Influence Maximization over Markovian Graphs: A Stochastic Optimization Approach

Buddhika Nettasinghe, *Student Member, IEEE* and Vikram Krishnamurthy, *Fellow, IEEE*

***Abstract*—This paper considers the problem of randomized influence maximization over a Markovian graph process: given a fixed set of nodes whose connectivity graph is evolving as a Markov chain, estimate the probability distribution (over this fixed set of nodes) that samples a node which will initiate the largest information cascade (in expectation). Further, it is assumed that the sampling process affects the evolution of the graph i.e. the sampling distribution and the transition probability matrix are functionally dependent. In this setup, recursive stochastic optimization algorithms are presented to estimate the optimal sampling distribution for two cases: 1) transition probabilities of the graph are unknown but, the graph can be observed perfectly 2) transition probabilities of the graph are known but, the graph is observed in noise. These algorithms consist of a neighborhood size estimation algorithm combined with a variance reduction method, a Bayesian filter and a stochastic gradient algorithm. Convergence of the algorithms are established theoretically and, numerical results are provided to illustrate how the algorithms work.**

***Index Terms*—Influence maximization, stochastic optimization, Markovian graphs, independent cascade model, variance reduction, Bayesian filter.**

I. Introduction

Influence maximization refers to the problem of identifying the most influential node (or the set of nodes) in a network, which was first studied in the seminal paper [1]. However, most work related to influence maximization so far has been limited by one or more of the following assumptions:

1) Deterministic network (with no random evolution)
2) Fully observed graph (instead of noisy observations of the graph)
3) Passive nodes (as opposed to active nodes that are responsive to the influence maximization process).

This paper attempts to relax the above three assumptions. We develop stochastic optimization algorithms for influence maximization over a randomly evolving, partially observed network of active nodes (see Fig. 1 for a schematic overview of our approach).

To understand the motivation behind this problem, consider a social network graph where nodes represent individuals and the directed edges represent connectivity between them. Assume that this graph evolves in a Markovian manner[1] with time. Further, each individual can pass/receive messages (also called infections depending on the context) from their neighbors by communicating over the directed edges of the graph. Communication over these edges will incur time delays that are independently and identically distributed (across the edges) according to a known distribution. An influence maximizer wants to periodically convey messages (e.g. viral marketing) that expire in a certain time window to the nodes in this evolving social network. Conveying each message to the nodes in the social network is achieved by sampling a node (called seed node) from the graph according to a probability distribution and, giving the message to that seed node. Then, the seed node initiates an information cascade by transmitting the message to its neighbors with random delays. The nodes that receive the message from their neighbors will continue to follow the same steps, until the message expires. It is assumed that the graph remains same throughout the diffusion of one message i.e. graph evolves on a slower time scale compared to the expiration time of a message. Further, we allow the nodes of the social network to be active: nodes are aware of the sampling distribution and, respond by modifying the transition probabilities of the graph according to that distribution (for example, due to the incentive that they receive for being the most influential node[2]). This makes the transition probability matrix functionally dependent on the sampling distribution. In this setting, the goal of the influence maximizer is to compute the sampling distribution which maximizes the expected total number of nodes that are infected (size of the information cascade) before a message expires (considering the randomness of sampling the nodes, message diffusion process as well as the graph evolution). This motivates us to pursue the aim of this paper, which is to devise a method for the influence maximizer to estimate the optimal sampling distribution recursively, with each message that is distributed.

The main results of this paper are two recursive stochastic gradient algorithms, for the influence maximizer to recursively estimate (track) the optimal sampling distribution for the following cases:

1) influence maximizer does not know the transition probability matrix but, has perfect (non-noisy) observations of the sample path of the graph.
2) influence maximizer knows the transition probability matrix but, has only partial (noisy) observations of the sample path of the graph evolution.

Authors are with the Department of Electrical and Computer Engineering, Cornell University and, Cornell Tech, 2 West Loop Rd, New York, NY 10044, USA e-mail: {dwn26, vikramk}@cornell.edu.

[1] Dynamics of social networks (such as seasonal variations in friendship networks) can naturally be modeled as Markov processes. Another example would be a vehicular network where, the inter-vehicle communication/connectivity graph has a Markovian evolution due to their movements. Refer [2] for an example in the context of social networks.

[2] Another example for active nodes is a network of computers that are adaptively modifying their connectivity network, depending on how vulnerable each computer is to a virus attack.

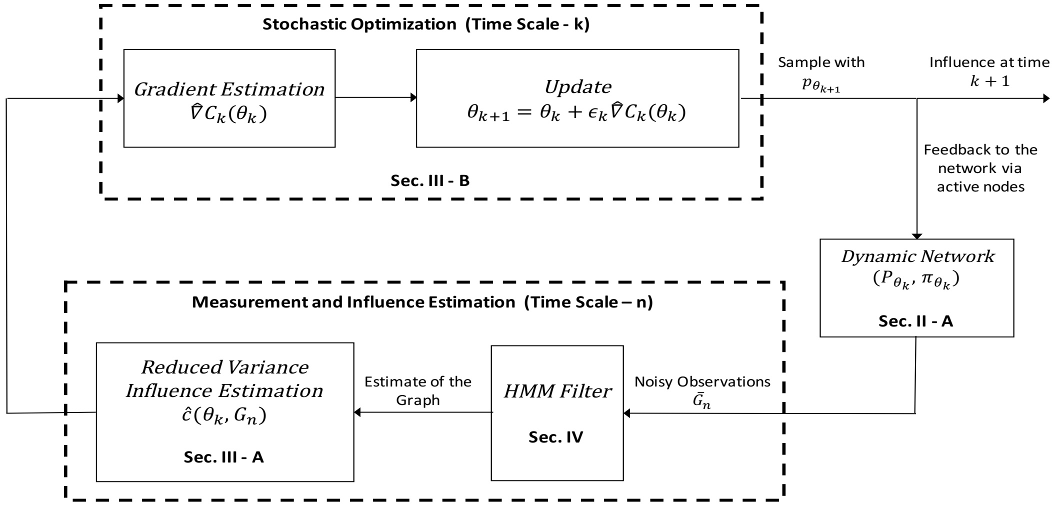


Fig. 1: Schematic diagram of the proposed stochastic optimization algorithm for influence maximization over a partially observed dynamic network of active nodes, showing how the algorithmic sub-components are integrated to form the closed loop (with feedback from the sampling process) stochastic optimization algorithm and their organization in the paper.

The key components of the above two algorithms (illustrated in Fig. 1) include the following.

- **Reduced variance neighborhood size estimation algorithm**: Influence maximization problems involve estimating the influence of nodes, which can be posed as a problem of estimating the (expected) sizes of node neighborhoods. For this, we use a stochastic simulation based neighborhood size estimation algorithm (which utilizes a modified Dijkstra's algorithm, combined with an exponential random variable assignment process), coupled with a variance reduction approach. It is shown that this reduced variance method improves the convergence of the proposed algorithms when tracking the optimal influence in a time evolving system.
- **Stochastic optimization with delayed observations of the graph process:** The observations of the graph sample path in the two algorithms (in main contributions) are not assumed to be available in real time. Instead, it is sufficient if the sample paths of finite lengths become available as batches of data with some time delay[3].These finite length graph sample paths are used in a stochastic optimization method that is based on the simultaneous perturbation stochastic approximation (SPSA) method, coupled with a finite sample path gradient estimation method for Markov processes. The proposed algorithms are applicable even for the more general case where, the system model (state space of the Markovian graph process, the functional dependency between the sampling distribution and transition probabilities, etc) is varying on a slower (compared to the stochastic optimization algorithm) time scale.
- **Bayesian filter for noisy graph sample paths:** In the algorithm for the second case (in main contributions), the sample paths are assumed to be observed in noise. In this case, a Bayesian filter is utilized to estimate the underlying state of the graph using the noisy sample path as the input. The estimates computed by the Bayesian filter are then utilized in the stochastic optimization process while preserving the (weak) convergence of the stochastic approximation algorithm.

Related Work: The influence maximization problem was first posed in [1] as a combinatorial optimization problem for two widely accepted models of information spreading in social networks: independent cascade model and the linear threshold model. [1] shows that solving this problem is NP-hard for both of these models and, utilizes a greedy (submodular function) maximization approach to devise algorithms with a $1 - \frac{1}{e}$ approximation guarantee. Since then, this problem and its variants have been widely studied using different techniques and models. [3], [4], [5], [6] studies the problem in a competitive/adversarial settings with multiple influence maximizers and provide equilibrium results and approximation algorithms. [7], [8] considers the case where all the nodes of the graph are not initially accessible to the influence maximizer and, proposes a multistage stochastic optimization method that harvests the power of a phenomenon called friendship paradox [9], [10]. Further, [11], [12] provide heuristic algorithms for the influence maximization problem on the independent cascade model, which are more efficient compared to the originally proposed algorithms in [1]. Our work is motivated by the problems studied in [13], [14]. [13] points out that estimating the expected cascade size under the independent cascade model can be posed as a neighborhood size estimation problem on a graph and, utilizes a size estimation framework proposed in [15] (and used previously in [16]) to obtain an unbiased estimate of this quantity. [14] highlights that the

[3] In most real world applications, one can only trace back the evolution of a social network over a period of time (length of the finite sample path), instead of monitoring it in real time. e.g. how the graph has evolved over the month of January becomes available to the influence maximizer only at the end of February due to delays in obtaining data.

study of the influence maximization problem has mostly been limited to the context of static graphs and, proposes a random probing method for the case where the graph may evolve randomly. Motivated by these observations, we focus on a continuous time variant [13], [17] of the independent cascade model and, allow the underlying social network graph to evolve slowly as a Markov process. The solution approaches proposed in this paper belongs to the class of recursive stochastic approximation methods. These methods have been utilized previously to solve many problems in the field of multi-agent networks [18], [19], [20].

Organization: Sec. II presents the network model, related definitions of influence on graphs, the definition of the main problem and finally, a discussion of some practical details. Sec. III presents the recursive stochastic optimization algorithm (along with convergence theorems) to solve the main problem for the case of fully observed graph with unknown transition probabilities (case 1 of the main contributions). Sec. IV extends to the case of partially observed graph with known transition probabilities (case 2 of the main contributions). Sec. V provides numerical results to illustrate the algorithms presented.

