Network Inference and Influence Maximization from Samples

Wei Chen [1] **Xiaoming Sun** [2 3] **Jialin Zhang** [2 3] **Zhijie Zhang** [2 3]

Abstract

Influence maximization is the task of selecting a small number of seed nodes in a social network to maximize the spread of the influence from these seeds, and it has been widely investigated in the past two decades. In the canonical setting, the whole social network as well as its diffusion parameters is given as input. In this paper, we consider the more realistic sampling setting where the network is unknown and we only have a set of passively observed cascades that record the set of activated nodes at each diffusion step. We study the task of influence maximization from these cascade samples (IMS), and present constant approximation algorithms for this task under mild conditions on the seed set distribution. To achieve the optimization goal, we also provide a novel solution to the network inference problem, that is, learning diffusion parameters and the network structure from the cascade data. Comparing with prior solutions, our network inference algorithm requires weaker assumptions and does not rely on maximum-likelihood estimation and convex programming. Our IMS algorithms enhance the learning-and-then-optimization approach by allowing a constant approximation ratio even when the diffusion parameters are hard to learn, and we do not need any assumption related to the network structure or diffusion parameters.

1. Introduction

Maximizing the spread of influence through a social network has been widely studied in the past two decades. It models the phenomenon in which a small set of initially *active* nodes called *seeds* takes some piece of information (news, ideas or opinions, etc.), and the information spreads over the network to *activate* the remaining nodes. The expected number of *final* active nodes is called the *influence spread* of the seed set. The *influence maximization* problem asks to pick at most k seeds in order to maximize the influence spread. Under many *diffusion models* such as the discrete time *independent cascade* (IC) model and *linear threshold* (LT) model (Kempe et al., 2003), the problem enjoys a $(1 - 1/e - \varepsilon)$-approximation (with small $\varepsilon > 0$), which is tight assuming P $\neq$ NP (Feige, 1998). It has found applications in many scenarios.

Traditional influence maximization problem requires as input the whole social network (as well as its parameters), based on which one can compute or estimate the influence spread function. In many scenarios, however, this might be too demanding, especially for those who do not have free access to the network. In this work, we consider influence maximization in the sampling setting where one only has access to a set of passively observed cascades spreading over an implicit social network. Each cascade records the set of activated nodes at each time step. Such sample data is available in many scenarios, especially on the Internet where the timestamps can be recorded in principle. We are interested in whether we can maximize the influence from such sample data. We model this problem as *influence maximization from samples* below:

Influence maximization from samples (IMS). *For an unknown social network G with IC model parameters, given t cascade samples where each seed is independently sampled with an unknown probability, can we find a seed set of size at most k such that its influence spread is a constant approximation of the optimal seed set, when t is polynomial to the size of G?*

En route to solving the above problem, a natural and reasonable approach is to first learn the network structure as well as its parameters, and then maximize the influence over the learned network. This leads to the well-studied *network inference* problem below.

Network Inference. *For an unknown social network G, given polynomial number of cascade samples where each seed is sampled independently with an unknown probability, estimate all IC model parameters such that with probability*

[1]Microsoft Research Asia, Beijing, China. [2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. [3]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China.. Correspondence to: Wei Chen <weic@microsoft.com>, Xiaoming Sun, Jialin Zhang, Zhijie Zhang <{sunxiaoming, zhangjialin, zhangzhijie}@ict.ac.cn>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

at least $1-\delta$, every parameter is estimated within a small additive error ε.

Our contributions in this work are mainly two fold. First, we revisit the network inference problem and design a brand new algorithm for it. While all previous algorithms are based on the maximum likelihood estimation and convex programming (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015; Pouget-Abadie & Horel, 2015), our algorithm builds on a closed-form expression for each edge probability in terms of quantities which can be efficiently estimated. As a result, our algorithm enjoys faster implementation, lower sample complexity and weaker assumptions comparing to previous algorithms. Our assumptions are also easy to verify from cascade samples. We will discuss these differences further in the end of Section 3.

Second, we provide several end-to-end IMS algorithms with constant approximation guarantee. Following the canonical learning-and-then-optimization framework, we first present an IMS algorithm by directly invoking our network inference algorithm. The algorithm thus needs the assumptions used for learning. Next, we present alternative algorithms which only need two simple assumptions on the seed distribution and impose no requirements on the underlying network. In contrast, all the known algorithms for network inference (including ours) impose some restrictions on the network. This result is highly non-trivial since it is impossible to resolve network inference problem on arbitrary graphs and hence the learning-and-then-optimization framework fails in this case. For instance, consider a complete graph and another graph with one edge removed from the complete graph, where all edge probabilities are 1. If each node is picked as a seed independently with probability $1/2$, one cannot distinguish them within polynomially many cascade samples. Our IMS follows the general optimization-from-samples framework (Balkanski et al., 2017b), and generalizes the recent result on optimization from structured samples for coverage functions (Chen et al.) (see the discussion below). Finally, we remark that our results not only apply to influence maximization, but also to other learning and optimization settings such as probabilistic maximum cover with application in online advertising (Chen et al., 2016).

1.1. Related Work

Influence maximization from samples follows the framework of *optimization from samples (OPS)* originally proposed by Balkanski et al. (2017b): given a set of polynomial number of samples $\{S_i, f(S_i)\}_{i=1}^t$ and constraint $\mathcal{M}$, can we find a set $S \in \mathcal{M}$ such that $f(S) \geq c \cdot \max_{T\in\mathcal{M}} f(T)$ for some constant c? The OPS framework is very important for the data-driven integration of learning and optimization where the underlying model (function f above) is not readily known. Surprisingly, Balkanski et al. (2017b) show that even for the maximum coverage problem, there is no constant approximation algorithm under the OPS model, despite prior results that a coverage function f is learnable from samples (Badanidiyuru et al., 2012) and constant optimization is available when f is known. Subsequently, several attempts (Balkanski et al., 2016; 2017a; Rosenfeld et al., 2018; Chen et al.) have been made to circumvent the impossibility result of (Balkanski et al., 2017b). Among them the most related one is the *optimization from structured samples* (OPSS) model for coverage functions (Chen et al.), where the samples carry additional structural information in the form of $\{S_i, N(S_i)\}_{i=1}^t$, where $N(S_i)$ contains the nodes covered by S_i. It was shown that if the samples are generated from a *"negatively correlated"* distribution, the maximum coverage problem enjoys constant approximation in the OPSS model. Recall that coverage functions can be regarded as influence spread functions defined over a bipartite graph with edge probabilities in $\{0, 1\}$. Thus, our result on IMS greatly generalizes OPSS to allow general graphs and stochastic cascades over edges.

End-to-end influence maximization from data has been explored by Goyal et al. (2011), but they only used a heuristic method to learn influence spread functions and then used the greedy method for influence maximization, so there was no end-to-end guarantee on IMS. A recent work (Balkanski et al., 2017a) revisited IMS problem under OPS model, and provided a constant approximation algorithm when the underlying network is generated from the stochastic block model. Our study is the first to provide IMS algorithms with theoretical guarantees that work on arbitrary networks.

Network inference has been extensively studied over the past decade (Gomez-Rodriguez et al., 2010; Myers & Leskovec, 2010; Gomez-Rodriguez et al., 2011; Du et al., 2012; Netrapalli & Sanghavi, 2012; Abrahao et al., 2013; Daneshmand et al.; Du et al., 2013; 2014; Narasimhan et al., 2015; Pouget-Abadie & Horel, 2015; He et al., 2016). While most of them focused on the continuous time diffusion model, there are several results under the discrete time IC model (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015; Pouget-Abadie & Horel, 2015), all of which build on the maximum likelihood estimation. We will compare these results with ours after we present our approach.

1.2. Organization

In Section 2, we describe the model, some concepts and notations as well as two Chernoff-type lemmas used in the analysis. In Section 3, we present an algorithm for network inference, i.e. estimating edge probabilities from samples, which can be adapted to recover the network structure. In Section 4, we present algorithms for influence maximization from samples. Finally, we conclude the paper in Section 5.

2. Preliminaries

Social network, IC model and influence maximization. A social network is modeled as a weighted directed graph $G = (V, E, p)$, where V is the set of $|V| = n$ nodes and E is the set of directed edges. Each edge $(u, v) \in E$ is associated with a weight or probability $p_{uv} \in [0, 1]$. For convenience, we assume that $p_{uv} = 0$ if $(u, v) \notin E$ and $p_{uv} > 0$ otherwise. We also use $N^{\text{in}}(v)$ to denote the in-neighbors of node $v \in V$.

The information or influence propagates through the network in discrete time steps. Each node $v \in V$ is either *active* or *inactive*, indicating whether it receives the information. Denote by $S_t \subseteq V$ the set of active nodes at time step t. The nodes in the set S_0 at time step 0 are called *seeds*. The diffusion is assumed to be *progressive*, which means a node will remain active once it is activated. So for all $t \geq 1$, $S_{t-1} \subseteq S_t$.

