_J(X,Y)\left(1-2\sqrt{\rho(X)}\right).
\end{aligned}$$

Thus

$$\begin{aligned}
d_J(X,Y) &\le \min\left(1, \frac{f_X(Y)+\rho(X)}{1-2\sqrt{\rho(X)}}\right) \\
&\le (f_X(Y)+\rho(X)) \cdot \min\left(\frac{1}{\rho(X)}, \frac{1}{1-2\sqrt{\rho(X)}}\right) \\
&\le 6\,(f_X(Y)+\rho(X)),
\end{aligned}$$

by an easy case distinction ($\rho(X) \le 1/6$ vs $\rho(X) > 1/6$).

Likewise, $d_J(X,Y') \le 6\,(f_X(Y')+\rho(X))$, so by the triangle inequality,

$$\begin{aligned}
d_J(Y,Y') &\le d_J(Y',X) + d_J(X,Y') \\
&= O(\rho(X) + f_X(Y) + f_X(Y')).
\end{aligned}$$

(b) Let $\tau = d_J(X,Y)$. Using Lemma 3,

$$\begin{aligned}
|X\cup Y\cup C| &\le |X\cup C| + |Y \setminus X| \\
&= |X\cup C| + (|X\cup Y| - |X|) \\
&\le |X\cup C| + \left(\frac{1}{1-\tau}|X| - |X|\right) \\
&= |X\cup C| + \frac{\tau}{1-\tau}|X| \\
&\le |X\cup C|\left(1+\frac{\tau}{1-\tau}\right) \\
&= \frac{|X\cup C|}{1-\tau}.
\end{aligned}$$

Likewise, we have $|X\cup Y\cup C| \le \frac{|Y\cup C|}{1-\tau}$. Thus,

$$|X\cup C|, |Y\cup C| \le \frac{|X\cup C|}{1-\tau}, \frac{|Y\cup C|}{1-\tau}.$$

and

$$(1-\tau)\frac{|Y\oplus C|}{|X\cup C|} \le \frac{|Y\oplus C|}{|Y\cup C|} \le \frac{1}{1-\tau}\frac{|Y\oplus C|}{|X\cup C|}.$$

Taking expectations over C, we obtain

$$(1-\tau)f_X(Y) \le \rho(Y) \le \frac{1}{1-\tau} f_X(Y).$$

(c) By parts a) and b), $\rho(Y) \le \rho(X)$ implies $d_J(X,Y) \le \rho(Y) + \rho(X) \le 2\rho(X)$ and $(1 - d_J(X,Y))f_X(Y) \le \rho(Y) \le \rho(X) = f_X(X)$.

□

A.2 Proof of Lemma 2

We need an auxiliary lemma.

Lemma 5. *Let $\widetilde{z}$ be the average of $t \ge 3$ independent random variables in $[0,1]$ each with expectation $z > 0$. Then*

(a) If $z \le \frac{1}{30}$,

$$\Pr\left[\widetilde{z} > \frac{1}{3}\right] \le \frac{30z}{e^{t/3}}.$$

(b) For all $C > 0$,

$$\Pr\left[\widetilde{z} > z + \sqrt{\frac{C}{t}}\right] \le e^{-2C}.$$

Proof. (a) Let $\delta = \frac{1}{3z} - 1$. By the multiplicative Chernoff bound,

$$\begin{aligned}\Pr\left[\widetilde{z} > \frac{1}{3}\right] &\le \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{zt}\\ &= (e^{1-3z}\cdot 3z)^{t/3}\\ &\le (3ez)\cdot(3ez)^{t/3-1}\\ &\le (3ez)\cdot e^{1-t/3}\\ &= (3e^2 z)\cdot e^{-t/3}\\ &\le 30ze^{-t/3}.\end{aligned}$$

(b) Let $\tau = \sqrt{C/t}$. By the additive Chernoff bound,

$$\Pr\left[\widetilde{z} > z + \tau\right] \le e^{-2t\tau^2} = e^{-2C}.$$

□

We proceed to prove Lemma 2.