II. Diffusion Model and the Problem of Randomized Influence Maximization

This section describes Markovian graph process, how information spreads in the graph (information diffusion model) and, provides the definition of the main problem. Further, motivation for the problem, supported by work in recent literature, is provided to highlight some practical details.

A. Markovian Graph Process, Frequency of the Messages and the Independent Cascade (IC) Diffusion Model

Markovian Graph Process: The social network graph at discrete time instants $n = 0, 1, \dots$ is modeled as a directed graph $G_n = (V, E_n)$, consisting of a fixed set of individuals V, connected by the set of directed edges E_n. The graph evolves as a Markov process $\{G_n = (V, E_n)\}_{n \geq 0}$ with a finite state space $\mathcal{G}$, and a parameterized regular transition matrix P_θ with a unique stationary distribution π_θ (where, θ denotes the parameter vector which lies in a compact subset of an M-dimensional Euclidean space). Henceforth, n is used to denote the discrete time scale on which the Markovian graph process evolves.

Frequency of the Messages: An influence maximizer distributes messages to the nodes in this evolving network. We assume that the messages are distributed periodically[4] at time instants $n = kN$ where, the positive integer N denotes the period.

Observations of the Finite Sample Paths: When the influence maximizer distributes the $(k+1)^{\text{th}}$ message at time $n = (k+1)N$, only the finite sample path of the Markovian graph process $\{G_n\}_{kN \leq n < kN + \bar{N}}$ for some fixed $\bar{N} \in \mathbb{N}$ with $\bar{N} < N$, is visible to the influence maximizer.

Information Diffusion model: As explained in the example in Sec. I, nodes pass messages they receive to their neighbors with random delays. This method of information spreading in graphs is formalized by independent cascade (IC) model of information diffusion. Various versions of this IC model have been studied in literature. We use a slightly different version of the IC model utilized in [13], [17] and, it is as follows briefly. When the influence maximizer gives a message to a set of seed nodes[5] $\mathcal{I}_n \subseteq V$ at time n, a time variable $t \in \mathbb{R}_{\geq 0}$ is initialized at $t = 0$. Here, t is the continuous time scale on which the diffusion of the message takes place and, is different from the discrete time scale $n \in \mathbb{Z}_{\geq 0}$ on which the graph evolves (and the messages are distributed periodically). Further, the time scale t is nested in the time scale n: t is set to $t = 0$ with each new message distributed by the influence maximizer. The set of seed nodes $\mathcal{I}_n$ (that receives the message from influence maximizer at n and $t = 0$) transmits the message through edges attached to them in graph $G_n = (V, E_n)$. Any edge $(j, i) \in E_n$ which transmits the message induces a random delay distributed according to probability density function $\pi_{tt}(\tau_{ji})$ (called transmission time distribution). Further, any edge transmits the message only once. Only the neighbor which infects a node first will be considered as the true parent node (considering the infection propagation) of the infected node. This process continues until the message expires at $t = T$ (henceforth referred to as the message expiration time) and, the diffusion process stops at that time. It is assumed that the discrete time scale n on which the graph evolves is slower than T i.e. the graph will remain the same at least for $t \in [0, T]$. Then, the same message spreading process will take place when the next message is distributed.

For any realization of this random message spreading process (taking place in the interval $t \in [0, T]$), the subgraph induced by the subset of edges through which the infection propagated (i.e. parent-child pairs defined by the message propagation) will be a directed graph with a single parent for each node. Hence, any realization of the random message spreading process will induce a Directed Acyclic Graph (DAG) as a subgraph. Further, due to the DAG induced by the propagation of a message, the infection time t_i of each node $i \in V$ satisfies the *shortest path property*: conditional on a graph $G_n = (V, E_n)$, and a set of pairwise transmission times $\{\tau_{ji}\}_{(ji) \in E_n}$, the infection time t_i of $i \in V$ is given by,

$$t_i = g_i(\{\tau_{ji}\}_{(ji) \in E_n} | \mathcal{I}) \tag{1}$$

where, $g_i(\{\tau_{ji}\}_{(ji) \in E_n} | \mathcal{I})$ denotes the length of the shortest path from $\mathcal{I} \subseteq V$ to $i \in V$, with edge lengths $\{\tau_{ji}\}_{(ji) \in E_n}$.

B. Influence of a Set of Nodes Conditional on a Graph

We use the following definition (from [13], [21]) of influence of a set of nodes $\mathcal{I} \subseteq V$, on a graph $G = (V, E)$, for the

[4] The assumption of periodic messages is not required for the problem considered in this paper and the proposed algorithms. It suffices if the messages are distributed with some minimum time gap between them (on the time scale n).

[5] We consider the case where the influence maximizer selects a set of seed nodes instead of one seed node (as in the motivating example in Sec. I) to keep the definitions of this section more general.

diffusion model introduced in Sec. II-A. Further, $\mathbb{E}_{\pi_{tt}}\{\cdot\}$ is used to denote expectation over a set of pairwise transmission times (associated with each edge) sampled independently from π_{tt}.

Definition 1. *The Influence, $\sigma_G(\mathcal{I},T)$, of a set of nodes $\mathcal{I}$, given a graph $G=(V,E)$ and a fixed time window T is,*

$$\sigma_G(\mathcal{I},T) = \mathbb{E}_{\pi_{tt}}\Big\{\sum_{i\in V}\mathbb{1}_{\{t_i\leq T\}}\big|\mathcal{I},G\Big\} \quad (2)$$

$$= \sum_{i\in V}\mathbb{P}_{\pi_{tt}}\{t_i\leq T\big|\mathcal{I},G\}. \quad (3)$$

In (2), the influence $\sigma_G(\mathcal{I},T)$ of the set of nodes $\mathcal{I}\subset V$ is the expected number of nodes infected within time T, by the diffusion process (characterized by the distribution of transmission delays π_{tt}) on the graph G, that started with the set of seed nodes $\mathcal{I}$.

Note that the set of infection times $\{t_i\}_{i\in V}$, in (2) are dependent random variables. Therefore, obtaining closed form expressions for marginal cumulative distributions in (3) involves computing $|V|-1$ dimensional integral which is not possible (in closed form) for many general forms of the transmission delay distribution. Further, numerical evaluation of these marginal distributions is also not feasible since it will involve discretizing the domain $[0,\infty)$ (refer [13] for a more detailed description about the computational infeasibility of the calculation of the expected value in (4)).

Seed selection by sampling: We assume that the seed nodes are accessible to the influence maximizer only through a parameterized probability distribution p_θ and, the influence maximizer can only tune the parameters of this probability distribution. A detailed discussion of this assumption is provided in Sec. II-D.

C. Randomized Influence Maximization over a Markovian Graph Process: Problem Definition

We define the influence of a parameterized probability distribution p_θ, $\theta\in\mathbb{R}^M$ over the set of nodes V (i.e. $p_\theta\in\Delta(V)$ where $\Delta(V)$ is the set of all probability distributions over the set V) of a Markovian graph process as below (henceforth, we will use $\sigma_G(i,T)$ to denote $\sigma_G(\{i\},T)$ with a slight notational misuse).

Definition 2. *The Influence, $C(\theta)$, of probability distribution $p_\theta \in \Delta(V)$, on a Markovian graph process $\{G_n=(V,E_n)\}_{n\geq 0}$ with a finite state space $\mathcal{G}$, a regular transition matrix P_θ and, a unique stationary distribution $\pi_\theta\in\Delta(\mathcal{G})$ is,*

$$C(\theta) = \mathbb{E}_{G\sim\pi_\theta}\{c(\theta,G)\} \quad (4)$$

where,

$$c(\theta,G) = \mathbb{E}_{i\sim p_\theta}\{\sigma_G(i,T)\}. \quad (5)$$

Eq. (5) averages influence $\sigma_G(i,T)$ using the sampling distribution p_θ, to obtain $c(\theta,G)$, which is the influence of the sampling distribution p_θ conditional on the graph G. Then, (4) averages $c(\theta,G)$ using the unique stationary distribution π_θ of the graph process in order to obtain $C(\theta)$, which is the influence of the sampling distribution p_θ over the Markovian graph process. We will refer to $c(\theta,G_n)$ as the conditional (on graph G_n) influence function (at time n) and, $C(\theta)$ as the influence function, of the sampling distribution p_θ.

Remark 1. The sampling distribution p_θ is treated as a function of θ, which also parameterizes the transition matrix P_θ of the graph process. This functional dependency models the feedback (via the active nodes) from the sampling of nodes (by the influence maximizer) to the evolution of the graph, as indicated by the feedback loop in Fig. 1 (and also discussed in an example setting in Sec. I).

In this context, the main problem studied in this paper can be defined as follows.

Problem Definition. *Randomized influence maximization over a Markovian graph process $\{G_n=(V,E_n)\}_{n\geq 0}$ with a finite state space $\mathcal{G}$, a regular transition matrix $P_{\theta_{k'}}$ and, a unique stationary distribution $\pi_{\theta_{k'}}$ aims to recursively estimate the time evolving optima,*

$$\theta^*_{k'} \in \underset{\theta\in\mathbb{R}^M}{\arg\max}\; C_{k'}(\theta) \quad (6)$$

where, $C_{k'}(\theta)$ is the influence function (Definition 2) that is evolving on the slower time scale $k'\in\mathbb{Z}_{\geq 0}$ (compared to the message period N over the time scale n) .

Remark 2. The reason for allowing the influence function $C_{k'}(\theta)$ (and therefore, the solution $\theta^*_{k'}$) in (6) to evolve over the slow time scale k' is because the functional dependency of the sampling distribution and the transition probability matrix (which gives how the graph evolution depends on sampling process) may change over time. Further, the state space $\mathcal{G}$ of the graph process may also evolve over time. Such changes (with time) in the system model are encapsulated by modeling the influence function as a time evolving quantity. However, to keep the notation manageable, we assume that the influence function $C_{k'}(\theta)$ does not evolve over time in the subsequent sections i.e. it is assumed that

$$C_{k'}(\theta) = C(\theta), \forall\, k'\in\mathbb{Z}_{\geq 0}. \quad (7)$$

This assumption is used to keep the notation manageable and, can be removed without affecting the main algorithms presented in this paper. Further, we assume that the $C(\theta)$ has a Lipschitz continuous derivative.

