Maximizing Influence in an Unknown Social Network

Bryan Wilder[12], **Nicole Immorlica**[3], **Eric Rice**[24], **Milind Tambe**[12]
[1]Department of Computer Science, [2]Center for Artificial Intelligence in Society [4]School of Social Work
University of Southern California
{bwilder, ericr, tambe}@usc.edu
[3]Microsoft Research, New England
nicimm@gmail.com

Abstract

In many real world applications of influence maximization, practitioners intervene in a population whose social structure is initially unknown. This poses a multiagent systems challenge to act under uncertainty about how the agents are connected. We formalize this problem by introducing *exploratory influence maximization*, in which an algorithm queries individual network nodes (agents) to learn their links. The goal is to locate a seed set nearly as influential as the global optimum using very few queries. We show that this problem is intractable for general graphs. However, real world networks typically have community structure, where nodes are arranged in densely connected subgroups. We present the ARISEN algorithm, which leverages community structure to find an influential seed set. Experiments on real world networks of homeless youth, village populations in India, and others demonstrate ARISEN's strong empirical performance. To formally demonstrate how ARISEN exploits community structure, we prove an approximation guarantee for ARISEN on graphs drawn from the Stochastic Block Model.

1 Introduction

In contexts ranging from health to international development, practitioners have used the social network of their target population to spread information and change behavior. Social interactions between population members form a multiagent system; the challenge is identify the most influential agents. While previous work has delivered computationally efficient algorithms for this *influence maximization* problem (Chen, Wang, and Wang 2010; Tang, Xiao, and Shi 2014; Yadav et al. 2016), this work assumes that the social network is given explicitly as input. However, in many real-world domains, the network is not initially known and must be gathered via laborious field observations. For example, collecting network data from vulnerable populations such as homeless youth, while crucial for health interventions, requires significant time spent gathering field observations (Rice et al. 2012). Social media data is often unavailable when access to technology is limited, for instance in developing countries or with vulnerable populations. Even when such data is available, it often includes many weak links which are not effective at spreading influence (Bond et al. 2012). For instance, a person may have hundreds of Facebook friends whom they barely know. In principle, the entire network could be reconstructed via surveys, and then existing influence maximization algorithms applied. However, exhaustive surveys are very labor-intensive and often considered impractical (Valente and Pumpuang 2007). For influence maximization to be relevant to many real-world problems, it must contend with limited *information* about the network, not just limited *computation*.

The major informational restriction is the number of nodes which may be surveyed to explore the network. Thus, a key question is: *how can we find influential nodes with a small number of queries?* We formalize this problem as *exploratory influence maximization* and seek a principled algorithmic solution, i.e., an algorithm which makes a small number of queries and returns a set of seed nodes which are approximately as influential as the globally optimal seed set. Existing field work uses heuristics, such as sampling some percentage of the nodes and asking them to nominate influencers (Valente and Pumpuang 2007). To our knowledge, no previous work directly addresses this question from an algorithmic perspective (see Section 3).

We show that for general graphs, any algorithm for exploratory influence maximization may perform arbitrarily badly unless it examines almost the entire network. However, real world networks often have strong *community* structure, where nodes form tightly connected subgroups which are only weakly connected to the rest of the network (Leskovec et al. 2009). Consequently, influence mostly propagates locally. Community structure has been used to develop computationally efficient influence maximization algorithms (Wang et al. 2010; Chen et al. 2014). Here, we use it to design a highly information-efficient algorithm. We make four contributions. *First*, we introduce exploratory influence maximization and show that it is intractable for general graphs. *Second*, we present the ARISEN algorithm, which exploits community structure to find influential nodes. *Third*, we show that ARISEN has strong empirical performance on an array of real world social networks. *Fourth*, we formally analyze ARISEN on graphs drawn from the Stochastic Block Model (SBM) (Fienberg and Wasserman 1981), a widely studied model of community structure. We prove that it approximates the optimal influence if the entire network were known by querying only a polylogarith-

mic number of nodes in the network size.

2 Exploratory influence maximization

As a motivating example, consider a homeless youth shelter which wishes to spread HIV prevention information (Rice et al. 2012). The shelter would try to select the most influential peer leaders to spread information, but the youths' social network is not initially known. Constructing the network requires a laborious survey (Rice et al. 2012). Our motivation is to mitigate this effort by querying only a few youth. Such queries require much less time than the day-long training peer leaders receive. We now formalize this problem.