Proof. For all $i \in [n]$, define

$$a_i = \mathop{\mathbb{E}}_{C\sim\mathcal{C}} \frac{[i \in Z \oplus C]}{|X \cup C|}, \qquad b_i = \mathop{\mathbb{E}}_{C\sim\mathcal{C}} \frac{[i \notin Z \oplus C]}{|X \cup C|}.$$

By linearity of expectation,

$$\begin{aligned}f_X(Y) &= \mathop{\mathbb{E}}_{c\sim\mathcal{C}}\left[\frac{\sum_{i\in[n]}[i \in Y \oplus C]}{|X\cup C|}\right]\\ &= \sum_{i\in[n]} \mathop{\mathbb{E}}_{c\sim\mathcal{C}} \frac{[i \in Y\oplus C]}{|X \cup C|}\\ &= \sum_{i\in Z\oplus C} a_i + \sum_{i \in Y\oplus Z} b_i\\ &= \sum_{i\notin Y\oplus Z} a_i + \sum_{i\in Y\oplus Z} b_i,\end{aligned}$$

where we used $Y \oplus C = (Z \oplus C) \oplus (Y \oplus Z)$. In particular,

$$f_X(Z) = \sum_{i\in[n]} a_i,$$

and the optimality of Z implies $f_X(Z \oplus \{i\}) \ge f_X(Z)$, i.e., $a_i \le b_i$ for all $i \in [n]$.

Now define the empirical counterparts of a_i, b_i:

$$\widetilde{a}_i = \mathop{\mathbb{E}}_{D\sim\mathcal{D}} \frac{[i \in Z \oplus D]}{|X \cup D|}, \qquad \widetilde{b}_i = \mathop{\mathbb{E}}_{D\sim\mathcal{D}} \frac{[i \notin Z \oplus D]}{|X \cup D|}.$$

Likewise, we have

$$\widetilde{f}_X(Y) = \sum_{i\notin Y\oplus Z} \widetilde{a}_i + \sum_{i\in Y\oplus Z} \widetilde{b}_i.$$

Let $\widetilde{Z}$ minimize $\widetilde{f}_X(Y)$. Then

$$Z \oplus \widetilde{Z} = \{i \in [n] \mid \widetilde{a}_i > \widetilde{b}_i\},$$

so

$$\begin{aligned}f_X(\widetilde{Z}) &= \sum_{i\notin \widetilde{Z}\oplus Z} a_i + \sum_{i\in\widetilde{Z}\oplus Z} b_i\\ &= \sum_{\widetilde{a}_i \le \widetilde{b}_i} a_i + \sum_{\widetilde{a}_i > \widetilde{b}_i} b_i\end{aligned}$$

and

$$f_X(\widetilde{Z}) - f_X(Z) = \sum_{\widetilde{a}_i > \widetilde{b}_i} (b_i - a_i).$$

Note that

$$\mathop{\mathbb{E}}_{C\sim\mathcal{C}}\left[\frac{1}{|X\cup D|}\right] = a_i + b_i \triangleq \alpha$$

and, by Lemma 4,

$$\frac{1-2\sqrt{\rho(X)}}{|X|} \le \mathop{\mathbb{E}}_{D\sim\mathcal{D}}\left[\frac{1}{|X\cup D|}\right] = a_i + b_i \triangleq \widetilde{\alpha}.$$

We set the following parameters for convenience:

$$\lambda \triangleq \sqrt{\frac{3+3\ln(2\ell/\delta)}{\ell}}, \quad \gamma \triangleq |\alpha - \widetilde{\alpha}|, \quad C = 10\log(\ell/\delta).$$

Since $\widetilde{\alpha}$ is the expectation of independent random variables in $[0, 1/|X|]$, we can apply the additive Chernoff bound:

$$\Pr\left[|X|\cdot\gamma > \sqrt{\frac{\ln(4/\delta)}{2\ell}}\right] \le 2\exp\left(-2\ell\frac{\ln(4/\delta)}{2\ell}\right) = \frac{\delta}{2}.$$

In particular, with probability at least $1 - \delta/2$, $\gamma \le \frac{\lambda}{|X|}$ and $\alpha - \gamma \le \widetilde{\alpha} \le \alpha + \gamma$.

Observe that $\widetilde{a}_i$ is the average of ℓ independent random variables in $[0, 1/|X|]$. Since $a_i = 0$ implies $\widetilde{a}_i = 0$ and hence $\widetilde{a}_i \le \widetilde{b}_i$ almost surely, we may assume $a_i > 0$ for all i. Letting $z_i = |X|\cdot a_i$ and using Lemma 5, we conclude that

(a) If $z_i \le \frac{1}{30}$,

$$\Pr\left[|X|\cdot\widetilde{a}_i > \frac{1}{3}\right] \le \frac{30|X|a_i}{e^{\ell/3}}.$$