Given a set of seeds S_0, the *independent cascade* (IC) model describes how the information propagates and S_t is generated for each $t \geq 1$. At time step t, first let $S_t = S_{t-1}$. Next for each node $v \notin S_{t-1}$, each node $u \in N^{\text{in}}(v) \cap (S_{t-1} \setminus S_{t-2})$ will try to activate v *independently* with probability p_{uv} (denote $S_{-1} = \emptyset$). Thus, v becomes active with probability $1 - \prod_{u \in N^{in}(v) \cap (S_{t-1} \setminus S_{t-2})}(1 - p_{uv})$ at this step. Once activated, v will be added into S_t. The propagation terminates when at the end of some time step t, $S_t = S_{t-1}$. Clearly, the process proceeds in at most $n - 1$ time steps and we use $(S_0, S_1, \cdots, S_{n-1})$ to denote the random sequence of the active nodes.

Let $\Phi(S_0) = S_{n-1}$ be the final active set given the seed set S_0, whose expected size is denoted by $\mathbb{E}[|\Phi(S_0)|]$ and is commonly called the *influence spread* of seed set S_0. *Influence maximization* asks to find a set of at most k seeds so as to maximize the influence spread of the chosen seed set. Formally, define the *influence spread function* $\sigma : 2^V \to \mathbb{R}_{\geq 0}$ such that $\sigma(S) = \mathbb{E}[|\Phi(S)|]$ for any $S \subseteq V$. Given a positive integer $k \leq n$, influence maximization corresponds to the following problem $\operatorname{argmax}_{S \subseteq V, |S| \leq k} \sigma(S)$.

The sampling setting. Standard influence maximization problem takes as input the social network $G = (V, E, p)$, based on which one can compute or estimate the influence spread function σ. In this paper, we consider the problem in the sampling setting where G is not given explicitly.

A *cascade* refers to a realization of the sequence of the active nodes $(S_0, S_1, \cdots, S_{n-1})$. By slightly abusing the notation, we still denote the cascade by $(S_0, S_1, \cdots, S_{n-1})$. In the sampling setting, a set of t *independent* cascades $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^t$ is given as input, where the seed set $S_{i,0}$ in cascade i is generated *independently* from a *seed set distribution* $\mathcal{D}$ over the node sets. Throughout this work, we assume that $\mathcal{D}$ is a product distribution where each node $u \in V$ is drawn *independently* with probability q_u. We aim to solve the following two problems.

1. **Network inference**[1]**.** Given a set of t samples $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^t$ defined as above, estimate the values of p_{uv} within an additive error. More formally, for some $\varepsilon \in (0, 1)$, compute a vector $\hat{p}$ such that $|\hat{p}_{uv} - p_{uv}| \leq \varepsilon$ for all $u, v \in V$.
2. **Influence maximization from samples (IMS).** Given a set of t samples $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^t$ defined as above, find a set S^A of at most k seeds such that $\sigma(S^A) \geq \kappa \cdot \sigma(S^*)$ for some constant $\kappa \in (0, 1)$, where S^* denotes the optimal solution.

Notations. Our algorithms actually only use $S_{i,0}$ and $S_{i,1}$ in those cascades to infer information about the graph, and we find it convenient to define some corresponding concepts and notations in advance. These concepts are indeed crucial to our algorithm design. For $v \in V$, we denote by $ap_{G,\mathcal{D}}(v)$ the activation probability of node v *in one time step* during a cascade $(S_0, S_1, \cdots, S_{n-1})$ on graph G starting with a random seed S_0 drawn from the distribution $\mathcal{D}$. Thus, $ap_{G,\mathcal{D}}(v) = \Pr_{G,\mathcal{D}}[v \in S_1]$. Note that it contains the possibility that v itself is a seed, i.e. $v \in S_0 \subseteq S_1$. For $u, v \in V$, we define $ap_{G,\mathcal{D}}(v \mid u) = \Pr_{G,\mathcal{D}}[v \in S_1 \mid u \in S_0]$ and $ap_{G,\mathcal{D}}(v \mid \bar{u}) = \Pr_{G,\mathcal{D}}[v \in S_1 \mid u \notin S_0]$, respectively, which are the corresponding probabilities conditioned on whether u is selected as a seed. When the context is clear, we often omit the subscripts G and $\mathcal{D}$ in the notation.

Chernoff-type bounds. Following are Chernoff-type bounds we will use in our analysis.

Lemma 1 (Multiplicative Chernoff bound, (Mitzenmacher & Upfal, 2005))**.** *Let $X_1, X_2, \cdots, X_n$ be independent random variables in $\{0, 1\}$ with $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \sum_{i=1}^n p_i$. Then, for $0 < a < 1$,*

$$\Pr[X \geq (1 + a)\mu] \leq e^{-\mu a^2/3},$$

and

$$\Pr[X \leq (1 - a)\mu] \leq e^{-\mu a^2/2}.$$

Lemma 2 (Additive Chernoff bound, (Alon & Spencer, 2008; Bansal & Sviridenko, 2006))**.** *Let $X_1, \cdots, X_n$ be independent random variables in $\{0, 1\}$ with $\Pr[X_i = 1] = p_i$. Let $X = \sum_{i=1}^n X_i$ and $\mu = \sum_{i=1}^n p_i$. Then for any $a > 0$, we have*

$$\Pr[X - \mu \geq a] \leq \exp(-a \min(1/5, a/4\mu)).$$

Moreover, for any $a > 0$, we have

$$\Pr[X - \mu \leq -a] \leq \exp(-a^2/2\mu).$$

[1] In the literature, network inference often means to recover network structure, i.e. the edge set E. Here we slightly abuse the terminology to also mean learning edge parameters.

3. Network Inference

In this section, we present a novel algorithm for estimating the edge probabilities of the underlying graph G, i.e. we need to find an estimate $\hat{p}$ of p such that $|\hat{p}_{uv} - p_{uv}| \le \varepsilon$ for all $u, v \in V$. While all previous studies rely on maximum likelihood estimation to estimate $\hat{p}$ (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015; Pouget-Abadie & Horel, 2015), our algorithm is based on the following key observation on the connection between p_{uv} and the one-step activation probabilities $ap(v)$ and $ap(v \,|\, \bar{u})$. We remark that our algorithm does not rely on the knowledge of edges in graph G, and in fact it can be used to also reconstruct the edges of the graph.

Lemma 3. *For any $u, v \in V$ with $u \neq v$,*

$$p_{uv} = \frac{ap(v) - ap(v \,|\, \bar{u})}{q_u(1 - ap(v \,|\, \bar{u}))}.$$

Proof. To avoid confusion, we write the underlying graph G and the seed distribution $\mathcal{D}$ explicitly in notation $ap(\cdot)$, i.e. $ap(v) = ap_{G,\mathcal{D}}(v)$. Consider the subgraph $G' = G \setminus \{u\}$ by removing node u. Node v has two chances to be activated *in one time step*: either by nodes in G' (including the case where v itself is a seed) or by node u. Since $\mathcal{D}$ is a product distribution, we have

$$ap_{G,\mathcal{D}}(v) = ap_{G',\mathcal{D}}(v) + (1 - ap_{G',\mathcal{D}}(v))q_u p_{uv}.$$

Besides, $ap_{G',\mathcal{D}}(v) = ap_{G,\mathcal{D}}(v \,|\, \bar{u})$ since when considering one-step activation of v, node u not being the seed is equivalent to removing it from the graph. Plugging the equality into the last one, we obtain

$$p_{uv} = \frac{ap_{G,\mathcal{D}}(v) - ap_{G,\mathcal{D}}(v \,|\, \bar{u})}{q_u(1 - ap_{G,\mathcal{D}}(v \,|\, \bar{u}))},$$

which proves the lemma. □

Equipped with the lemma, we are able to estimate p_{uv} by estimating q_u, $ap(v)$ and $ap(v \,|\, \bar{u})$ respectively from cascade samples. Let $t_u = |\{i \in [t] \mid u \in S_{i,0}\}|$ be the number of cascades where u is a seed, $t_{\bar{u}} = |\{i \in [t] \mid u \notin S_{i,0}\}|$ the number of cascades where u is not a seed, $t^v = |\{i \in [t] \mid v \in S_{i,1}\}|$ the number of cascades where v is activated in one time step and $t^v_{\bar{u}} = |\{i \in [t] \mid u \notin S_{i,0}, v \in S_{i,1}\}|$ the number of cascades where u is not a seed and v is activated in one time step. Then, $\hat{q}_u = t_u/t, \widehat{ap}(v) = t^v/t$ and $\widehat{ap}(v \,|\, \bar{u}) = t^v_{\bar{u}}/t_{\bar{u}}$ are good estimates of $q_u, ap(v)$ and $ap(v \,|\, \bar{u})$, respectively. The formal procedure is formulated as Algorithm 1.

Algorithm 1 needs to work under Assumption 1 below, which ensures that all quantities are well estimated. Assumption 1 consists of two conditions. The first means that node $v \in V$ has a non-negligible probability of not being activated in one time step. The second means that the probability of a node $u \in V$ being selected as a seed is neither too low nor too high.