Influence maximization: The influence maximization problem (Kempe, Kleinberg, and Tardos 2003), starts with a graph $G = (V, E)$, where $|V| = n$ and $|E| = m$. We assume that G is undirected; social links are typically reciprocal (Squartini et al. 2012). An influencer selects K seed nodes, aiming to maximize the expected size of the resulting influence cascade. We assume that influence propagates according to the independent cascade model (ICM), the most prevalent model in the literature. Initially, all nodes are inactive except for the seeds. When a node activates, it independently activates each of its neighbors with probability q. q is often assumed to be the same for all edges (Chen, Wang, and Wang 2010; Kempe, Kleinberg, and Tardos 2003; Yadav et al. 2016). Let $f(S)$ denote the expected number of activated nodes with seed set $S \subseteq V$. The objective is to compute $\arg\max_{|S|\leq K} f(S)$.

Local information: The edge set E is not initially known. Instead, the algorithm explores portions of the graph using local operations. We use the popular "Jump-Crawl" model (Brautbar and Kearns 2010), where the algorithm may either jump to a uniformly random node, or crawl along an edge from an already surveyed node to one of its neighbors. When visited, a node reveals all of its edges. We say that the *query cost* of an algorithm is the total number of nodes visited using either operation. Our goal is to find influential nodes with a query cost that is much less than n, the total number of nodes.

Stochastic Block Model: In our formal analysis, we assume that the graph is drawn from the SBM. The SBM originated in sociology (Fienberg and Wasserman 1981) and lately has been intensively studied in computer science and statistics (see e.g. (Abbe and Sandon 2015; Krzakala et al. 2013; Mossel, Neeman, and Sly 2015)). In the SBM, the network is partitioned into disjoint communities $C_1 ... C_L$. Each within-community edge is present independently with probability p_w and each between-community edge is present independently with probability p_b. Recall that the Erdős-Rényi random graph $\mathcal{G}(n, p)$ is the graph on n nodes where every edge is independently present with probability p. In the SBM, community C_i is internally drawn as $\mathcal{G}(|C_i|, p_w)$ with additional random edges to other communities. While the SBM is a simplified model, our experimental results show that ARISEN also performs well on real-world graphs. ARISEN takes as input the parameters n, p_w, and p_b, but is not given any prior information about the realized draw of the network. It is reasonable to assume that the model parameters are known since they can be estimated using existing network data from a similar population (in our experiments, we show that this approach works well). For instance in HIV prevention, homeless youth social networks have been shown to exhibit community structure and several studies have gathered networks from which to infer p_w and p_b (Yadav et al. 2016; Rice et al. 2012).

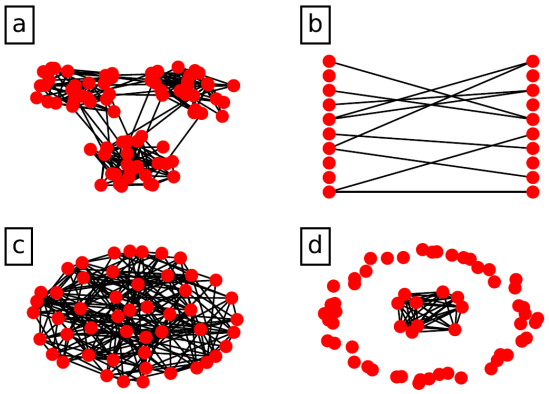

Figure 1: Example SBM networks. (a) A community structured network ($p_w = 0.1$, $p_b = 0.005$). (b) A bipartite graph (2 communities, $p_w = 0$, $p_b = 0.1$). (c) An Erdős-Rényi graph (1 community, $p_w = 0.2$). (d) One small community with $p_w = 1$; the rest in a community with $p_w = 0, p_b = 0$.

Our theoretical analysis will use a particular range of values for p_w and p_b. As formally defined, the SBM encompasses a wide range of possible topologies, depending on how the parameters p_w and p_b are set. Figure 1 gives a few examples, ranging from the a bipartite graph to an Erdős-Rényi graph. The community-structure graph that we intend to model is Figure 1(a). We later define a parameter range which produces such networks.

Objective: We now formalize the objective that our algorithm will optimize. We compare to the globally optimal solution, i.e, the best performance if the entire network were known. Let $f_E(S)$ give the expected number of nodes influenced by seed set S when the set of realized edges are E. Let $\mathcal{A}(E)$ be the (possibly random) seed set containing our algorithm's selections given edge set E. Let OPT be the expected value of the globally optimal solution which seeds K nodes. We aim to prove that $\mathbb{E}[f_E(\mathcal{A}(E))] \geq \alpha OPT$ for some approximation ratio α, where the expectation is over the randomness in the graph, the algorithm's choices, and the ICM.