D. Discussion about Key Aspects of the System Model

1) Seed Selection by Sampling: In our model, we assumed that the seeds are accessible to the influence maximizer only through a probability distribution p_θ and, the influence maximizer can only tune the parameters of this probability distribution. This assumption is largely motivated by the various (tunable) network sampling schemes that are deployed in the context of large networks. For example, consider the following example.
Example: Two widely used social network sampling methods for a graph $G=(V,E)$ are:

- Method 1: Sampling random links and then choosing a random end node of that link i.e. sampling neighbors (random ends of random links) from the network uniformly. In this method, each node $v \in V$ with degree $d(v)$ is sampled with probability $\frac{d(v)}{\sum_{v\in V} d(v)}$.
- Method 2: Sampling nodes from V uniformly.

Each of the above two sampling mechanisms may not be the best for maximizing the influence over the Markovian graph process (with respect to the stationary distribution). Hence, the influence maximizer might be interested in achieving a middle ground via a tuning parameter. One approach of doing this is to sample $v \in V$ with probability $p_\theta(v) = \frac{d(v)^\theta}{\sum_{v\in V} d(v)^\theta}$ where $\theta \in [0,1]$. Then, when the tunable parameter $\theta = 0$, this is equal to the sampling method 2 and when $\theta = 1$ this is equal to sampling method 1. Further, $\theta \in (0,1)$ achieves a trade off between the two methods. Assume that the influence maximizer has a sampling method that obtains samples from p_θ for any given $\theta \in [0,1]$. Then, the aim of the influence maximizer is to optimize θ so that the expected influence $C(\theta)$ defined in (4) is maximized. In Sec. V-B, we use this example to illustrate how the influence maximizer can optimize $C(\theta)$ recursively using the proposed algorithms in this paper.

Similarly, this parametric sampling distribution might represent any practical setting where the influence maximizer can access seeds only via a sampling distribution[6].

2) Networks as Markovian Graphs: We assumed that the graph evolution is Markovian. In a similar context to ours, [23] states that "Markovian evolving graphs are a natural and very general class of models for evolving graphs" and, studies the information spreading protocols on them during the stationary phase. Further, [24] considers information broadcasting methods on Markovian graph processes since they are "general enough for allowing us to model basically any kind of network evolution". Hence, motivated by their generality and natural interpretations, we adopt Markovian graph processes.

3) Functional Dependency of the Sampling Process and Graph Evolution: [25] considers a *weakly adversarial random broadcasting network*: a randomly evolving broadcast network whose state at the next time instant is sampled from a distribution that minimizes the probability of successful communications. Analogous to this, the functional dependency in our model may represent how the network evolves adversely to the influence maximization process (or some other underlying network dynamic which is responsive to the influence maximization). The influence maximizer need not be aware of such dependencies to apply the algorithms that will be presented in the next sections.

4) Extension to Multiple Seeds: In the problem definition and formulation of Sec. II-C, we assumed that only one seed node is selected by sampling with the distribution p_θ. However, the problem formulation and the algorithms presented in this paper can easily be extended to the case where influence maximizer selects a seed set consisting of multiple nodes. For example, assume that the influence maximizer utilizes the distribution p_θ to sample b number of seeds independently. Then, the objective would be to optimize

$$C(\theta) = \mathbb{E}_{G\sim\pi_\theta}\{c(\theta, G)\} \tag{8}$$

where,

$$c(\theta, G) = \mathbb{E}_{\{\mathcal{I}:|\mathcal{I}|=b,\, i \overset{iid}{\sim} p_\theta, \forall i\in\mathcal{I}\}}\{\sigma_G(\mathcal{I}, T)\}. \tag{9}$$

Another possibility is to use a set of sampling distributions $p_{\theta_1}, \dots, p_{\theta_b}$ with $\theta_1, \dots, \theta_b \in [0,1]$ to sample the set of seeds independently. Hence, the formulation and the algorithms in this paper can be extended to optimize any parametric probability distribution defined over the space of seed sets.

III. Stochastic Optimization Method: Perfectly Observed Graph Process with Unknown Dynamics

In this section, we propose a stochastic optimization method for the influence maximizer to recursively estimate the solution of the optimization problem in (6) under the Assumption 1 stated below.

Assumption 1. *The influence maximizer can fully observe the sample paths of the Markovian graph process, but does not know the transition probabilities with which it evolves.*

The schematic overview of the approach for solving (6) is shown in Fig. 1 (where, the HMM filter is not needed in this section due to the Assumption 1). In the next two subsections, the conditional influence estimation algorithm and the stochastic optimization algorithm will be presented.

A. Estimating the Conditional Influence Function using Cohen's Algorithm

The exact computation of the node influence $\sigma_G(i, T)$ in closed form or estimating it with a naive sampling approach is computationally infeasible (as explained in Sec. II-B). As a solution, [13] shows that the shortest path property of the IC model (explained in Sec. II-A) can be used to convert (2) into an expression involving a set of independent random variables as follows:

$$\sigma_G(\mathcal{I}, T) = \mathbb{E}_{\pi_{tt}}\Big\{\sum_{i\in V} \mathbb{1}_{\{g_i(\{\tau_{ji}\}_{(ji)\in E}|\mathcal{I})\le T\}}\Big|\mathcal{I}, G\Big\} \tag{10}$$

where, $g_i(\{\tau_{ji}\}_{(ji)\in E}|\mathcal{I})$ is the shortest path as defined previously in (1). Further, note from (10) that influence of the set $\mathcal{I}$, $\sigma_G(\mathcal{I}, T)$ is the expected T-distance neighborhood (expected number of nodes within T distance from the seed nodes) i.e.

$$\sigma_G(\mathcal{I}, T) = \mathbb{E}_{\pi_{tt}}\big\{\big[|\mathcal{N}_G(\mathcal{I}, T)|\big]\big|\mathcal{I}, G\big\} \tag{11}$$

where,

$$\mathcal{N}_G(\mathcal{I}, T) = \{i \in V : g_i(\{\tau_{ji}\}_{(ji)\in E}|\mathcal{I}) \le T\}. \tag{12}$$

Hence, we only need a neighborhood size estimation algorithm and samples from π_{tt} to estimate the influence $\sigma_G(\mathcal{I}, T)$ of the set $\mathcal{I} \subset V$.

[6] The sampling methods 1 and 2 are also related to a graph theoretic phenomenon named *friendship paradox* [9]. [22] explores how this phenomenon offers a trade-off between the bias and the variance of a randomized polling method (to forecast elections) in social networks. Hence, the tunable sampling scheme $p_\theta(v) = \frac{d(v)^\theta}{\sum_{v\in V} d(v)^\theta}$ where $\theta \in [0,1]$ could offer a useful means of achieving the optimal bias-variance trade off (via a recursive stochastic optimization algorithm) in such application settings as well.

1) Cohen's Algorithm: Based on (11), [13] utilizes a neighborhood size estimation algorithm proposed in [15] in order to obtain an unbiased estimate of influence $\sigma_G(\mathcal{I}, T)$. This algorithm is henceforth referred to as Cohen's algorithm. The main idea behind the Cohen's algorithm is the fact that the minimum of a finite set of unit mean exponential random variables is an exponential random variable with an exponent term equal to the total number random variables in the set. Hence, for a given graph $G = (V, E)$ and a transmission delay set $L_u = \{\tau_{ji}\}^u_{(ji)\in E}$ (where u denotes the index of the set of transmission delays), this algorithm assigns m number of exponential random variable sets $l_{u,j} = \{r_v^{u,j} \sim \exp(1) : v \in V\}$, where, $j = 1, \ldots, m$. Then, a modified Dijkstra's algorithm (refer [15], [13] for a detailed description of the steps of this algorithm) finds the smallest exponential random variable $\bar{r}^{\mathcal{I}}_{u,j}$ within $T-$distance from the set $\mathcal{I}$, for each $l_{u,j}$ with respect to the transmission time set L_u. Then, [15] shows that $\mathbb{E}\{\frac{m-1}{\sum_{j=1}^m \bar{r}^{\mathcal{I}}_{u,j}} \big| L_u\}$ is an unbiased estimate of $\mathcal{N}_G(\mathcal{I}, T)$ conditional on L_u. Further, this Cohen's algorithm for estimating $\sigma_G(\mathcal{I}, T)$ has a lower computational complexity which is near linear in the size of network size, compared to the computational complexity of a naive simulation approach (repeated calling of shortest path algorithm and averaging) to estimate $\sigma_G(\mathcal{I}, T)$ [13].

2) Reduced Variance Estimation of Influence using Cohen's algorithm: We propose Algorithm 1 in order to estimate the conditional influence function $c(\theta, G)$. First four steps of Algorithm 1 are based on a reduced variance version of the Cohen's algorithm employed in [13] called CONTINEST. The unbiasedness and the reduced (compared to the algorithm used in [13]) variance of the estimates obtained using Algorithm 1 are established in Theorem 1.

Theorem 1. *Consider a graph $G = (V, E)$.*

I. Given $v \in V$, $X(v, G)$ in (15) is an unbiased estimate of node influence $\sigma_G(v, T)$, with a variance

$$Var(X(v, G)) \leq \frac{1}{s}\Big(\frac{\sigma_G(v, T)^2}{m-2} + \frac{(m-1)Var(|\mathcal{N}_G(v, T)|)}{m-2}\Big) \tag{17}$$

II. The output $\hat{c}(\theta, G)$ of Algorithm 1 is an unbiased estimate of the conditional influence function $c(\theta, G)$ defined in (5). Variance of $\hat{c}(\theta, G)$ is bounded above by the variance in the case where L_u and $L_{s/2+u}$ (in Step 2 of Algorithm 1) are independently generated.