Algorithm 1 Estimate Edge Probabilities

```
Input: A set of cascades $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^t$.
Output: $\{\hat{p}_{uv}\}_{u,v \in V}$ such that $|\hat{p}_{uv} - p_{uv}| \le \varepsilon$ for all $u, v \in V$.
for each $u \in V$ do
    Compute $\hat{q}_u = t_u/t$, where $t_u = |\{i \in [t] \mid u \in S_{i,0}\}|$.
end for
for each $v \in V$ do
    Compute $\widehat{ap}(v) = t^v/t$, where $t^v = |\{i \in [t] \mid v \in S_{i,1}\}|$.
end for
for each $v \in V$ do
    for each $u \in V$ do
        Compute $\widehat{ap}(v \,|\, \bar{u}) = t^v_{\bar{u}}/t_{\bar{u}}$, where $t_{\bar{u}} = |\{i \in [t] \mid u \notin S_{i,0}\}|$ and $t^v_{\bar{u}} = |\{i \in [t] \mid u \notin S_{i,0}, v \in S_{i,1}\}|$.
        Let $\hat{p}_{uv} = \frac{\widehat{ap}(v) - \widehat{ap}(v \,|\, \bar{u})}{\hat{q}_u(1 - \widehat{ap}(v \,|\, \bar{u}))}$.
    end for
end for
return $\{\hat{p}_{uv}\}_{u,v \in V}$.
```

Assumption 1 (Assumptions for Edge Probabilities Estimation)**.** *For some parameters $\alpha \in (0, 1], \gamma \in (0, 1/2]$,*

1. *$ap(v) \le 1 - \alpha$ for all $v \in V$.*
2. *$\gamma \le q_u \le 1 - \gamma$ for all $u \in V$.*

We now give an analysis of Algorithm 1. The proof of Lemma 4 is presented in the appendix.

Lemma 4. *Under Assumption 1, for any $\eta \in (0, 4/5), \delta \in (0, 1)$, for $\hat{q}_u$, $\widehat{ap}(v)$, and $\widehat{ap}(v \,|\, \bar{u})$ defined in Algorithm 1, if the number of samples $t \ge \frac{16}{\gamma\eta^2} \ln \frac{12n}{\delta}$, with probability at least $1 - \delta$, we have*

1. *$|\hat{q}_u - q_u| \le \eta q_u$ for all $u \in V$,*
2. *$|\widehat{ap}(v) - ap(v)| \le \eta$ for all $v \in V$,*
3. *$|\widehat{ap}(v \,|\, \bar{u}) - ap(v \,|\, \bar{u})| \le \eta$ for all $u, v \in V$.*

Theorem 1. *Under Assumption 1, for any $\varepsilon, \delta \in (0, 1)$, let $\eta = \varepsilon\alpha\gamma/4 < 1/8$, and $\{\hat{p}_{uv}\}_{u,v \in V}$ be the set of edge probabilities returned by Algorithm 1. If the number of cascades $t \ge \frac{16}{\gamma\eta^2} \ln \frac{12n}{\delta} = \frac{256}{\varepsilon^2\alpha^2\gamma^3} \ln \frac{12n}{\delta}$, with probability at least $1 - \delta$, for any $u, v \in V$, $|\hat{p}_{uv} - p_{uv}| \le \varepsilon$.*

Proof. With probability at least $1 - \delta$, all the events in Lemma 4 occur. We assume that this is exactly the case in the following. Since $ap(v\,|\,\bar{u}) \le ap(v) \le 1-\alpha$, we have $1 - ap(v\,|\,\bar{u}) \ge \alpha$. By the value of η and the assumption that $q_u \ge \gamma$, we have

$$\eta \le \frac{\varepsilon\gamma}{4}(1-ap(v\,|\,\bar{u})) \le \frac{\varepsilon}{4}q_u(1-ap(v\,|\,\bar{u})). \tag{1}$$

To prove $\hat{p}_{uv} \le p_{uv} + \varepsilon$, we have

$$\begin{aligned}\hat{p}_{uv} &= \frac{\widehat{ap}(v) - \widehat{ap}(v\,|\,\bar{u})}{\hat{q}_u(1-\widehat{ap}(v\,|\,\bar{u}))}\\ &\le \frac{ap(v) - ap(v\,|\,\bar{u}) + 2\eta}{(1-\eta)q_u(1-ap(v\,|\,\bar{u})-\eta)}\\ &\le \frac{ap(v) - ap(v\,|\,\bar{u}) + 2\eta}{(1-\eta)(1-\varepsilon\gamma/4)q_u(1-ap(v\,|\,\bar{u}))}\\ &\le \frac{p_{uv}+\varepsilon/2}{(1-\eta)(1-\varepsilon\gamma/4)} \le p_{uv}+\varepsilon. \end{aligned} \tag{2}$$

The first inequality holds due to Lemma 4. The second inequality holds by applying the first inequality in Eq. (1). The third inequality holds due to Lemma 3 and the second inequality in Eq. (1). To see the correctness of the last inequality, first observe that

$$\begin{aligned}&(p_{uv}+\varepsilon)(1-\eta)(1-\varepsilon\gamma/4)\\ &\ge (p_{uv}+\varepsilon)(1-\eta-\varepsilon\gamma/4)\\ &\ge (p_{uv}+\varepsilon) - (1+\epsilon)(\eta+\varepsilon\gamma/4).\end{aligned}$$

Next, note that

$$(1+\epsilon)(\eta+\varepsilon\gamma/4) = (1+\epsilon)(1+\alpha)\varepsilon\gamma/4 \le (1+\varepsilon)\varepsilon/4 \le \varepsilon/2.$$

The equality is due to the definition of η. The two inequalities hold since $\alpha \in (0,1], \gamma \in (0,1/2]$ and $\varepsilon \in (0,1)$, respectively. Combining the above two observations, we have the desired inequality

$$(p_{uv}+\varepsilon)(1-\eta)(1-\varepsilon\gamma/4) \ge (p_{uv}+\varepsilon)-\varepsilon/2 = p_{uv}+\varepsilon/2.$$

On the other hand, to prove $\hat{p}_{uv} \ge p_{uv} - \varepsilon$, first assume that $p_{uv} \ge \varepsilon$, since otherwise the claim would be trivial for $\hat{p}_{uv} \ge 0$. We now have

$$\begin{aligned}\hat{p}_{uv} &= \frac{\widehat{ap}(v) - \widehat{ap}(v\,|\,\bar{u})}{\hat{q}_u(1-\widehat{ap}(v\,|\,\bar{u}))}\\ &\ge \frac{ap(v) - ap(v\,|\,\bar{u}) - 2\eta}{(1+\eta)q_u(1-ap(v\,|\,\bar{u})+\eta)}\\ &\ge \frac{ap_G(v) - ap(v\,|\,\bar{u}) - 2\eta}{(1+\eta)(1+\varepsilon\gamma/4)q_u(1-ap(v\,|\,\bar{u}))}\\ &\ge \frac{p_{uv}-\varepsilon/2}{(1+\eta)(1+\varepsilon\gamma/4)} \ge p_{uv}-\varepsilon. \end{aligned}$$

The first inequality holds due to Lemma 4. The second inequality holds by applying the first inequality in Eq. (1).

```
Algorithm 2 Recover Network Structure
Input: A set of cascades $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t}$, parameter $\beta \in (0,1)$.
Output: An estimated edge set $\hat{E}$.
  $\{\hat{p}_{uv}\}_{u,v\in V}$ = Estimate-Edge-Probabilities
      $((S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t})$. { With estimation accuracy $\beta/2$.}
  return $\hat{E} = \{(u,v) \mid \hat{p}_{uv} > \beta/2\}$.
```

The third inequality holds due to Lemma 3 and the second inequality in Eq. (1). The last inequality follows from a similar argument as the one for the last inequality in Eq.(2), and we omit it for conciseness. □

With the ability of estimating edge probabilities, we further show that we can recover the graph structure by a standard threshold approach (Netrapalli & Sanghavi, 2012; Pouget-Abadie & Horel, 2015). The formal procedure is depicted as Algorithm 2, which estimates the edge probabilities to a prescribed accuracy and returns the edges whose estimated probabilities are above a prescribed threshold. Its guarantee is shown in Theorem 2, which shows that no "zero-probability edge" is incorrectly recognized as an edge. Besides, only small-probability edges are omitted, which is reasonable for practical use.