Hardness result: We seek algorithms whose query cost grows slowly with n. The following shows that no algorithm with strictly sublinear query cost obtains a constant factor approximation for general graphs. The notation $o(1)$ refers to a term which goes to 0 as $n \to \infty$.

Theorem 1. *There exists a family of graphs on which any algorithm with query cost $O(n^{1-\epsilon})$ for some $\epsilon > 0$ has approximation ratio no better than $o(1)$.*

Proof. Consider a family of graphs which consist of a clique on $\log n$ nodes along with $n - \log n$ isolated nodes. Let

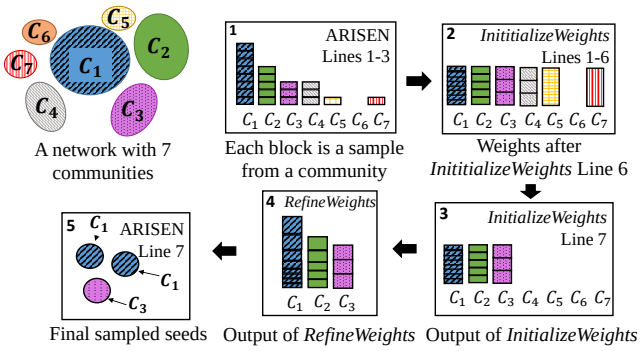

Figure 2: Example run of ARISEN with $K = 3$ (explained further in text). Each block is one sample, with current weight proportional to its height (e.g., in Frame 2, C_5 has one sample with very high weight).

$q = 1$ and $K = 1$. The algorithm gets influence $\log n$ if it selects a node in the clique, and influence 1 otherwise. The probability it ever samples the clique is at most $1 - (1 - \frac{\log n}{n})^{O(n^{1-\epsilon})} \leq 1 - e^{-\frac{\log n}{O(n^{\epsilon})}}(1 - \frac{\log^2 n}{n})^{O(n^{\epsilon})} = o(1)$. Hence, its expected influence is $o(1)\log n + 1$, while OPT is $\log n$, giving approximation ratio $\frac{o(1)\log n+1}{\log n} = o(1)$. □

3 Related work

First, Yadav et al. (2016) and Wilder et al. (2017) studied dynamic influence maximization over a series of rounds. Some edges are "uncertain" and are only present with some probability; the algorithm can gain information about these edges in each round. However, most edges are known in advance. By contrast, our work does not require *any* known edges. Mihara et al. (2015) also consider influence maximization over a series of rounds, but in their work the network is initially unknown. In each round, the algorithm makes some queries, selects some seed nodes, and observes all of the nodes which are activated by its chosen seeds. The ability to observe activated nodes makes our problem incomparable with theirs because activations can reveal a great deal about the network and give the algorithm information that even their benchmark does not have. Further, activations are unobservable in many domains (e.g. medical ones) for privacy and legal reasons. Carpentier and Valko (2016) study a bandit setting where the algorithm does not know the network but observes the number of activations at each round. However, in applications of interest (e.g., HIV prevention) it is not feasible to conduct many low-reward trial campaigns.

Another line of work concerns local graph algorithms, where a local algorithm only uses the neighborhoods around individual nodes. Borgs et al. (2012) study local algorithms for finding the root node in a preferential attachment graph and for constructing a minimum dominating set. Other work aims to find nodes with high PageRank using local queries (Bressan, Peserico, and Pretto 2013; Borgs et al. 2014). These algorithms are not suitable for our problem since a great deal of previous work has observed that seeding high PageRank nodes can prove highly suboptimal for influence maximization (Kimura et al. 2009; Chen, Wang, and Wang 2010; Jung, Heo, and Chen 2012). Essentially, PageRank identifies a set of nodes that are *individually* central, while influence maximization aims to find a set of nodes which are *collectively* best at diffusing information. We also emphasize that our technical approach is entirely distinct from work on PageRank. Lastly, Alon et al. (2015). attempt to infer a ground truth from the opinions of agents with an unknown social network, a different task from ours with correspondingly distinct techniques.

4 The ARISEN algorithm

We now introduce our main contribution, the ARISEN algorithm (*Approximating with Random walks to Influence a Socially Explored Network*). Figure 2 shows an example, explained in detail later. The idea behind ARISEN (Algorithm 1) is to sample a set of T random nodes $\{v_1 ... v_T\}$ from G and explore a small subgraph H_i around each v_i by taking R steps of a random walk (Lines 1-3). R and T are inputs; Section 6 gives settings which obtain theoretical guarantees. Intuitively, T should be greater than K (the number of seeds) so we can be sure of sampling each of the largest K communities. R is discussed with Algorithm 2. The subgraphs H_i are used to construct a weight vector $\boldsymbol{w}$ where w_i gives the weight associated with v_i (Lines 5-6). The algorithm then independently samples each seed from $\{v_1 ... v_T\}$ with probability proportional to $\boldsymbol{w}$ (Line 7).