Proof. See Appendix A. □

Theorem 1 shows that the estimate $X(v, G)$ computed in Algorithm 1 has a smaller variance compared to the estimate computed by the Cohen's algorithm based influence estimation method (named CONTINEST) used in [13] which has a variance of $\frac{1}{s}\big(\frac{\sigma_G(v,T)^2}{m-2} + \frac{(m-1)Var(|\mathcal{N}_G(v,T)|)}{m-2}\big)$. The reason for this reduced variance is the correlation created by using the same set of uniform random numbers (indexed by u) to generate a pair of transmission time sets (L_u and $L_{s/2+u}$). Due to this use of same random number number for multiple realizations, this method is referred to as the method of common random numbers [26]. This reduced variance in the estimates $X(v, G)$ results in a reduced variance in the estimate $\hat{c}(\theta, G)$ of the conditional influence $c(\theta, G)$.

```
Algorithm 1: Conditional influence estimation algorithm
Input: Sampling distribution $p_\theta$, Cumulaive Distribution
       $F_{tt}(\cdot)$ of the transmission time distribution $\pi_{tt}$,
       Directed graph $G = (V, E)$
Output: Estimate of conditional influence function:
        $\hat{c}(\theta, G)$

For all $v \in V$, execute the following steps simultaneously.
  1) Generate $s/2$ sets of uniform random variables:
       $\{\{U^u_{ji}\}_{(ji)\in E} : u = 1, 2, \ldots, s/2\}$
  2) For each $u = 1, \cdots, s/2$, generate a correlated pair of
     random transmission time sets as follows:
       $L_u = \{F^{-1}_{tt}(U^u_{ji})\}_{(ji)\in E}$                  (13)
       $L_{s/2+u} = \{F^{-1}_{tt}(1 - U^u_{ji})\}_{(ji)\in E}$        (14)
  3) For each set $L_u$ where $u = 1, \cdots, s$, assign $m$ sets of
     independent exponential random variables:
     $l_{u,j} = \{r^{u,j}_v \sim \exp(1) : v \in V\}$, $j = 1, \ldots, m$.
  4) Compute the minimum exponential random variable
     $\bar{r}^v_{u,j}$ that is within $T$-distance from $v$ using the modified
     Dijkstra's algorithm, for each $l_{u,j}$. Calculate,
       $X(v, G) = \frac{1}{s}\sum_{u=1}^{s} \frac{m-1}{\sum_{j=1}^m \bar{r}^v_{u,j}}$   (15)
  5) Compute,
       $\hat{c}(\theta, G) = \sum_{v\in V} p_\theta(v) X(v, G).$   (16)
```

Remark 3. In Algorithm 1, the influence of each individual node is computed and then, the estimate $\hat{c}(\theta, G)$ is computed in Step 5 by weighting them according to known p_θ. However, in the case of a set of seeds $\mathcal{I} \subset V$ (discussed in Sec. II-D4), it might not be practically feasible to calculate the influence of each seed set and weight them using the sampling distribution due to the exponentially large number of seed sets. In that case, a particular number of seed sets can be sampled using p_θ first and, the average influence of the sampled seed sets can be used as the estimate $\hat{c}(\theta, G)$. Computing the influence $X(\mathcal{I}, G)$ of a seed set $\mathcal{I}$ can be achieved without significant additional computation (compared to computing the influence $X(v, G)$ of a node v) using least label list approach used in the CONTINEST algorithm [13] and, this would still produce an unbiased estimate.

B. Stochastic Optimization Algorithm

We propose Algorithm 2 for solving the optimization problem (6), utilizing the conditional influence estimates obtained via Algorithm 1. Algorithm 2 is based on the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm (see

Algorithm 2: SPSA based algorithm to estimate θ^*

```
Input: Initial parameterization $\theta_0$, Transmission time distribution $\pi_{tt}$, Observations of the graph process $\{G_n\}_{n\geq 0}$
Output: Estimate of the (locally) optimal solution $\theta^*$ of $C(\theta)$

For $k = 0, 1, \ldots,$ execute the following steps.
  1) Simulate the $M$ dimensional vector $d_k$ with random elements
       $d_k(i) = \begin{cases} +1 & \text{with probability } 0.5 \\ -1 & \text{with probability } 0.5. \end{cases}$
  2) Set $\theta = \theta_k + \Delta d_k$ where, $\Delta > 0$.
  3) Sample a node from the network using $p_\theta$ and, distribute the $2k^{\text{th}}$ message with the sampled node as seed.
  4) Obtain $\hat{c}(\theta, G_n)$ for
     $n = 2kN, 2kN+1, \ldots, 2kN+\bar{N}-1$ using Algorithm 1 and, calculate
       $\hat{C}_k(\theta_k + \Delta d_k) = \frac{1}{\bar{N}} \sum_{n=2kN}^{2kN+\bar{N}-1} \hat{c}(\theta, G_n).$   (18)
  5) Set $\theta = \theta_k - \Delta d_k$. Sample a node from the network using $p_\theta$ and, distribute the $2k+1^{\text{th}}$ message with the sampled node as seed.
  6) Obtain $\hat{c}(\theta, G_n)$ for
     $n = (2k+1)N, (2k+1)N+1, \ldots, (2k+1)N+\bar{N}-1$ using Algorithm 1 and, calculate
       $\hat{C}_k(\theta_k - \Delta d_k) = \frac{1}{\bar{N}} \sum_{n=(2k+1)N}^{(2k+1)N+\bar{N}-1} \hat{c}(\theta, G_n).$
  7) Obtain the gradient estimate,
       $\hat{\nabla} C_k(\theta_k) = \frac{\hat{C}_k(\theta_k + \Delta d_k) - \hat{C}_k(\theta_k - \Delta d_k)}{2\Delta} d_k.$   (19)
  8) Update sampling distribution parameter $\theta_k$ via stochastic gradient algorithm
       $\theta_{k+1} = \theta_k + \epsilon \hat{\nabla} C_k(\theta_k)$   (20)
     where, $\epsilon > 0$.
```

[27], [28] for details). In general, SPSA algorithm utilizes a finite difference estimate $\hat{\nabla} C_k(\theta_k)$, of the gradient of the function $C(\theta)$ at the point θ_k in the (k^{th} iteration of the) recursion,

$$\theta_{k+1} = \theta_k + \epsilon_k \hat{\nabla} C_k(\theta_k). \tag{21}$$

In the k^{th} iteration of the Algorithm 2, the influence maximizer passes a message to the network using a seed node sampled from the distribution p_θ where, $\theta = \theta_k + \Delta d_k$ (step 3). Sampling with p_θ causes the transition matrix to be become P_θ (recall Remark 1). Then, in step 4, (18) averages the conditional influence estimates $\hat{c}(\theta, G_n)$ over $\bar{N}$ consecutive time instants (where, $\bar{N}$ is the length of the available sample path as defined in Sec. II-A) to obtain $\hat{C}_k(\theta_k + \Delta d_k)$, which is an asymptotically convergent estimate (by the law of large numbers for Markov Chains [29], [30]) of $C(\theta_k + \Delta d_k)$. Similarly, steps 5 and 6 obtain $\hat{C}_k(\theta_k - \Delta d_k)$, which is an estimate of $C(\theta_k - \Delta d_k)$. Using these influence function estimates, (19) computes the finite difference gradient estimate $\hat{\nabla} C_k(\theta_k)$ in step 7. Finally, step 8 updates the M-dimensional parameter vector using the gradient estimate computed in (19). Some remarks about this algorithm are as follows.

Remark 4. Algorithm 2 operates in two nested time scales which are as follows (from the fastest to the slowest):

1) t - continuous time scale on which the information diffusion takes place in a given realization of the graph.
2) n - discrete time scale on which the graph evolves.

Further, updating the parameter vector takes place periodically over the scale n, with a period of $2N$ (where, N is the time duration between two messages as defined in Sec. II-A).

Remark 5. Note that all elements of the parameter vector are simultaneously perturbed in the SPSA based approach. Therefore, the parameter vector is updated once every two messages. This is in contrast to other finite difference methods such as Kiefer-Wolfowitz method, which requires $2M$ number of messages (where, M is the dimension of the parameter vector as defined in Sec. II-A) for each update.

Next, we establish the convergence of Algorithm 2 using standard results which gives sufficient conditions for the convergence of recursive stochastic gradient algorithms (for details, see [28], [31], [32]).

Theorem 2. *The sequence $\{\theta_k\}_{k\geq 0}$ in (20) converges weakly to a locally optimal parameter θ^*.*

Proof. See Appendix B. □

IV. Stochastic Optimization Method: Partially Observed Graph Process with Known Dynamics

In this section, we assume that the influence maximizer can observe only a small part of the social network graph at each time instant. The aim of this section is to combine the stochastic optimization framework proposed in Sec. III to this partially observed setting.

A. Partially Observed Graph Process

In some applications, the influence maximizer can observe only a small part of the full network G_n, at any time instant n. Let $G^{\bar{V}}$ denote the subgraph of $G = (V, E)$, induced by the set of nodes $\bar{V} \subseteq V$. Then, we consider the case where, the observable part is the subgraph of G_n which is induced by a fixed subset of nodes $\bar{V} \subseteq V$ i.e. the observable part at time n is $G_n^{\bar{V}}$ ($G^{\bar{V}}$ denotes the subgraph of G, induced by the set of nodes $\bar{V}$)[7]. Then, the observation space $\bar{\mathcal{G}}$ of the Markovian

[7] For example consider the friendship network $G_n = (V, E_n)$ of all the high school students in a city at time n. The smaller observable part could be the friendship network formed by the set of students in a particular high school $V' \subset V$, which is a subgraph of the friendship network G_n. The influence maximizer then needs to perform influence maximization by observing this subgraph.

graph process can be defined as,

$$\bar{\mathcal{G}} = \bigcup_{G\in\mathcal{G}} G^{\bar{V}}, \tag{22}$$

which consists of the subgraphs induced by $\bar{V}$ in each graph $G \in \mathcal{G}$ (the case $\bar{V} = V$ corresponds to the perfectly observed case). For each $G \in \mathcal{G}$ and $\bar{G} \in \bar{\mathcal{G}}$, the observation likelihoods, denoted by $B_{G\bar{G}}$ are defined as,

$$B_{G\bar{G}} = \mathbb{P}(G_n^{\bar{V}} = \bar{G} \,\big|\, G_n = G). \tag{23}$$

In our system model, these observation likelihoods can take only binary values:

$$B_{G\bar{G}} = \begin{cases} 1 & \text{if } G^{\bar{V}} = \bar{G} \\ 0 & \text{otherwise} \end{cases}. \tag{24}$$

i.e. $B_{G\bar{G}} = 1$ if the subgraph formed by the set of nodes $\bar{V}$ in graph $G \in \mathcal{G}$ is $\bar{G} \in \mathcal{Y}$ and, 0 otherwise. In this setting, our main assumption is the following.