Theorem 2. *Under Assumption 1, if the number of cascades* $t \ge \frac{1024}{\alpha^2\beta^2\gamma^3}\ln\frac{4n}{\delta}$*, with probability at least* $1-\delta$*, the edge set* $\hat{E}$ *returned by Algorithm 2 satisfies (1)* $\hat{E} \subseteq E$*, and (2) if* $p_{uv} > \beta$*, then* $(u,v) \in \hat{E}$*. As a corollary, if* $p_{uv} > \beta$ *for all* $(u,v) \in E$*, then* $\hat{E} = E$*.*

Proof. By Theorem 1, $|\hat{p}_{uv} - p_{uv}| \le \varepsilon = \beta/2$ for all $u, v \in V$ with probability at least $1-\delta$. If $(u,v) \notin E$, then $p_{uv} = 0$ and hence $\hat{p}_{uv} \le \beta/2$, which implies $(u,v) \notin \hat{E}$. Thus, $\hat{E} \subseteq E$. If $(u,v) \in E$ and $p_{uv} > \beta$. Then, $\hat{p}_{uv} \ge p_{uv} - \beta/2 > \beta/2$ and hence $(u,v) \in \hat{E}$. Finally, if $p_{uv} > \beta$ for all $(u,v) \in E$, then $E \subseteq \hat{E}$ and hence $\hat{E} = E$, which concludes the proof. □

Discussion. It is worth comparing the result in (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015; Pouget-Abadie & Horel, 2015) with ours, since all of them studied network inference under the IC model. Specifically, Netrapalli & Sanghavi (2012) initiated the study of recovering network structure and did not consider the estimation of edge parameters. Narasimhan et al. (2015); Pouget-Abadie & Horel (2015) studied how to estimate edge parameters. Both of them used the Euclidean norm of the edge probability vector as the measurement of accuracy, while we use the infinite norm and therefore our estimation is accurate for every single edge. Besides, in (Narasimhan et al., 2015), it was additionally assumed that the network structure is known in advance. In (Pouget-Abadie & Horel, 2015), totally different assumptions were used, which seems incomparable to ours, and thus we will not further compare against it below.

There are several important differences besides the above. First, the approaches taken are different. All the algorithms in the previous works build on the maximum likelihood estimation (MLE) and require to solve a convex program, while we directly find a closed-form expression for the edge probability p_{uv}, thus rendering fast implementation.

Second, the assumptions required are different. The assumptions $p_{uv} > \beta$ for all $u, v \in V$ and $\gamma \leq q_u \leq 1 - \gamma$ for all $u \in V$ are also required in the previous works (though may be presented in different forms). The key difference is the condition $ap(v) \leq 1 - \alpha$ for all $v \in V$. In (Netrapalli & Sanghavi, 2012), its role is replaced by the *correlation decay* condition, which requires that $\sum_{u \in N^{in}(v)} p_{uv} \leq 1 - \alpha$ for all $v \in V$. In (Narasimhan et al., 2015), it is instead assumed that $1 - \prod_{u \in N^{in}(v)} (1 - p_{uv}) \leq 1 - \alpha$ for all $v \in V$. By observing that $ap(v) \leq 1 - \prod_{u \in N^{in}(v)} (1 - p_{uv}) \leq \sum_{u \in N^{in}(v)} p_{uv}$ (see the appendix), it is easy to see that our assumptions are the weakest compared with those in (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015). Besides, Assumption 1 enjoys the advantage that it is verifiable, since one can find suitable values for α and γ by estimating $ap(v)$ and q_u from cascade samples. However, it is impossible to verify the assumptions in (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015) based only on cascade samples. We remark that our network inference algorithm replies on the assumption that each seed node is independently sampled. This assumption is also made in (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015) for the MLE method, but conceptually it might be easier to relax this assumption with MLE. We leave the relaxation of the independence sampling assumption of our method as a future work.

Finally, our algorithm has lower sample complexity compared with those in (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015). Assume that $ap(v) \leq 1 - \prod_{u \in N^{in}(v)} (1 - p_{uv}) \leq \sum_{u \in N^{in}(v)} p_{uv} \leq 1 - \alpha$. Then, Netrapalli & Sanghavi (2012) needs $\tilde{O}(\frac{1}{\alpha^7 \beta^2 \gamma} D^2 \log \frac{n}{\delta})$ samples to recover network structure, where D is the maximum in-degree of the network, while we only need $O(\frac{1}{\alpha^2 \beta^2 \gamma^3} \ln \frac{4n}{\delta})$ samples by Theorem 2. On the other hand, assume that the network structure is known and $m = |E|$. Narasimhan et al. (2015) needs $\tilde{O}(\frac{1}{\varepsilon^2 \alpha^2 \beta^2 \gamma^2 (1-\gamma)^4} mn \ln \frac{n}{\delta})$ samples to achieve $||\hat{p} - p||_2^2 \leq \varepsilon$, while we only need $O(\frac{1}{\varepsilon \alpha^2 \gamma^2} m \ln \frac{n}{\delta})$ samples by achieving $|\hat{p}_{uv} - p_{uv}| \leq \sqrt{\frac{\varepsilon}{m}}$.

4. Influence Maximization from Samples

In this section, we present several algorithms for influence maximization from samples. In Section 4.1, we present an *approximation-preserving* algorithm under Assumption 1.

Algorithm 3 Influence Maximization from Samples under Assumption 1

```
Input: A set of cascades $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^t$ and $k \in \mathbb{N}_+$.
  $\{\hat{p}_{uv}\}_{u,v \in V}$ = Estimate-Edge-Probabilities
      $((S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^t)$. { With estimation accuracy $\varepsilon k/(2n^3)$.}
  Let $\hat{G} = (V, E, \hat{p})$.
  Let $S^A = A(\hat{G}, k)$, where $A$ is a $\kappa$-approximation
      algorithm for influence maximization.
  return $S^A$.
```

In Section 4.2, we show that under an alternative assumption (Assumption 2), there is a *constant* approximation algorithm for the problem. An attractive feature of Assumption 2 (compared to Assumption 1) is that it has no requirement on the network. We also show that by slightly strengthening Assumption 2, we again obtain an *approximation-preserving* algorithm.

4.1. Influence Maximization under Assumption 1

Our first influence maximization algorithm is presented as Algorithm 3. It follows the canonical learning-and-then-optimization approach by first learning a surrogate graph $\hat{G} = (V, E, \hat{p})$ from the cascades and then executing any κ-approximation algorithm A for standard influence maximization on $\hat{G}$ to obtain a solution as output. The construction of $\hat{G}$ builds on Algorithm 1 and is obtained by estimating all the edge probabilities to a sufficiently small additive error. Algorithm 3 works under Assumption 1, since Algorithm 1 does.

The correctness of Algorithm 3 relies on Lemma 5, which translates the estimation error in edge probabilities into the learning error in the influence spread function. We use it in Theorem 3 to prove that with high probability, Algorithm 3 almost preserves the approximation ratio of any standard influence maximization algorithm A. In the following, we use $\sigma^p(\cdot)$ to specify the graph parameters explicitly.

Lemma 5 ((Narasimhan et al., 2015))**.** *Fix $S \subseteq V$. For any two edge probability vectors $\mathrm{p}, \hat{\mathrm{p}}$ with $\|\mathrm{p} - \hat{\mathrm{p}}\|_1 \leq \varepsilon/n$, we have $|\sigma^{\mathrm{p}}(S) - \sigma^{\hat{\mathrm{p}}}(S)| \leq \varepsilon$.*

Theorem 3. *Under Assumption 1, for any $\epsilon \in (0,1)$ and $k \in \mathbb{N}_+$, suppose that the number of cascades $t \geq \frac{1024}{\varepsilon^2 \alpha^2 \gamma^3} \frac{n^6}{k^2} \ln \frac{12n}{\delta}$. Let A be a κ-approximation algorithm for influence maximization. Let S^A be the set returned by Algorithm 3 and S^* be the optimal solution on the original graph. We have*

$$\Pr[\sigma(S^A) \geq (\kappa - \varepsilon)\sigma(S^*)] \geq 1 - \delta.$$

Proof. By Theorem 1, with probability at least $1 - \delta$, for any $u, v \in V$, $|\hat{p}_{uv} - p_{uv}| \leq \varepsilon k/(2n^3)$. Hence, $\|p - \hat{p}\|_1 =$

$\sum_{u,v\in V} |p_{uv} - \hat{p}_{uv}| \le \varepsilon k/(2n)$. Applying this condition to Lemma 5, we have that $|\sigma^p(S) - \sigma^{\hat{p}}(S)| \le \varepsilon k/2$ for every seed set S. We thus have

$$\begin{aligned}\sigma(S^A) &\ge \sigma^{\hat{p}}(S^A) - \varepsilon k/2 \ge \kappa \cdot \sigma^{\hat{p}}(S^*) - \varepsilon k/2\\ &\ge \kappa \cdot (\sigma(S^*) - \varepsilon k/2) - \varepsilon k/2\\ &= \kappa \cdot \sigma(S^*) - (1+\kappa)\varepsilon k/2 \ge (\kappa - \varepsilon)\sigma(S^*).\end{aligned}$$

The second inequality holds since S^A is a κ-approximation on $\hat{G}$. The last inequality holds since $\sigma(S^*) \ge k \ge (1+\kappa)k/2$. □

Compared with our learning algorithms for network inference, Algorithm 3 has an additional overhead of n^6/k^2 in the number of cascades. This is because it needs to estimate edge probabilities within an additive error of at most $\varepsilon k/(2n^3)$. One can also invoke known network inference algorithms other than ours in Algorithm 3 to obtain a similar approximate solution, but as discussed above, this only incurs higher sample complexity. We are not aware of any approach to reduce the sample complexity and leave it as an interesting open problem.