The challenge is to construct weights $\boldsymbol{w}$ which balance two opposing goals. First, we would oftentimes like to disperse the seed nodes throughout the network. For instance, if each community has equal size, we would like to seed K different communities. Second, we would other times like to place more seeds in large communities. For instance, if one community has 10,000 nodes and each other has only 100 nodes, we should seed the large community more. ARISEN navigates this tradeoff with the following ingredients (Algorithms 1-2). First, INITIALIZEWEIGHTS uses the random walk around each v_i to estimate the size of the community that v_i lies in. From these estimates, it constructs a $\boldsymbol{w}$ that, in expectation, seeds the largest K communities. Second, REFINEWEIGHTS tests if a $\boldsymbol{w'}$ that puts more weight on large communities would increase the expected influence. The main novelty is to carry out these steps using purely local information, since we will not generally be able to tell which of the v_i are in the same community.

Algorithm 3 ARISEN(R, T, B)

1: **for** $i = 1...T$ **do**
2: Sample v_i uniformly random from G.
3: $H_i = R$ nodes on a random walk from v_i.
4: **end for**
5: $\boldsymbol{w}, \boldsymbol{H'}$ = INITIALIZEWEIGHTS$(\boldsymbol{H}, K, R, T, B)$
6: $\boldsymbol{w'}$ = REFINEWEIGHTS$(\boldsymbol{w}, \boldsymbol{H'})$
7: Sample $u_1 ... u_K \overset{iid}{\sim} \boldsymbol{w'}$
8: **return** $u_1 ... u_K$

Algorithm 1 InitializeWeights($\boldsymbol{H}, K, R, T, B$)

1: **for** $i = 1...T$ **do**
2: Form H_i' by discarding the first B nodes of H_i and keeping each remaining node v_j w.p. $\frac{1}{d(v_j)}$
3: $\hat{d} = \frac{1}{R}\sum_{u \in H_i'} d(u)$
4: $\hat{S}_i = \frac{\hat{d} - p_b n}{p_w - p_b}$
5: **end for**
6: $w_j = \frac{n}{\hat{S}_j T},\ j = 1...T$
7: $\tau = \max\{\hat{S}_j | \sum_{\{i|\hat{S}_i \geq \hat{S}_j\}} w_i \geq K\}$
8: For any j with $\hat{S}_j < \tau$, set $w_j = 0$.
9: **return** $\boldsymbol{w}$, $\boldsymbol{H'}$

Algorithm 2 RefineWeights($\boldsymbol{w}, \boldsymbol{H}$)

1: **for** $i = 1...T$ **do**
2: $v_i = \arg\max_{v \in H_i} f(v, H_i)$
3: **end for**
4: $\boldsymbol{w'} = \boldsymbol{w}$
5: **sort** $\boldsymbol{w'}$ in increasing order by $f(v_i, H_i)$
6: **for** $i = 1...T$ **do**
7: **while** EstVal($2w_i', w_{-i}$) > EstVal($\boldsymbol{w'}$) **do**
8: $w_i' = 2w_i'$
9: **end while**
10: $w_i' = \text{BinarySearch}(w_i', 2w_i')$
11: **end for**
12: **return** $\boldsymbol{w'}$

We first formalize the objective that ARISEN optimizes, which is a lower bound on its true influence. Let $f(X, C_i)$ denote the influence of seed set X on the subgraph C_i and $g(X) = \sum_{i=1}^{L} f(X, C_i)$, i.e., the influence spread within each community without considering between-community edges. ARISEN aims to optimize $\mathbb{E}[g(X)]$. Note that $f(X, G) \geq g(X)$ always holds. When p_b is low and little influence spreads between communities (which is the case that we study), g is a good proxy for the true influence. We now explain ARISEN in detail, and how it optimizes the surrogate objective g. Our focus on g is justified in Section 6, where we bound the gap between $\mathbb{E}[g(X)]$ and OPT.