Assumption 2. *The measurement likelihood matrix B and the parameterized transition probability matrix P_θ, are known to the influence maximizer but, the (finite) sample paths of the Markovian graph process are observed in noise.*

Then, the aim of the influence maximizer is to recursively solve the problem of influence maximization given in (6), utilizing the information assumed to be known in Assumption 2. However, solving (6) is non-trivial even under the Assumption 2, as emphasized in the following remark.

Remark 6. Even when P_θ is known, it is intractable to compute π_θ in closed form for many cases [33]. Hence, assumption 2 does not imply that a closed form expression of the objective function in (4) can be obtained and hence, solving (6) still remains a non-trivial problem.

B. Randomized Influence Maximization using HMM Filter Estimates

Assumption 2 made in IV-A makes it possible to implement an HMM filter (see [33] for a detailed treatment of HMM filters and related results). The HMM filter is a finite dimensional Bayesian filter which recursively (with each observation) computes π_n^θ which is the probability distribution of the state of the graph, conditional on the sequence of observations $\{G_0^{\bar{V}}, \ldots, G_n^{\bar{V}}\}$. Algorithm 3 gives the HMM filter algorithm and, Theorem 3 establishes the asymptotic convergence of the influence function estimate obtained from from it.

Theorem 3. *The finite sample estimate $\hat{C}_N^\theta$ of the influence function obtained in (27) is an asymptotically unbiased estimate of the influence function $C(\theta)$ i.e.*

$$\lim_{N\to\infty} \mathbb{E}\{\hat{C}_N^\theta\} = C(\theta). \tag{28}$$

Further, $\{\theta_k\}_{k\geq 0}$ in (20) converges weakly to a locally optimal parameter θ^*, when estimates computed in steps 4 and 6 of Algorithm 2 are replaced by the estimates obtained using the using Algorithm 3.

Algorithm 3: Hidden Markov Model (HMM) Filter Algorithm for Tracking the Graph Process

Input: P_θ, B and intial prior π_0
Output: Finite sample estimate $\hat{C}_N^\theta$ of the influence function $C(\theta)$

1) For every time instant $n = 1, 2, \ldots, N-1$, given observation $G_{n+1}^{\bar{V}}$, update the $|\mathcal{G}|$-dimensional posterior:

$$\pi_{n+1}^\theta = T(\pi_n, G_{n+1}^{\bar{V}}) = \frac{B_{G_{n+1}^{\bar{V}}} P_\theta' \pi_n^\theta}{\sigma(\pi_n^\theta, G_{n+1}^{\bar{V}})} \tag{25}$$

where,

$$\sigma(\pi_n^\theta, G_{n+1}^{\bar{V}}) = \mathbf{1}' B_{G_{n+1}^{\bar{V}}} P_\theta' \pi_n^\theta \tag{26}$$

and, $\mathbf{1}$ denotes the column vector with elements equal to one.

2) Compute the estimate of the influence function $C(\theta)$,

$$\hat{C}_N^\theta = \frac{\sum_{n=0}^{N-1} \hat{c}_\theta' \pi_n^\theta}{N}. \tag{27}$$

where, $\hat{c}_\theta$ denotes the column vector with elements $\hat{c}(\theta, G_i)$, $G_i \in \mathcal{G}$.

Proof. See Appendix C. □

Finally, the Algorithm 2 can be modified by replacing the estimates computed in steps 4 and 6 with the influence function estimates obtained using the HMM filter in Algorithm 3.

V. Numerical Results

In this section, we apply the stochastic optimization algorithm presented in Sec. III to two example settings and, illustrate its convergence with a feasible number of iterations.

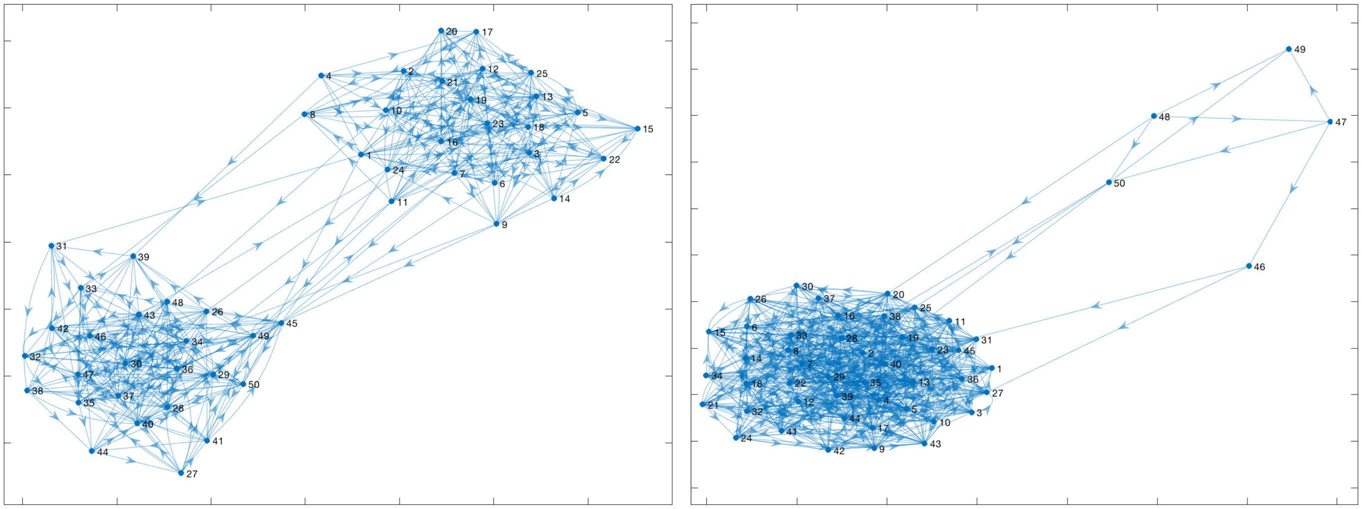
A. Experimental Setup - 1

We use the Stochastic Block Model (SBM) as a generative models to create the graphs used in this section. These models have been widely studied in statistics [34], [35], [36], [37], [38] and network science [39], [40] as generative models that closely resemble the real world networks.

State space of the graph process: We consider the graph process obtained by Markovian switching between the two graphs in Fig. 2: a graph where two dense equal sized clusters exist (Graph G^1) and, a graph where most of the nodes (45 out of 50) are in a single dense cluster (Graph G^2). These graphs are sampled from SBM models with the following parameter values: G^1 with cluster sizes 25, 25 with, within cluster edge probability $p_w^{SBM} = 0.3$, between cluster edge probability $p_b^{SBM} = 0.01$ and, G^2 with cluster sizes 45, 5 with, within cluster edge probability $p_w^{SBM} = 0.3$, between cluster edge probability $p_b^{SBM} = 0.01$. This graph process is motivated by the clustered and non-clustered states of a social network.

Sampling distribution and the graph evolution: We consider the case where the influence maximizer samples from a subset of nodes that consists of the two nodes indexed by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2018.2832011, IEEE Transactions on Signal and Information Processing over Networks



(a) Graph $G^1 = (V, E^1)$ with two equal sized clusters

(b) Graph $G^2 = (V, E^2)$ with a single large cluster

Fig. 2: State space $\mathcal{G}$, of the Markovian graph process $\{G_n\}_{n \geq 0}$ for experimental setup - 1

v_1 and v_2 using the parameterized probability distribution $p_\theta = [\cos^2(\theta) \quad \sin^2(\theta)]'$. These two nodes are located in different clusters in the graphs G^1, G^2. Also, the transition probabilities may depend on this sampling distribution (representing for example, the adversarial networks/nodes as explained in Sec. II-D). However, exact functional characterizations of such dependencies are not known in many practical applications. Also, the form of these dependencies may change over time as well (recall Remark 1). This experimental setup considers a graph process with a stationary distribution $\pi_\theta = p_\theta$ in order to have a closed form influence function as a ground truth (in order to compare the accuracy of the estimates). In an actual implementation of the algorithm, this functional dependency need not be known to the influence maximizer.

Influence Functions: The transmission time distribution was selected to be an exponential distribution with mean 1 for edges within a cluster and, an exponential distribution with mean 10 for between cluster edges (in the SBM). Further, the message expiration time was selected to be $T = 1.5$. Then, the influences of nodes v_1, v_2 on graphs G_1 and G_2 were estimated to be as follows by evaluating the integral in Definition 1 with a naive simulation method (repeated use of the shortest path algorithm): $\sigma_{G^1}(v_1, T) = 25.2$, $\sigma_{G^1}(v_2, T) = 23.2$, $\sigma_{G^2}(v_1, T) = 45.1$, $\sigma_{G^2}(v_2, T) = 5.8$. These values, along with the expressions for p_θ and π_θ were used to obtain the following expression for the influence function (defined in Definition 2) to be compared with the outputs of the algorithm estimates:

$$\begin{aligned} C(\theta) &= \pi_\theta(G^1)(\sigma_{G^1}(v_1, T)p_\theta(v_1) + \sigma_{G^1}(v_2, T)p_\theta(v_2)) \\ &\quad + \pi_\theta(G^2)(\sigma_{G^2}(v_1, T)p_\theta(v_1) + \sigma_{G^2}(v_2, T)p_\theta(v_2)) \\ &= 17.9\sin^2(\theta) - 37.3\sin^4(\theta) + 25.2. \end{aligned} \tag{29}$$

In this context, the goal of our algorithm is to locate the value of θ which maximizes this function, without using knowledge of $C(\theta)$ or π_θ.

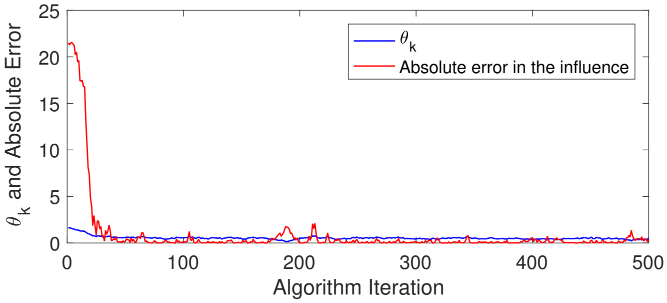


Fig. 3: Parameter value θ_k and the absolute value of the error (difference of current and maximum influence) versus algorithm iteration, showing the convergence of the algorithm in a feasible number of iterations.