4.2. Influence Maximization under Assumptions Independent of the Network

Condition 1 of Assumption 1 depends on diffusion network, and thus our Algorithm 3 may not be applicable to all networks. In this section, we show that under an alternative assumption (Assumption 2), which is entirely independent of the diffusion network, there still exists a *constant* approximation algorithm (Algorithm 4) for influence maximization from samples.

Assumption 2 (Assumptions for Influence Maximization from Samples, Independent of the Network)**.** *For some constant $c > 0$ and parameter $\gamma \in (0, 1/2]$,*

1. $\sum_{u\in V} q_u \le ck$.
2. $\gamma \le q_u \le 1 - \gamma$ *for all* $u \in V$.

Assumption 2 consists of two conditions. The condition $\sum_{u\in V} q_u \le ck$ replaces the condition $ap(v) \le 1 - \alpha$ in Assumption 1. It means that a random seed set drawn from the product distribution $\mathcal{D}$ has an expected size at most linear in k (but not necessarily bounded above by k). Assumption 2 puts forward two plausible requirements for the seed distribution $\mathcal{D}$ and has no requirement for the underlying network. Thus, in principle, one can handle *any* social networks, as long as the seed set sampling is reasonable according to Assumption 2.

We now describe the high-level idea of Algorithm 4. It might be surprising at first glance that one can remove the condition $ap(v) \le 1 - \alpha$ for all $v \in V$. After all, it is very hard to learn information about incoming edges of v if $ap(v)$ is very close to 1. To handle this difficulty, recall that $ap(v)$ is defined as the activation probability of v in one time step. Hence, if $ap(v)$ is close to 1, v will be active with high probability starting from a random seed set. The observation suggests that one can divide nodes into two parts according to their $ap(\cdot)$. For the nodes with small $ap(\cdot)$, Assumption 1 is satisfied and one can find a good approximation for them via a similar approach as Algorithm 3. For the nodes with large $ap(\cdot)$, a random seed set is already a good approximation for them. So there is no need to learn their incoming edges. A technical issue here is that a random seed set may not be a feasible solution for the maximization task. This is the reason why we introduce Condition 1 of Assumption 2, by which the expected size of the seed set is at most linear in k. So we can replace it by its random subset of size k while keeping a constant approximation. To summarize, we find two candidate solutions whose union must be a good approximation over the whole network. If we choose one of them randomly, we will finally obtain a feasible solution with constant approximation.

Following the guidance of the above idea, Algorithm 4 first computes an estimate $\widehat{ap}(v)$ of $ap(v)$ for all $v \in V$ and partitions V into two disjoint subsets $V_1 = \{v \in V \mid \widehat{ap}(v) < 1 - \delta/(4n)\}$ and $V_2 = V \setminus V_1$. It then estimates the probabilities of incoming edges of V_1 using Algorithm 1 and sets the probabilities of incoming edges of V_2 to 1 directly for technical reasons. The constructed graph is denoted by $\hat{G}$. Let T_1 be a κ-approximation on $\hat{G}$ and $T_2 = S_{1,0}$ be the first random seed set. Finally, Algorithm 4 selects T_1 or T_2 with equal probability, and if it selects T_2 while $|T_2| > k$, it further selects a random subset of T_2 with size k, as the final output seed set S^A.

We now give an analysis of Algorithm 4. Our analysis requires a technical lemma (Lemma 6) which bounds the influence of the seed set when setting the probabilities of incoming edges of R to 1 by the influence of the seed set augmented with R. Its proof is presented in the appendix.

Lemma 6. *Let $G = (V, E, p)$ be a directed graph and $R \subseteq V$. Let $G' = (V, E, p')$ be a directed graph obtained from G as follows: $p'_{uv} = 1$ if $v \in R$ and $p'_{uv} = p_{uv}$ otherwise. Then, for any $S \subseteq V$, we have $\sigma^p(S \cup R) \ge \sigma^{p'}(S)$.*

Theorem 4. *Under Assumption 2, suppose that the number of cascades $t \ge \frac{36864}{\varepsilon^2\delta^2\gamma^3}\frac{n^8}{k^2}\ln\frac{36n}{\delta} + \frac{72n^2}{\delta^2}\ln\frac{12n}{\delta}$, and the number of samples used to estimate $ap(v)$'s is $t' = \frac{72n^2}{\delta^2}\ln\frac{12n}{\delta}$. Let A be a κ-approximation algorithm for influence maximization. Assume that $k \ge \frac{3}{c}\ln\frac{3}{\delta}$. Let S^A be the set returned by Algorithm 4 and S^* be the optimal solution on the original graph. We have that S^A is a feasible solution, and*

$$\Pr[\mathbb{E}[\sigma(S^A)] \ge \min\left\{\frac{1}{2c}, 1\right\}\frac{\kappa - \varepsilon}{2}\sigma(S^*)] \ge 1 - \delta,$$

Algorithm 4 Influence Maximization from Samples under Assumption 2

```
Input: A set of cascades $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t}$, $k \in \mathbb{N}_+$, error probability $\delta > 0$, number of samples $t' \in [t]$ used to estimate $ap(v)$'s.
 1: Set $V_1 = V$ and $V_2 = \emptyset$.
 2: for each $v \in V$ do
 3:     Use the first $t'$ samples $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t'}$ to compute $\widehat{ap}(v) = t^v/t'$, where $t^v = |\{i \in [t'] \mid v \in S_{i,1}\}|$.
 4:     if $\widehat{ap}(v) \geq 1 - \delta/(4n)$ then
 5:         Set $\hat{p}_{uv} = 1$ for all $u \in V$.
 6:         $V_1 = V_1 \setminus \{v\}$ and $V_2 = V_2 \cup \{v\}$.
 7:     end if
 8: end for
 9: $\{\hat{p}_{uv}\}_{u \in V, v \in V_1}$ = Estimate-Edge-Probabilities $((S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=t'+1}^{t})$ on $V_1$. {With accuracy $\varepsilon k/(2n^3)$, $\alpha = \delta/(6n)$ in Assumption 1.}
10: Let $\hat{G} = (V, E, \hat{p})$.
11: $T_1 = A(\hat{G}, k)$, where $A$ is a $\kappa$-approximation algorithm for influence maximization.
12: $T_2 = S_{1,0}$.
13: Let $T$ be a random set by picking $T_1$ and $T_2$ with equal probability. If $T \leq k$, let $S^A = T$; otherwise, let $S^A$ be a uniformly random subset of $T$ with $|S^A| = k$.
14: return $S^A$.
```

where the probability is taken over the randomness of $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t}$ and the expectation is taken over the randomness from line 13 of Algorithm 4.

Proof. Let $G = (V, E, p)$ be the original graph. Let $V_1 = \{v \in V \mid \widehat{ap}(v) \leq 1 - \frac{\delta}{4n}\}$ and $V_2 = V \setminus V_1$, defined as in Algorithm 4. Let $B = \{(u, v) \mid u \in V, v \in V_2\}$ be the set of all edges pointing to some node in V_2. Let $G' = (V, E, p')$ be a directed graph obtained from G as follows: $p'_{uv} = 1$ if $(u, v) \in B$ and $p'_{uv} = p_{uv}$ otherwise. Let $\hat{G} = (V, E, \hat{p})$ be a directed graph obtained from G' by replacing p_{uv} with $\hat{p}_{uv}$ for any $(u, v) \notin B$. Clearly, $\hat{G}$ is exactly the same graph we constructed in Algorithm 4.

For any node $v \in V$, by Lemma 2, when $t' = \frac{72n^2}{\delta^2} \ln \frac{12n}{\delta}$,

$$
\begin{aligned}
&\Pr[|\widehat{ap}(v) - ap(v)| \geq \delta/(12n)] \\
&\leq \exp(-t'(\delta/(12n)) \min(1/5, (\delta/(12n)) \cdot (1/ap(v))) \\
&+ \exp(-t'(\delta/(12n))^2/2ap(v)) \\
&\leq \delta/(6n) + \delta/(6n) = \delta/(3n).
\end{aligned}
$$

By union bound, with probability $1 - \delta/3$, for all nodes $v \in V$, $|\widehat{ap}(v) - ap(v)| \leq \delta/(12n)$. Specifically, for a node $v \in V_1$ with $\widehat{ap}(v) \leq 1 - \delta/(4n)$, we have $ap(v) \leq \widehat{ap}(v) + \delta/(12n) \leq 1 - \delta/(6n)$. For a node $v \in V_2$ with $\widehat{ap}(v) > 1 - \delta/(4n)$, we have $ap(v) \geq \widehat{ap}(v) - \delta/(12n) \geq 1 - \delta/(3n)$.