(b) For all $C > 0$,

$$\Pr\left[|X|\cdot\widetilde{a}_i > \sqrt{\frac{C}{\ell}}\right] \le e^{-2C}.$$

Write

$$B_1 = \left\{ i \in [n] \mid z_i \le \frac{1}{30} \wedge \widetilde{b}_i \ge \widetilde{a}_i \right\},$$

$$B_2 = \left\{ i \in [n] \mid z_i > \frac{1}{30} \wedge b_i \ge a_i + \sqrt{\frac{C}{\ell}} \right\},$$

$$A = [n] \setminus (B_1 \cup B_2).$$

Note that for small enough λ, the conditions $\widetilde{b}_i \ge \widetilde{a}_i$ and $|\alpha - \widetilde{\alpha}| \le \lambda$ imply $z_i > 1/30$. By the above,

$$\begin{aligned}
\mathbb{E}\left[\sum_{i \in B_1} (b_i - a_i) \right] &\le \frac{1}{|X|}\,\mathbb{E}[|B_1|] \\
&= \frac{1}{|X|} \sum_{i \mid z_i \le 1/30} \Pr[i \in B_1] \\
&\le \frac{30|X|}{|X| e^{\ell/3}} \sum_{z_i \le 1/30} a_i \\
&= \frac{30}{e^{\ell/3}} f_X(Z) \\
&\le O(\delta \cdot \lambda f_X(Z))
\end{aligned}$$

by our choice of λ, so Markov's inequality implies that with probability at least $1 - \delta/4$ we have

$$\sum_{i \in B_1} (b_i - a_i) \le O(\lambda f_X(Z)).$$

Likewise,

$$\begin{aligned}
\mathbb{E}\left[\sum_{i \in B_2} (b_i - a_i) \right] &\le \frac{1}{|X|}\,\mathbb{E}[|B_2|] \\
&= \frac{1}{|X|} \sum_{z_i > 1/30} \Pr[i \in B_2] \\
&\le \frac{e^{-2C}}{|X|} \left| \left\{ i \in [n] \mid a_i > \frac{1}{30|X|} \right\} \right|
\end{aligned}$$

The size of the set $B_3 = \{i \in [n] \mid a_i > \frac{1}{30|X|}\}$ is at most $30|X| \sum_{i \in B_3} a_i \le 30|X| f_X(Z)$, therefore with probability at least $1 - \delta/4$ we have

$$\sum_{i \in B_2} (b_i - a_i) \le \frac{120 e^{-2C} f_X(Z)}{\delta} = O(\lambda f_X(Z)).$$

Finally, notice that

$$\begin{aligned}
\sum_{i \in A \wedge \widetilde{a}_i < \widetilde{b}_i} (b_i - a_i) &\le \sum_{a_i \ge 1/30} \sqrt{\frac{C}{\ell}} \\
&\le 30\sqrt{\frac{C}{\ell}} \sum_i a_i \\
&= 30\sqrt{\frac{C}{\ell}} f_X(Z) \\
&\le O(\lambda f_X(Z)).
\end{aligned}$$

Therefore with probability at least $1 - \delta$,

$$f_X(\widetilde{Z}) \le (1 + O(\lambda))\, f_X(Z).$$

This establishes the first part of the theorem. To prove the second, suppose that $\widetilde{f}_X(Y) \le (1+\beta)\widetilde{f}_X(\widetilde{Z})$. Observe that

$$f_X(\widetilde{Y}) - f_X(Z) = \sum_{i \in \widetilde{Y} \oplus Z} b_i - a_i.$$

and

$$\widetilde{f}_X(\widetilde{Y}) - \widetilde{f}_X(\widetilde{Z}) = \sum_{i \in \widetilde{Y} \oplus \widetilde{Z}} \widetilde{b}_i - \widetilde{a}_i \le \beta \widetilde{f}_X(\widetilde{Z}).$$

If we split the set $T = \widetilde{Y} \oplus \widetilde{Z}$ into $T \cap B_1, T \cap B_2$ and $T \cap A$ and apply our bounds for the size of B_1 and B_2, we conclude that with probability $1 - \delta$,

$$\sum_{i \in \widetilde{Y} \oplus \widetilde{Z}} (b_i - a_i) \le O(\lambda) \cdot \widetilde{f}_X(\widetilde{Z}) + \sum_{i \in T \cap A, z_i \le 1/30, \widetilde{b}_i \le \widetilde{a}_i} (b_i - a_i)$$

and

$$\begin{aligned}
\sum_{i \in T \cap A, z_i \le 1/30, \widetilde{b}_i \le \widetilde{a}_i} (b_i - a_i) &= \sum_{i \in T \cap A, z_i \le 1/30, \widetilde{b}_i \le \widetilde{a}_i} O(\widetilde{b}_i - \widetilde{a}_i) \\
&= O(\beta f_X(\widetilde{Z})).
\end{aligned}$$

Thus

$$\begin{aligned}
f_X(\widetilde{Y}) - f_X(Z) &\le |f_X(\widetilde{Z}) - f_X(Z)| + \sum_{i \in \widetilde{Y} \oplus \widetilde{Z}} (b_i - a_i) \\
&= O(\beta + \lambda) f_X(Z).
\end{aligned}$$

□