Convergence of the Recursive Algorithm for Influence Maximization: Algorithm 2 was utilized in an experimental setting with the parameters specified in the Sec. V-A. For this, the length of the observable sample path of the graph process was assumed to be $\bar{N} = 30$. Further, in the Algorithm 1, the number of transmission time sets (s) and the number of exponential random variable sets (m) were both set to be 10.

With these numerical values for the parameters, Fig. 3 shows the variation of the absolute error (absolute value of the difference between the current and maximum influence) and the parameter value, against the iteration of the algorithm. From this, it can be seen that the algorithm finds optimal parameter in less than 50 iterations. Further, any change in the system model will result in a suboptimal (expected) influence only for 50 iterations since the algorithm is capable of tracking the time evolving optima. Hence, this shows that the stochastic optimization algorithm is capable of estimating the optimal parameter value with a smaller (less than 50) number of iterations.

Effect of variance reduction in convergence and tracking the optima of a time-varying influence function: Here we aim

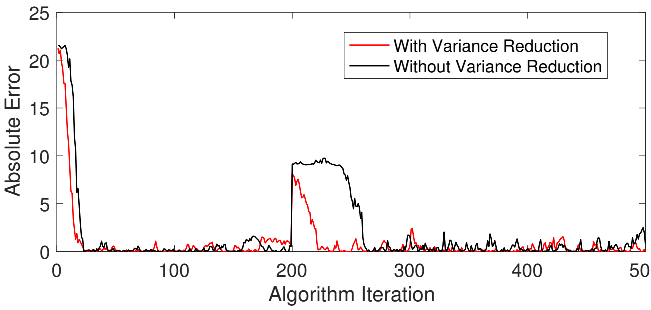


Fig. 4: Absolute value of the error versus algorithm iteration with and without the common random number based variance reduction. This depicts the importance of reduced variance in tracking the optima of a time evolving influence function.

to see how the proposed stochastic approximation algorithm can track the optima when the system model changes on a slower time scale and, the effect of reduced variance Algorithm 1 on the accuracy of tracking. For this, the experimental setup described in V-A was utilized again and, a sudden change in the influence function (by changing the the state space of the graph process and the functional dependency) was introduced at the iteration number 200. In this setting, Fig. 4 depicts the variation of the absolute error in influence with the algorithm iteration for two cases: with the reduced variance Algorithm 1 (red curve) and without the variance reduction approach (black curve). It can be seen from Fig. 4 that the variance reduction method improves the speed of convergence to the optima initially (iterations 1 to 50) and, also in tracking the optima after a sudden change in the influence function (iterations 200 to 300). Further, after convergence (between iterations 50 to 200 and 300 to 500), it can be seen that reduced variance approach is less noisy compared to the method without variance reduction. Hence, this shows that the proposed approach is capable of tracking the optimal sampling distribution in a slowly evolving system such as, varying graph state space, evolving functional dependencies, etc.

B. Experimental Setup - 2

In this subsection, we illustrate the performance of the proposed stochastic optimization algorithm on a graph process generated from a real world network. The aim is to depict how the proposed stochastic optimization algorithm performs in the case of a perfectly observed graph process with characteristics similar to a real world network.

We start with the co-authorship data publicly available at “ftp://ftp.cs.rochester.edu/pub/u/joel/papers.lst”. The co-authorship graph was constructed using this data (for papers produced during 1995-1996 period) and the largest connected component $G = (V, E)$ containing 4660 nodes (Fig. 5) is utilized for our experiments. The probability distribution $p_\theta(v) = \frac{d(v)^\theta}{\sum_{v\in V} d(v)^\theta}, v \in V$ with $\theta \in [0, 1]$ is assumed to be the sampling method available to the influence maximizer (the motivation behind this distribution was discussed in Sec. II-D1).

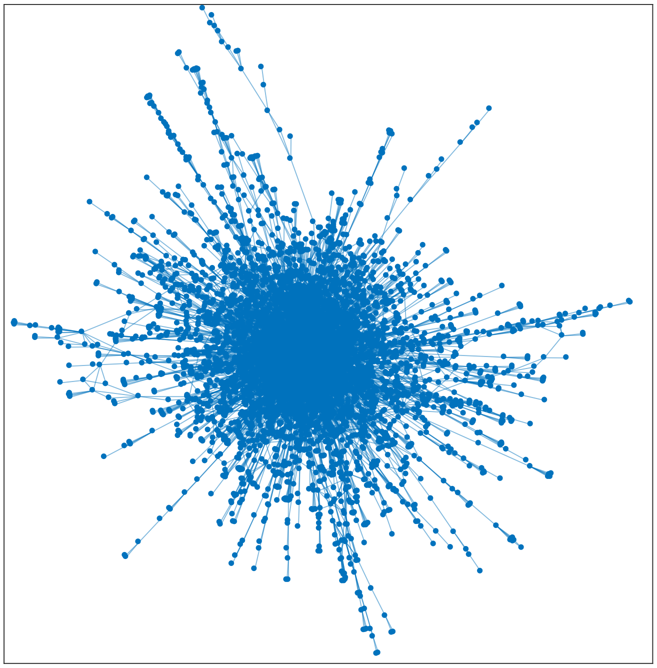


Fig. 5: The largest connected component $G = (V, E)$ of the co-authorship graph of computer science publication during the period of 1995 - 1996. The state space of the Markovian graph process for experimental setup - 2 is obtained by changing the neighbor-degree correlation of $G = (V, E)$ to three different values: -0.2, 0.0, 0.2, using Newman’s edge rewiring procedure.

State space of the graph process: The graph $G = (V, E)$ was modified to obtain three different graphs G_-, G_0 and G_+ that constitute the state space $\mathcal{G}$ of the graph process. These three graphs are obtained by changing the neighbor degree correlation[8] of the original graph G to three different values: $r_{kk}(G_+) = 0.2$, $r_{kk}(G_0) = 0.0$ and $r_{kk}(G_-) = -0.2$, using the Newman’s edge rewiring procedure [41]. Then, in G_-, the nodes with low degrees are centered around high degree nodes while in G_+, high (respectively, low) degree nodes are connected to other high (respectively, low) degree nodes. In G_0, the degrees of the nodes connected by a random link are uncorrelated. Further, since the degree sequence of the original graph is preserved by the edge rewiring procedure, the degree of any node in the above three graphs will remain the same (as in the original graph G). Therefore, the probability $p_\theta(v)$ with which any particular node $v \in V$ is sampled remains same for all three graphs.

Evolution of the graphs: The distribution π_θ (which is functionally dependent on the sampling distribution p_θ) of the graph process was chosen such that the probability assigned to G_+ decreases with θ and the probability assigned to G_- increases with θ. More precisely, π_θ was chosen to be proportional to $[\theta^2 \quad 0.1 \quad 10^{-\theta}]'$.

Motivation for the Experimental Setting: The sampling

[8] Neighbor-degree correlation coefficient $r_{kk}(G)$ of a graph $G = (V, E)$ is the Pearson correlation coefficient of the degrees $d(Y), d(U)$ of the nodes at the end of a randomly chosen link (Y, U) from the set of edges G [41].

distribution $p_\theta(v) = \frac{d(v)^\theta}{\sum_{v \in V'} d(v)^\theta}, v \in V$ is a trade off between two widely used sampling distributions: sampling a random node from a random link ($\theta = 1$) and uniform sampling of nodes ($\theta = 0$). Further, when θ is closer to 1, the sampling will be biased towards high degree nodes in V and when θ is closer to 0, it will be closer to a uniform distribution over the set of nodes V'. Then, π_θ can be interpreted as a graph process which tries to prevent the diffusion of information (adversarial graph process): as θ increases from 0 to 1 (biasing the sampling towards high degree nodes), the probability $\pi_\theta(G_-)$ of G_- increases (and the probability $\pi_\theta(G_+)$ of G_+ decreases) to separate the high degree nodes from each other (and thus preventing a rapid diffusion of information).

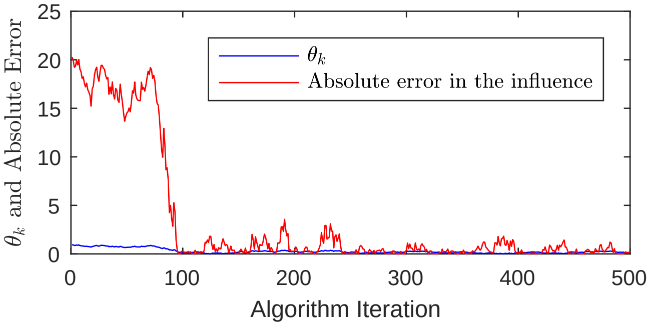


Fig. 6: Parameter value θ_k and the absolute value of the error (difference of current and maximum influence) versus algorithm iteration for the experimental setting - 2, showing the convergence of the algorithm in a feasible number of iterations.

Experimental Results: Experiments on the setup described above were performed. The true influence function $C(\theta)$ was estimated using a naive simulation method in order to compare the performance of the method. The results of the experiments are shown in Fig. 6. It can be seen that the algorithm is capable of locating the optimal sampling distribution in less than 100 iterations of the algorithm. It can be seen that the convergence of the algorithm in this case is slower and has a larger variance compared to the experimental setting - 1 (less than 50 iterations) where the graph contained only 50 nodes. However, considering the fact that the size of the graph in the experimental setting - 2 is two orders of magnitude larger compared to experimental setting - 1, it is a reasonable trade-off in the performance of the algorithm.

VI. Conclusion

This paper considered the problem of randomized influence maximization over a Markovian Graph Process: given a fixed set of nodes whose connectivity graph is evolving as a Markov chain, estimate the probability distribution (over this fixed set of nodes) that samples a node which will initiate the largest information cascade (in expectation). The evolution of the graph was allowed to functionally depend on the sampling probability distribution in order to keep the problem more general. This was formulated as a problem of tracking the optimal solution of a (time-varying) optimization problem where, a closed form expression of the objective function (influence function) is not available. In this setting, two stochastic gradient algorithms were presented to estimate the optimal sampling distribution for two cases: 1) transition probabilities of the graph are unknown but, the graph can be observed perfectly 2) transition probabilities of the graph are known but, the graph is observed in noise. These algorithms are based on the Simultaneous Perturbation Stochastic Approximation Algorithm that requires only the noisy estimates of the influence function. These noisy estimates of the influence function were obtained by combining a neighborhood size estimation algorithm with a variance reduction method and then, averaging over a finite sample path of the graph process. The convergence of the proposed methods were established theoretically and, illustrated with numerical examples. The numerical results show that, with the reduced variance approach, the algorithms are capable of tracking the optimal influence even in a time varying system model (with changing graph state spaces, etc.).