Since for any $v \in V_2$, $ap(v) \geq 1 - \delta/(3n)$, by union bound, it means that with probability at least $1 - \delta/3$ in one time step all nodes in V_2 are activated. Assume that indeed all nodes in V_2 are activated in one time step. Then, we have $\sigma^p(T_1 \cup T_2) = \sigma^p(T_1 \cup S_{1,0}) \geq \sigma^p(T_1 \cup V_2)$. By plugging $S = T_1, R = V_2$ into Lemma 6, we obtain $\sigma^p(T_1 \cup V_2) \geq \sigma^{p'}(T_1)$. Therefore, by submodularity of σ, $\sigma^p(T_1) + \sigma^p(T_2) \geq \sigma^p(T_1 \cup T_2) \geq \sigma^{p'}(T_1)$.

Since $\{\hat{p}\}_{u,v}$ is obtained by running Estimate-Edge-Probability with parameters $\varepsilon k/(2n^3), \alpha = \delta/(6n), \gamma$, when the number of cascades $t - t' \geq \frac{36864}{\varepsilon^2 \delta^2 \gamma^3} \frac{n^8}{k^2} \ln \frac{36n}{\delta}$, with probability $1 - \delta/3$, we have $|\hat{p}_{uv} - p_{uv}| \leq \varepsilon k/(2n^3)$ for any $(u, v) \notin B$. Since $\hat{p}_{uv} = p'_{uv} = 1$ for $(u, v) \in B$, we have $\|\hat{p} - p'\|_1 \leq \varepsilon k/(2n)$. Therefore, by Lemma 5, for any $S \subseteq V$, $|\sigma^{\hat{p}}(S) - \sigma^{p'}(S)| \leq \varepsilon k/2$. We have

$$
\begin{aligned}
\sigma^{p'}(T_1) &\geq \sigma^{\hat{p}}(T_1) - \varepsilon k/2 \geq \kappa \cdot \sigma^{\hat{p}}(S^*) - \varepsilon k/2 \\
&\geq \kappa \cdot (\sigma^{p'}(S^*) - \varepsilon k/2) - \varepsilon k/2 \\
&\geq \kappa \cdot (\sigma^{p}(S^*) - \varepsilon k/2) - \varepsilon k/2 \geq (\kappa - \varepsilon)\sigma^p(S^*).
\end{aligned}
$$

The second inequality holds since T_1 is a κ approximation of S^* on $\hat{G}$. The forth inequality holds since $\sigma^{p'}(S) \geq \sigma^p(S)$ for any $S \subseteq V$, due to $p'_{uv} \geq p_{uv}$ for any $(u, v) \in E$. The last inequality holds as long as $(1 + \kappa)k/2 \leq k \leq \sigma^p(S^*)$, which holds trivially since $\kappa \leq 1$ and $\sigma^p(S^*) \geq k$.

Combining the previous inequalities, we have

$$
\sigma^p(T_1) + \sigma^p(T_2) \geq \sigma^{p'}(T_1) \geq (\kappa - \varepsilon)\sigma^p(S^*),
$$

which implies that $\mathbb{E}[\sigma^p(T)] = \frac{1}{2}(\sigma^p(T_1) + \sigma^p(T_2)) \geq \frac{1}{2}(\kappa - \varepsilon)\sigma^p(S^*)$.

Finally, since $\sum_{u \in V} q_u \leq ck$, $\Pr[|S_{1,0}| \geq 2ck] \leq e^{-ck/3} \leq \delta/3$ when $k \geq \frac{3}{c} \ln \frac{3}{\delta}$. Assume that $|T_2| = |S_{1,0}| \leq 2ck$. If $T = T_1$ or $T = T_2$ but $|T_2| \leq k$, then $S^A = T$. If $T = T_2$ and $|T_2| > k$, then S^A is a uniform subset of T with size k. Since $\sigma(\cdot)$ is submodular, we have $\mathbb{E}[\sigma^p(S^A)] \geq \min\left\{\frac{1}{2c}, 1\right\} \mathbb{E}[\sigma^p(T)] \geq \min\left\{\frac{1}{2c}, 1\right\} \frac{\kappa - \varepsilon}{2} \sigma^p(S^*)$.

To conclude, by union bound, with probability at least $1 - \delta$, S^A is a feasible solution and $\mathbb{E}[\sigma^p(S^A)] \geq \min\left\{\frac{1}{2c}, 1\right\} \frac{\kappa - \varepsilon}{2} \sigma^p(S^*)$. □

Improving the approximation ratio. Compared with Algorithm 3, Algorithm 4 has a worse (though still constant) approximation ratio. We show that if the constant c in Assumption 2 equals to some prescribed small $\varepsilon \in (0, 1/3)$, we can modify Algorithm 4 to be an approximation-preserving algorithm as follows: let $T_1 = A(\hat{G}, (1 - 2\varepsilon)k)$ and returns $T_1 \cup T_2$ directly. It is easy to see that the modified algorithm works since T_1 loses little in the approximation

Algorithm 5 Influence Maximization from Samples under Assumption 2 with $c = \varepsilon$

```
Input: A set of cascades $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t}$ and $k \in \mathbb{N}_+$, parameter $\varepsilon \in (0, 1/3)$, error probability $\delta > 0$, number of samples $t' \in [t]$ used to estimate $ap(v)$'s.
Set $V_1 = V$ and $V_2 = \emptyset$.
for each $v \in V$ do
    Use the first $t'$ samples $(S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=1}^{t'}$ to compute $\widehat{ap}(v) = t^v/t'$, where $t^v = |\{i \in [t'] \mid v \in S_{i,1}\}|$.
    if $\widehat{ap}(v) \geq 1 - \delta/(4n)$ then
        Set $\hat{p}_{uv} = 1$ for all $u \in V$.
        $V_1 = V_1 \setminus \{v\}$ and $V_2 = V_2 \cup \{v\}$.
    end if
end for
$\{\hat{p}_{uv}\}_{u \in V, v \in V_1}$ = Estimate-Edge-Probabilities
    $((S_{i,0}, S_{i,1}, \cdots, S_{i,n-1})_{i=t'+1}^{t})$ on $V_1$. {With accuracy $\varepsilon k/(2n^3)$, $\alpha = \delta/(6n)$ in Assumption 1.}
Let $\hat{G} = (V, E, \hat{p})$.
$T_1 = A(\hat{G}, (1-2\varepsilon)k)$, where $A$ is a $\kappa$-approximation algorithm for influence maximization.
$T_2 = S_{1,0}$.
return $S^A = T_1 \cup T_2$.
```

ratio and $T_1 \cup T_2$ is feasible with high probability. The formal procedure is presented in Algorithm 5 and its guarantee is presented below.

Theorem 5. *Under Assumption 2 with $c = \varepsilon \in (0, 1/3)$, suppose that the number of cascades $t \geq \frac{36864}{\varepsilon^2\delta^2\gamma^3}\frac{n^8}{k^2}\ln\frac{36n}{\delta} + \frac{72n^2}{\delta^2}\ln\frac{12n}{\delta}$ and the number of samples used to estimate $ap(v)$'s is $t' = \frac{72n^2}{\delta^2}\ln\frac{12n}{\delta}$. Let A be an κ-approximation algorithm for influence maximization. Assume that $k \geq \frac{3}{\varepsilon}\ln\frac{3}{\delta}$. Let S^A be the set returned by Algorithm 5 and S^* be the optimal solution on the original graph. We have*

$$\Pr[|S^A| \leq k \wedge \sigma(S^A) \geq (\kappa - 3\varepsilon)\sigma(S^*)] \geq 1 - \delta.$$

Proof. By a similar analysis for Algorithm 4, $\sigma^p(T_1 \cup T_2) \geq \sigma^{p'}(T_1)$ with probability at least $1-\delta/3$, and for any $S \subseteq V$, $|\sigma^{\hat{p}}(S) - \sigma^{p'}(S)| \leq \varepsilon k/2$ with probability at least $1 - \delta/3$. We thus have

$$\begin{aligned}
\sigma^{p'}(T_1) &\geq \sigma^{\hat{p}}(T_1) - \varepsilon k/2 \\
&\geq \kappa(1-2\varepsilon) \cdot \sigma^{\hat{p}}(S^*) - \varepsilon k/2 \\
&\geq \kappa(1-2\varepsilon) \cdot (\sigma^{p'}(S^*) - \varepsilon k/2) - \varepsilon k/2 \\
&\geq \kappa(1-2\varepsilon) \cdot (\sigma^{p}(S^*) - \varepsilon k/2) - \varepsilon k/2 \\
&\geq (\kappa-2\varepsilon) \cdot \sigma^{p}(S^*) - (1+\kappa)\varepsilon k/2 \\
&\geq (\kappa-3\varepsilon) \cdot \sigma^{p}(S^*).
\end{aligned}$$

The second inequality holds since T_1 is a $\kappa(1-2\varepsilon)$ approximation of S^* on $\hat{G}$. The forth inequality holds since $\sigma^{p'}(S) \geq \sigma^p(S)$ for any $S \subseteq V$, due to $p'_{uv} \geq p_{uv}$ for any $(u, v) \in E$. The last inequality holds as long as $(1+\kappa)k/2 \leq k \leq \sigma^p(S^*)$, which holds trivially since $\kappa \leq 1$ and $\sigma^p(S^*) \geq k$. Therefore, we have $\sigma^p(S^A) = \sigma^p(T_1 \cup T_2) \geq (\kappa - 3\varepsilon)\sigma^p(S^*)$.

Finally, since $\sum_{u \in V} q_u \leq \varepsilon k$, by Lemma 1, $\Pr[|S_{1,0}| \geq 2\varepsilon k] \leq e^{-\varepsilon k/3} \leq \delta/3$ when $k \geq \frac{3}{\varepsilon}\ln\frac{3}{\delta}$. If $|S_{1,0}| \leq 2\varepsilon k$, $|S^A| = |T_1 \cup T_2| \leq (1-2\varepsilon)k + 2\varepsilon k = k$.

To conclude, by union bound, with probability at least $1-\delta$, $|S^A| \leq k$ and $\sigma(S^A) \geq (\kappa - 3\varepsilon)\sigma(S^*)$. □

5. Conclusion and Future Work

In this paper, we conduct a rigorous theoretical treatment to the influence maximization from samples (IMS) problem, and provide several end-to-end IMS algorithms with constant approximation guarantee. We also provide a novel and efficient algorithm for network inference with weaker assumptions.

There are many future directions to extend and improve this work. First, our IMS algorithms require a large number of samples (though polynomial) since we have to estimate edge probabilities to a very high accuracy. It is very interesting to investigate how to improve the sample complexity by leveraging sparsity and different importance of edges in the networks. Second, our samples contain activation sets at every step. One can further study how to do IMS when we only observe the final activation set. Other directions include studying IMS for other stochastic diffusion models (e.g. LT model or the cumulative activation model in (Shan et al., 2019)), relaxing the independent seed node sampling assumption, and going beyond influence maximization to study other optimization tasks directly from data samples.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China Grants No. 61832003, 61872334, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000, the 973 Program of China Grant No. 2016YFB1000201, K.C. Wong Education Foundation.

References

Abrahao, B. D., Chierichetti, F., Kleinberg, R., and Panconesi, A. Trace complexity of network inference. In *KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pp. 491–499. ACM, 2013.

Alon, N. and Spencer, J. H. *The Probabilistic Method, Third Edition*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 2008.

Badanidiyuru, A., Dobzinski, S., Fu, H., Kleinberg, R., Nisan, N., and Roughgarden, T. Sketching valuation functions. In *SODA 2012, Kyoto, Japan, January 17-19, 2012*, pp. 1025–1035, 2012.

Balkanski, E., Rubinstein, A., and Singer, Y. The power of optimization from samples. In *NIPS 2016, December 5-10, 2016, Barcelona, Spain*, pp. 4017–4025, 2016.

Balkanski, E., Immorlica, N., and Singer, Y. The importance of communities for learning to influence. In *NIPS 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5862–5871, 2017a.

Balkanski, E., Rubinstein, A., and Singer, Y. The limitations of optimization from samples. In *STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pp. 1016–1027. ACM, 2017b.

Bansal, N. and Sviridenko, M. The santa claus problem. In Kleinberg, J. M. (ed.), *STOC 2006, Seattle, WA, USA, May 21-23, 2006*, pp. 31–40. ACM, 2006.

Chen, W., Sun, X., Zhang, J., and Zhang, Z. Optimization from structured samples for coverage functions. In *ICML 2020*.

Chen, W., Wang, Y., Yuan, Y., and Wang, Q. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016. A preliminary version appeared as Chen, Wang, and Yuan, “combinatorial multi-armed bandit: General framework, results and applications”, ICML 2013.

Daneshmand, H., Gomez-Rodriguez, M., Song, L., and Schölkopf, B. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML 2014*.

Du, N., Song, L., Smola, A. J., and Yuan, M. Learning networks of heterogeneous influence. In *NIPS 2012, December 3-6, 2012, Lake Tahoe, Nevada, United States*, pp. 2789–2797, 2012.

Du, N., Song, L., Gomez-Rodriguez, M., and Zha, H. Scalable influence estimation in continuous-time diffusion networks. In *NIPS 2013, December 5-8, 2013, Lake Tahoe, Nevada, United States*, pp. 3147–3155, 2013.

Du, N., Liang, Y., Balcan, M., and Song, L. Influence function learning in information diffusion networks. In *ICML 2014, Beijing, China, 21-26 June 2014*, 2014.

Feige, U. A threshold of ln *n* for approximating set cover. *J. ACM*, 45(4):634–652, 1998.

Gomez-Rodriguez, M., Leskovec, J., and Krause, A. Inferring networks of diffusion and influence. In *KDD 2010, Washington, DC, USA, July 25-28, 2010*, pp. 1019–1028. ACM, 2010.

Gomez-Rodriguez, M., Balduzzi, D., and Schölkopf, B. Uncovering the temporal dynamics of diffusion networks. In *ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*, pp. 561–568. Omnipress, 2011.

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

He, X., Xu, K., Kempe, D., and Liu, Y. Learning influence functions from incomplete observations. In *NIPS 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2065–2073, 2016.

Kempe, D., Kleinberg, J. M., and Tardos, É. Maximizing the spread of influence through a social network. In *KDD 2003, Washington, DC, USA, August 24 - 27, 2003*, pp. 137–146. ACM, 2003.

Mitzenmacher, M. and Upfal, E. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

Myers, S. A. and Leskovec, J. On the convexity of latent social network inference. In *NIPS 2010, December 6-9, 2010, Vancouver, British Columbia, Canada*, pp. 1741–1749. Curran Associates, Inc., 2010.

Narasimhan, H., Parkes, D. C., and Singer, Y. Learnability of influence in networks. In *NIPS 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 3186–3194, 2015.

Netrapalli, P. and Sanghavi, S. Learning the graph of epidemic cascades. In *SIGMETRICS 2012, London, United Kingdom, June 11-15, 2012*, pp. 211–222. ACM, 2012.

Pouget-Abadie, J. and Horel, T. Inferring graphs from cascades: A sparse recovery framework. In *ICML 2015, Lille, France, 6-11 July 2015*, 2015.

Rosenfeld, N., Balkanski, E., Globerson, A., and Singer, Y. Learning to optimize combinatorial functions. In *ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018.

Shan, X., Chen, W., Li, Q., Sun, X., and Zhang, J. Cumulative activation in social networks. *Sci. China Inf. Sci.*, 62 (5):52103:1–52103:21, 2019.

Appendix

A. Comparing Assumptions

We summarize the assumptions used in (Netrapalli & Sanghavi, 2012; Narasimhan et al., 2015) below and show that they are strictly stronger than our assumptions.

Assumption 3 (Assumptions in (Netrapalli & Sanghavi, 2012))**.** *For some parameters $\alpha, \beta \in (0,1)$,*

1. *$p_{uv} \geq \beta$ for all $(u,v) \in E$.*
2. *(Correlation decay) $\sum_{u \in N^{in}(v)} p_{uv} < 1 - \alpha$ for all $v \in V$.*
3. *$q_u d_v < 1/2$ for all $u, v \in V$.*

Assumption 4 (Assumptions in (Narasimhan et al., 2015))**.** *For some parameters $\beta \geq \alpha \in (0, 1/2)$ and $\gamma \in (0,1)$,*

1. *$p_{uv} \geq \beta$ for all $(u,v) \in E$.*
2. *$1 - \prod_{u \in N^{in}(v)}(1 - p_{uv}) \leq 1 - \alpha$ for all $v \in V$.*
3. *$\gamma \leq q_u \leq 1 - \gamma$ for all $u \in V$.*

Lemma 7. *$ap(v) \leq 1 - \prod_{u \in N^{in}(v)}(1 - p_{uv}) \leq \sum_{u \in N^{in}(v)} p_{uv}$.*

Proof. The first inequality follows from $ap(v) = 1 - \prod_{u \in N^{in}(v)}(1 - q_u p_{uv}) \leq 1 - \prod_{u \in N^{in}(v)}(1 - p_{uv})$, since $q_u \leq 1$ for all $u \in V$. The second inequality follows from the claim below.