Appendix A
Proof of Theorem 1

Let the size of the $T-$distance neighborhood $\mathcal{N}_G(v,T)$ of a node $v \in V$ of graph $G = (V,E)$ conditional on a transmission time set $L_u = \{\tau_{ji}\}^u_{(ji)\in E}$ be denoted by $h^v(L_u)$. Further, Let $\hat{h}^v(L_u)$ denote $\frac{m-1}{\sum_{j=1}^m \bar{r}^v_{u,j}}$.

$$\mathbb{E}\{X(v,G)\} = \mathbb{E}\left\{\frac{1}{s}\sum_{u=1}^{s}\frac{m-1}{\sum_{j=1}^m \bar{r}^v_{u,j}}\right\} \tag{30}$$

$$= \frac{1}{s}\sum_{u=1}^{s}\mathbb{E}\{\hat{h}^v(L_u)\} \tag{31}$$

$$= \frac{1}{s}\sum_{u=1}^{s}\mathbb{E}\{\mathbb{E}\{\hat{h}^v(L_u)|L_u\}\} \tag{32}$$

$$= \frac{1}{s}\sum_{u=1}^{s}\mathbb{E}\{h^v(L_u)\} \tag{33}$$

(from conditional unbiasedness proved in [15])

$$= \sigma_G(v,T) \text{ (by (11))} \tag{34}$$

To analyze the variance of $X(v,G)$, first note that $h^v(\{\tau_{ji}\}^u_{(ji)\in E})$ is a monotonically decreasing function of all its elements $\{\tau_{ji}\}^u_{(ji)\in E}$. Further, the following result about monotone functions of random variables from [26] will be used to establish the result.

Lemma 4. *If $g(x_1,...,x_n)$ is a monotone function of each of its arguments, then, for a set $U_1,...,U_n$ of independent random numbers.*

$$\text{Cov}(g(U_1,...,U_n), g(1-U_1,...,1-U_n)) \leq 0. \tag{35}$$

Now, consider the variance of $\frac{\hat{h}^v(L_u)+h(L_{s/2+u})}{2}$ where L_u and $L_{s/2+u}$ are the pair of correlated transmission time sets

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSIPN.2018.2832011, IEEE Transactions on Signal and Information Processing over Networks

as defined in (13) and (14).

$$\mathrm{Var}\left(\frac{\hat{h}^v(L_u)+\hat{h}(L_{s/2+u})}{2}\right) = \frac{1}{4}\Bigg(\mathrm{Var}\left(\hat{h}^v(L_u)\right) + \mathrm{Var}\left(\hat{h}^v(L_{s/2+u})\right) + 2\,\mathrm{Cov}\left(\hat{h}^v(L_u),\hat{h}^v(L_{s/2+u})\right)\Bigg) \quad (36)$$

$$= \frac{1}{2}\Bigg(\mathrm{Var}\left(\hat{h}^v(L_u)\right) + \mathrm{Cov}\left(\hat{h}^v(L_u),\hat{h}^v(L_{s/2+u})\right)\Bigg) \quad (37)$$

(since $\hat{h}^v(L_u)$ and $\hat{h}^v(L_{s/2+u})$ are identically distributed)

Now consider $\mathrm{Cov}\left(\hat{h}^v(L_u),\hat{h}^v(L_{s/2+u})\right)$. By using the law of total covariance,

$$\mathrm{Cov}\left(\hat{h}^v(L_u),\hat{h}^v(L_{s/2+u})\right) = \mathbb{E}\big\{\,\mathrm{Cov}\left(\hat{h}^v(L_u),\hat{h}^v(L_{s/2+u})\big|\{U^u_{ji}\}_{(ji)\in E}\right)\big\} + \mathrm{Cov}\left(\mathbb{E}\{\hat{h}^v(L_u)|\{U^u_{ji}\}_{(ji)\in E}\},\mathbb{E}\{\hat{h}^v(L_{s/2+u})|\{U^u_{ji}\}_{(ji)\in E}\}\right) \quad (38)$$

$$= \mathrm{Cov}\left(\mathbb{E}\{\hat{h}^v(L_u)|\{U^u_{ji}\}_{(ji)\in E}\},\mathbb{E}\{\hat{h}^v(L_{s/2+u})|\{U^u_{ji}\}_{(ji)\in E}\}\ \right) \quad (39)$$

(since $\hat{h}^v(L_u)$ and $\hat{h}^v(L_{s/2+u})$ are uncorrelated given $\{U^u_{ji}\}_{(ji)\in E}$) (40)

$$= \mathrm{Cov}\left(h^v(L_u),h^v(L_{s/2+u})\right) \quad (41)$$

(from the conditional unbiasedness proved in [15])

$$= \mathrm{Cov}\left(h^v(\{F^{-1}_{tt}(U^u_{ji})\}_{(ji)\in E}),h^v(\{F^{-1}_{tt}(1-U^u_{ji})\}_{(ji)\in E})\right) \quad (42)$$

$F^{-1}_{tt}(\cdot)$ is a monotone function (inverse of a CDF) and, $h^v(\{\tau_{ji}\}^u_{(ji)\in E})$ is also monotone in all its arguments $\{\tau_{ji}\}^u_{(ji)\in E}$. Hence, the composite function $h^v(\{F^{-1}_{tt}(U^u_{ji})\}_{(ji)\in E})$ is monotone is all its arguments $\{U^u_{ji}\}_{(ji)\in E}$ (because, the composition of monotone functions is monotone). Then, from Lemma 4, it follows that

$$\mathrm{Cov}\left(\hat{h}^v(L_u),\hat{h}^v(L_{s/2+u})\right) \leq 0. \quad (43)$$

Then, from (37), it follows that,

$$\mathrm{Var}\left(\frac{\hat{h}^v(L_u)+\hat{h}(L_{s/2+u})}{2}\right) \leq \frac{1}{2}\,\mathrm{Var}\left(\hat{h}^v(L_u)\right.. \quad (44)$$

Then, by applying the total variance formula to the left hand side of (44) and, using the fact $\mathrm{Var}\{\hat{h}^v(L_u)|L_u\} = \frac{h^2(L_u)}{m-2}$(from [15]), we get,

$$\mathrm{Var}\left(\frac{\hat{h}^v(L_u)+\hat{h}(L_{s/2+u})}{2}\right) \leq \frac{1}{2}\left(\frac{\sigma_G(v,T)^2}{m-2} + \frac{(m-1)\,\mathrm{Var}(|\mathcal{N}_G(v,T)|)}{m-2}\right) \quad (45)$$

and, the proof follows by noting that $X(v,G)$ is the average of $\frac{\hat{h}^v(L_u)+h(L_{s/2+u})}{2}$ for $u=1,\cdots,s/2$.

Proof of Part II of Theorem 1 follows from similar arguments as above and hence omitted.

Appendix B
Proof of Theorem 2

The following result from [31] will be used to establish the weak convergence of the sequence $\{\theta_k\}_{k\geq 0}$ obtained in Algorithm 2.

Consider the stochastic approximation algorithm,

$$\theta_{k+1} = \theta_k + \epsilon H(\theta_k, x_k), \quad k = 0,1,\ldots \quad (46)$$

where $\epsilon > 0$, $\{x_k\}$ is a random process and, $\theta \in \mathbb{R}^p$ is the estimate generated at time $k=0,1,\ldots$. Further, let

$$\theta^\epsilon(t) = \theta_k \text{ for } t \in [k\epsilon, k\epsilon+\epsilon], \quad k = 0,1,\ldots, \quad (47)$$

which is a piecewise constant interpolation of $\{\theta_k\}$. In this setting, the following result holds.

Theorem 5. *Consider the stochastic approximation algorithm (46). Assume*

SA1: *$H(\theta,x)$ us uniformly bounded for all $\theta\in\mathbb{R}^p$ and $x\in\mathbb{R}^q$.*

SA2: *For any $l\geq 0$, there exists $h(\theta)$ such that*

$$\frac{1}{N}\sum_{k=l}^{N+l-1}\mathbb{E}_l\{H(\theta,x_k)\} \rightarrow h(\theta) \text{ as } N\rightarrow\infty. \quad (48)$$

where, $\mathbb{E}_l\{\cdot\}$ denotes expectation with respect to the sigma algebra generated by $\{x_k : k < l\}$.

SA3: *The ordinary differential equation (ODE)*

$$\frac{d\theta(t)}{dt} = h(\theta(t)), \quad \theta(0)=\theta_0 \quad (49)$$

has a unique solution for every initial condition.

Then, the interpolated estimates $\theta^\epsilon(t)$ defined in (47) satisfies

$$\lim_{\epsilon\rightarrow 0}\mathbb{P}\big(\sup_{0\leq t\leq T}|\theta^\epsilon(t)-\theta(t)|\geq\eta\big) = 0 \textit{ for all } T>0,\eta>0 \quad (50)$$

where, $\theta(t)$ is the solution of the ODE (49).

The condition SA1 in Theorem 5 can be replaced by uniform integrability and the result still holds [33].

Next, we show how Algorithm 2 fulfills the assumptions SA1, SA2, SA3 in Theorem 5. Detailed steps of similar proofs related to stochastic approximation algorithms can be found in [42] and Chapter 17 of [33].