Claim 1. *For any $x_1, \cdots, x_n \in [0,1]$, $\prod_{i=1}^{n}(1 - x_i) \geq 1 - \sum_{i=1}^{n} x_i$.*

Proof of the Claim. The claim holds trivially when $n = 1$. Assume that the claim holds for any $k < n$. Then,

$$\prod_{i=1}^{n}(1 - x_i) = \prod_{i=1}^{n-1}(1 - x_i)(1 - x_n) \geq (1 - \sum_{i=1}^{n-1} x_i)(1 - x_n) \geq 1 - \sum_{i=1}^{n} x_i.$$

The two inequalities both hold by induction. □

B. Omitted Proofs

B.1. Proof of Lemma 4

Lemma 4. *Under Assumption 1, for any $\eta \in (0, 4/5), \delta \in (0,1)$, for $\hat{q}_u$, $\widehat{ap}(v)$, and $\widehat{ap}(v \,|\, \bar{u})$ defined in Algorithm 1, if the number of samples $t \geq \frac{16}{\gamma \eta^2} \ln \frac{12n}{\delta}$, with probability at least $1 - \delta$, we have*

1. *$|\hat{q}_u - q_u| \leq \eta q_u$ for all $u \in V$,*
2. *$|\widehat{ap}(v) - ap(v)| \leq \eta$ for all $v \in V$,*
3. *$|\widehat{ap}(v \,|\, \bar{u}) - ap(v \,|\, \bar{u})| \leq \eta$ for all $u, v \in V$.*

Proof. For a node $u \in V$, for $i \in [t]$, let X_i be a 0-1 random variable such that $X_i = 1$ iff $u \in S_{i,0}$. Thus $\hat{q}_u = \sum_{i=1}^{t} X_i / t$. By Lemma 1,

$$\Pr[|\hat{q}_u - q_u| \geq \eta q_u] = \Pr[|\sum_{i=1}^{t} X_i - t q_u| \geq \eta t q_u] \leq 2\exp(-t q_u \eta^2/3) \leq 2\exp(-t\gamma\eta^2/3) \leq \delta/(3n).$$

The last inequality requires that $t \geq \frac{3}{\gamma\eta^2} \ln \frac{6n}{\delta}$.

For a node $v \in V$, for $i \in [t]$, let Y_i be a 0-1 random variable such that $Y_i = 1$ iff $v \in S_{i,1}$. Thus $\widehat{ap}(v) = \sum_{i=1}^{t} Y_i / t$. By Lemma 2 ($a = \eta t, \mu = t \cdot ap(v)$),

$$\begin{aligned}\Pr[|\widehat{ap}(v) - ap(v)| \geq \eta] &\leq \exp(-\eta t \min(1/5, \eta/(4 \cdot ap(v)))) + \exp(-\eta^2 t/(2 \cdot ap(v))) \\ &\leq \exp(-\eta t \min(1/5, \eta/4)) + \exp(-\eta^2 t/2) \leq \delta/(6n) + \delta/(6n) = \delta/(3n).\end{aligned}$$

The second inequality holds since $ap(v) \leq 1$ by definition. The last inequality requires that $t \geq \frac{4}{\eta^2} \ln \frac{6n}{\delta}$.

For $u \in V$, let $t_{\bar{u}} = |\{i \in [t] \mid u \notin S_{i,0}\}|$ be the number of cascades where u is not a seed. Since $q_u \leq 1 - \gamma$, $8 \ln(12n/\delta)/\eta^2 \leq t(1 - q_u)/2$. By Lemma 1 ($a = 1/2, \mu = t(1 - q_u)$),

$$\Pr[t_{\bar{u}} \leq 8 \ln(12n/\delta)/\eta^2] \leq \Pr[t_{\bar{u}} \leq t(1 - q_u)/2] \leq \exp(-t(1 - q_u)/8) \leq \exp(-t\gamma/8) \leq \delta/(6n^2).$$

The last inequality holds as long as $t \geq 16 \ln(6n/\delta)/\gamma$.

Given a fixed ℓ, assume that there are $t_{\bar{u}} = \ell$ cascades where u is not a seed. For $i \in [\ell]$, let Z_i be a 0-1 random variable such that $Z_i = 1$ iff $v \in S_{i,1}$ in the i-th cascade, among the ℓ cascades where u is not a seed. Then, $\widehat{ap}(v \,|\, \bar{u}) = \sum_{i=1}^{\ell} Z_i/\ell$. By Lemma 2 ($a = \eta\ell, \mu = \ell \cdot ap(v \,|\, \bar{u})$),

$$\begin{aligned}\Pr[|\widehat{ap}(v \,|\, \bar{u}) - ap(v \,|\, \bar{u})| \geq \eta \,|\, t_u = \ell] &\leq \exp(-\eta\ell \min(1/5, \eta/(4 \cdot ap(v \,|\, \bar{u})))) + \exp(-\eta^2 \ell/(2 \cdot ap(v \,|\, \bar{u}))) \\ &\leq \exp(-\eta\ell \min(1/5, \eta/4)) + \exp(-\eta^2 \ell/2).\end{aligned}$$

The last inequality holds since $ap(v \,|\, u) \leq 1$ by definition. If $\ell \geq 8 \ln(12n/\delta)/\eta^2$, then

$$\Pr[|\widehat{ap}(v \,|\, \bar{u}) - ap(v \,|\, \bar{u})| \geq \eta \,|\, t_u = \ell] \leq \delta/(6n^2).$$

By law of total probability,

$$\begin{aligned}&\Pr[|\widehat{ap}(v \,|\, \bar{u}) - ap(v \,|\, \bar{u})| \geq \eta] \\ &\leq \left(\sum_{\ell < 8 \ln(12n/\delta)/\eta^2} + \sum_{\ell \geq 8 \ln(12n/\delta)/\eta^2} \right) \Pr[t_u = \ell] \Pr[|\widehat{ap}(v \,|\, \bar{u}) - ap(v \,|\, \bar{u})| \geq \eta \mid t_u = \ell] \\ &\leq \Pr[t_{\bar{u}} < 8 \ln(12n/\delta)/\eta^2] + \Pr[t_{\bar{u}} \geq 8 \ln(12n/\delta)/\eta^2] \cdot \delta/(6n^2) \\ &\leq \delta/(6n^2) + \delta/(6n^2) = \delta/(3n^2).\end{aligned}$$

The lemma follows immediately by union bound. □

B.2. Proof of Lemma 6

To prove Lemma 6, we will use *live-edge graphs* to interpret the IC model and help understand the influence spread. Formally, a live-edge graph corresponding the IC model is a random subgraph of G such that each edge (u, v) is picked independently with probability p_{uv}. Let $\sigma_u(S)$ be the probability that u is reachable from S in the live-edge graph. Then, $\sigma_u(S)$ is also the probability that u is activated by S and hence the influence spread function $\sigma(S) = \sum_{u \in V} \sigma_u(S)$. For a node $u \in V$, a fixed edge set A and a seed set S, let $\mathbf{1}_u(A, S)$ be the indicator variable that u is reachable from S through edges in A. Then, $\sigma_u(S)$ can be written as

$$\sigma_u(S) = \sum_{A \subseteq E} \prod_{(a,b) \in A} p_{ab} \prod_{(a,b) \in E \setminus A} (1 - p_{ab}) \mathbf{1}_u(A, S).$$

Finally, when needed, we will use notations σ^p and σ_u^p to specify the edge probabilities explicitly.

Lemma 6. *Let $G = (V, E, p)$ be a directed graph and $R \subseteq V$. Let $G' = (V, E, p')$ be a directed graph obtained from G as follows: $p'_{uv} = 1$ if $v \in R$ and $p'_{uv} = p_{uv}$ otherwise. Then, for any $S \subseteq V$, we have $\sigma^p(S \cup R) \geq \sigma^{p'}(S)$.*

Proof. Let $B = \{(u, v) \mid u \in V, v \in R\}$ be the set of in-edges of R. By definition, $p'_{ab} = 1$ for $(a, b) \in B$ and $p'_{ab} = p_{ab}$ otherwise. Let $\mathcal{B} \subseteq B$ be a random subset of B such that for each edge $(a, b) \in B$, $(a, b) \in \mathcal{B}$ independently with probability

p_{ab}. For a fixed $u \in V$, we consider its activation probability. First, we have

$$\begin{aligned}\sigma_u^p(S \cup R) &= \sum_{A \subseteq E \setminus B} \prod_{(a,b) \in A} p_{ab} \prod_{(a,b) \in E \setminus (A \cup B)} (1 - p_{ab}) \mathbb{E}[\mathbf{1}_u(A \cup \mathcal{B}, S \cup R)] \\ &= \sum_{A \subseteq E \setminus B} \prod_{(a,b) \in A} p_{ab} \prod_{(a,b) \in E \setminus (A \cup B)} (1 - p_{ab}) \mathbf{1}_u(A \cup B, S \cup R).\end{aligned}$$

The expectation in the first equality is taken over the randomness of $\mathcal{B}$. The second equality holds since R itself is part of the seed set, and it makes no difference whether its in-edges B are picked into the live-edge graph. Next, by the definition of p',

$$\sigma_u^{p'}(S) = \sum_{A \subseteq E \setminus B} \prod_{(a,b) \in A} p_{ab} \prod_{(a,b) \in E \setminus (A \cup B)} (1 - p_{ab}) \mathbf{1}_u(A \cup B, S).$$

It follows that $\sigma_u^p(S \cup R) \geq \sigma_u^{p'}(S)$ and therefore $\sigma^p(S \cup R) \geq \sigma^{p'}(S)$. □