Consider $\sup_k \mathbb{E}||\hat{\nabla} C_k(\theta_k)||$ where, $\hat{\nabla} C_k(\theta_k)$ is as defined in Algorithm 2.

$$\sup_k \mathbb{E}||\hat{\nabla} C_k(\theta_k)|| = \sup_k \mathbb{E}\Bigg|\Bigg|\Bigg(\frac{1}{\bar{N}}\sum_{n=2kN}^{2kN+\bar{N}-1} \hat{c}(\theta, G_n) \tag{51}$$

$$- \frac{1}{\bar{N}}\sum_{n=(2k+1)N}^{(2k+1)N+\bar{N}-1} \hat{c}(\theta, G_n)\Bigg)\frac{d_k}{2\Delta}\Bigg|\Bigg|$$

$$= \sup_k \frac{\sqrt{M}}{2\Delta\bar{N}}\mathbb{E}\Bigg|\sum_{n=2kN}^{2kN+\bar{N}-1} \hat{c}(\theta, G_n) \tag{52}$$

$$- \sum_{n=(2k+1)N}^{(2k+1)N+\bar{N}-1} \hat{c}(\theta, G_n)\Bigg| \tag{53}$$

$$\leq \sup_k \frac{\sqrt{M}}{2\Delta\bar{N}}\mathbb{E}\Bigg\{\sum_{n=2kN}^{(2k+1)N+\bar{N}-1} \big|\hat{c}(\theta, G_n)\big|\Bigg\} \tag{54}$$

(By triangle inequality)

$$= \sup_k \frac{\sqrt{M}}{2\Delta\bar{N}}\sum_{n=2kN}^{(2k+1)N+\bar{N}-1} \mathbb{E}\{\hat{c}(\theta, G_n)\} \tag{55}$$

(Since $\hat{c}(\theta, G_n) \geq 0$)

$$= \sup_k \frac{\sqrt{M}}{2\Delta\bar{N}}\sum_{n=2kN}^{(2k+1)N+\bar{N}-1} \mathbb{E}\{c(\theta, G_n)\} \tag{56}$$

(Conditioning on G_n and,
using Part II of Theorem 1)

$$\leq \sup_k \frac{\sqrt{M}}{2\Delta\bar{N}}\sum_{n=2kN}^{(2k+1)N+\bar{N}-1} \{\max_{v\in V, G\in\mathcal{G}} \sigma_G(v,T)\} \tag{57}$$

$$= \frac{\sqrt{M}}{\Delta}\max_{v\in V, G\in\mathcal{G}} \sigma_G(v,T) \tag{58}$$

(maximum exists since $V, \mathcal{G}$ are finite sets)

Hence the uniform integrability condition (alternative for SA1) is fulfilled.

Next, by the law of large numbers, as Δ (perturbation size in (19)) tends to zero, $\hat{\nabla} C_k(\theta_k)$ in Algorithm 2 fulfills the SA2 condition.

SA3 is fulfilled by the (global) Lipschitz continuity of the gradient $\nabla_\theta C(\theta)$ which is a sufficient condition for the existence of a unique solution for a non-linear ODE (for any initial condition) [43].

APPENDIX C
PROOF OF THEOREM 3

Consider the expected value of the finite sample estimate of the influence function (output of Algorithm 3)

$$\mathbb{E}\{\hat{C}_N^\theta\} = \mathbb{E}\{\frac{\sum_{n=1}^N \hat{c}^T \pi_n^\theta}{N}\} \tag{59}$$

$$= \frac{c^T}{N}\sum_{n=1}^N \mathbb{E}\{\pi_n^\theta\} \tag{60}$$

(by Theorem 1)

$$= \frac{c^T}{N}\sum_{n=1}^N ({P_\theta}^n)^T \pi_0 \tag{61}$$

Then, the result follows by noting that (61) is a Cesàro summation of a convergent sequence.

The weak convergence of the sequence $\{\theta_k\}_{k\geq 0}$ (with HMM filter estimates) follows again from Theorem 5. Since the proof is mostly similar to the Proof of Theorem 2, we skip the proof.

REFERENCES

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, ACM, 2003.

[2] M. Hamdi, V. Krishnamurthy, and G. Yin, "Tracking a Markov-modulated stationary degree distribution of a dynamic random graph," *IEEE Transactions on Information Theory*, vol. 60, no. 10, pp. 6609–6625, 2014.

[3] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International Workshop on Web and Internet Economics*, pp. 306–311, Springer, 2007.

[4] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks.," in *WINE*, vol. 6484, pp. 539–550, Springer, 2010.

[5] T. Carnes, C. Nagarajan, S. M. Wild, and A. Van Zuylen, "Maximizing influence in a competitive social network: a follower's perspective," in *Proceedings of the ninth international conference on Electronic commerce*, pp. 351–360, ACM, 2007.

[6] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate," in *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 379–390, SIAM, 2011.

[7] L. Seeman and Y. Singer, "Adaptive seeding in social networks," in *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pp. 459–468, IEEE, 2013.

[8] T. Horel and Y. Singer, "Scalable methods for adaptively seeding a social network," in *Proceedings of the 24th International Conference on World Wide Web*, pp. 441–451, International World Wide Web Conferences Steering Committee, 2015.

[9] S. L. Feld, "Why your friends have more friends than you do," *American Journal of Sociology*, vol. 96, no. 6, pp. 1464–1477, 1991.

[10] S. Lattanzi and Y. Singer, "The power of random neighbors in social networks," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pp. 77–86, ACM, 2015.

[11] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208, ACM, 2009.

[12] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1029–1038, ACM, 2010.

[13] N. Du, Y. Liang, M.-F. Balcan, M. Gomez-Rodriguez, H. Zha, and L. Song, "Scalable influence maximization for multiple products in continuous-time diffusion networks," *Journal of Machine Learning Research*, vol. 18, no. 2, pp. 1–45, 2017.

[14] H. Zhuang, Y. Sun, J. Tang, J. Zhang, and X. Sun, "Influence maximization in dynamic social networks," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pp. 1313–1318, IEEE, 2013.

[15] E. Cohen, "Size-estimation framework with applications to transitive closure and reachability," *Journal of Computer and System Sciences*, vol. 55, no. 3, pp. 441–453, 1997.

[16] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pp. 88–97, IEEE, 2010.

[17] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," *arXiv preprint arXiv:1105.0697*, 2011.

[18] V. Krishnamurthy, O. N. Gharehshiran, M. Hamdi, *et al.*, "Interactive sensing and decision making in social networks," *Foundations and Trends® in Signal Processing*, vol. 7, no. 1-2, pp. 1–196, 2014.

[19] A. H. Sayed *et al.*, "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[20] O. N. Gharehshiran, V. Krishnamurthy, and G. Yin, "Distributed tracking of correlated equilibria in regime switching noncooperative games," *IEEE Transactions on Automatic Control*, vol. 58, no. 10, pp. 2435–2450, 2013.

[21] M. G. Rodriguez and B. Schölkopf, "Influence maximization in continuous time diffusion networks," *arXiv preprint arXiv:1205.1682*, 2012.

[22] B. Nettasinghe and V. Krishnamurthy, "What do your friends think? efficient polling methods for networks using friendship paradox," *arXiv preprint arXiv:1802.06505*, 2018.

[23] A. E. Clementi, F. Pasquale, A. Monti, and R. Silvestri, "Information spreading in stationary Markovian evolving graphs," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pp. 1–12, IEEE, 2009.

[24] A. Clementi, P. Crescenzi, C. Doerr, P. Fraigniaud, F. Pasquale, and R. Silvestri, "Rumor spreading in random evolving graphs," *Random Structures & Algorithms*, vol. 48, no. 2, pp. 290–312, 2016.

[25] A. E. Clementi, A. Monti, F. Pasquale, and R. Silvestri, "Broadcasting in dynamic radio networks," *Journal of Computer and System Sciences*, vol. 75, no. 4, pp. 213–230, 2009.

[26] S. M. Ross, *Simulation (5th Edition)*. Elsevier, 2013.

[27] J. C. Spall, "Multivariate stochastic approximation using a simultaneous perturbation gradient approximation," *IEEE transactions on automatic control*, vol. 37, no. 3, pp. 332–341, 1992.

[28] J. C. Spall, "Simultaneous perturbation stochastic approximation," *Introduction to Stochastic Search and Optimization: Estimation, Simulation, and Control*, pp. 176–207, 2003.

[29] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2010.

[30] J. R. Norris, *Markov chains*. No. 2, Cambridge university press, 1998.

[31] H. J. Kushner and G. Yin, *Stochastic approximation and recursive algorithms and applications*. No. 35 in Applications of mathematics, New York: Springer, 2003.

[32] V. Krishnamurthy, M. Maskery, and G. Yin, "Decentralized adaptive filtering algorithms for sensor activation in an unattended ground sensor network," *IEEE Transactions on Signal Processing*, vol. 56, no. 12, pp. 6086–6101, 2008.

[33] V. Krishnamurthy, *Partially Observed Markov Decision Processes*. Cambridge University Press, 2016.

[34] Y. Zhao, E. Levina, J. Zhu, *et al.*, "Consistency of community detection in networks under degree-corrected stochastic block models," *The Annals of Statistics*, vol. 40, no. 4, pp. 2266–2292, 2012.

[35] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *IEEE Transactions on Information Theory*, vol. 62, no. 1, pp. 471–487, 2016.

[36] K. Rohe, S. Chatterjee, B. Yu, *et al.*, "Spectral clustering and the high-dimensional stochastic blockmodel," *The Annals of Statistics*, vol. 39, no. 4, pp. 1878–1915, 2011.

[37] M. Lelarge, L. Massoulié, and J. Xu, "Reconstruction in the labelled stochastic block model," *IEEE Transactions on Network Science and Engineering*, vol. 2, no. 4, pp. 152–163, 2015.

[38] D. E. Fishkind, D. L. Sussman, M. Tang, J. T. Vogelstein, and C. E. Priebe, "Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown," *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 23–39, 2013.

[39] B. Karrer and M. E. Newman, "Stochastic blockmodels and community structure in networks," *Physical Review E*, vol. 83, no. 1, p. 016107, 2011.

[40] B. Wilder, N. I. E. Rice, and M. Tambe, "Influence maximization with an unknown network by exploiting community structure," 2017.

[41] M. E. Newman, "Assortative mixing in networks," *Physical review letters*, vol. 89, no. 20, p. 208701, 2002.

[42] V. Krishnamurthy and G. G. Yin, "Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime," *IEEE Transactions on Information Theory*, vol. 48, no. 2, pp. 458–476, 2002.

[43] H. K. Khalil, "Noninear systems," *Prentice-Hall, New Jersey*, vol. 2, no. 5, pp. 5–1, 1996.