Initial weights

In the SBM, each community C_i has expected average degree $d_i = |C_i|p_w + (n - |C_i|)p_b$. Solving for $|C_i|$, we can estimate the size of the community from its average degree. Since we do not have direct access to d_i, InitializeWeights estimates d_i (and hence $|C_i|$) using the nodes sampled in the random walk (Algorithm 2, Lines 3-4); we discard the first B nodes in this sampling to avoid biasing the estimate. Since a random walk is biased towards high degree nodes, we use rejection sampling (Line 2) to obtain an unbiased estimate. In order to choose seed nodes using the estimated sizes, a natural idea would be to choose the K samples with the largest estimated size. However, this fails because large communities are sampled more often and will be seeded many times, which is redundant. E.g., in the example in Figure 2, placing all of the seeds in C_1 would be suboptimal compared to also seeding C_2. The difficulty is that using local information, we will not know which samples belong to the same community. One solution is to weight each sample *inversely* to its size (Line 6), and then sample seeds with probability proportional to the weights. This evens out the sampling bias towards large communities. Using weighted sampling gives us a principled way to prioritize samples and facilitates later steps which tune the weights to improve performance. In Figure 2, all communities have total weight of 1 after inverse weighting (Frame 2).

Next, the weights are truncated so that only the largest K communities receive nonzero weight (Line 7). After this step, the largest K communities have weight 1 and all smaller communities have weight 0 (at least approximately, due to sampling errors). For example, Frame 3 of Figure 2 shows that only C_1, C_2 and C_3 have nonzero weight. Suppose that we draw K seeds using the resulting weights. In each draw, each of the top K communities is seeded with probability approximately $\frac{1}{K}$. Thus, the cumulative probability that each is seeded is nearly $1 - \left(1 - \frac{1}{K}\right)^K \geq 1 - 1/e$. This reasoning is formalized in our theoretical guarantees.

Refining the weights

The initial weights suffice to obtain the approximation guarantee proved below and are the best possible for some networks. However, they are overly pessimistic in other cases, such as when some communities are much larger than others. In such cases, it would be better to focus more seed nodes on large communities. RefineWeights tunes the weights produced by InitializeWeights to account for such scenarios. In essence, RefineWeights tries to exploit easier cases where some communities are much larger than others by producing new weights $\boldsymbol{w'}$.

RefineWeights (Algorithm 2) starts in Line 2 by defining v_i to be the most influential node in the sampled subgraph H_i (instead of the random starting node). Lines 5-11 successively modify each element of $\boldsymbol{w}$. Starting with the weights corresponding to the highest-value communities, RefineWeights asks whether g would be increased by doubling the w_i under consideration (Line 7). If yes, we set $w_i = 2w_i$ and ask if it can be doubled again. If no, RefineWeights performs a binary search between w_i and $2w_i$ to find the best setting (Line 10). Then, it moves on to the weight corresponding to the next community. In the example in Figure 2, Frame 4 shows that the weights of samples from C_1 and C_2 have been increased. Each change is made only if it improves g, so we have:

Proposition 1. Let $\boldsymbol{w}$ the output of InitializeWeights and $\boldsymbol{w'}$ be the output of RefineWeights. Then, $\mathbb{E}_{X \sim \boldsymbol{w'}}[g(X)] \geq \mathbb{E}_{X \sim \boldsymbol{w}}[g(X)]$.

The key difficulty is determining if each modification increases g. In the EstVal procedure, we provide a way to estimate g using only local knowledge:

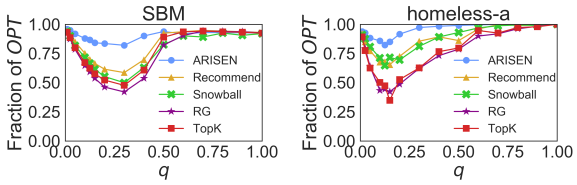

Figure 3: Influence compared to OPT as q varies.

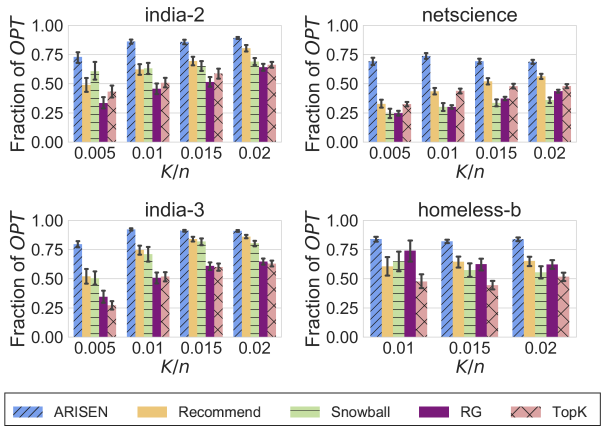

Figure 4: Influence spread compared to OPT as K varies with $q = 0.15$.

Proposition 2. $\textsc{EstVal}(\boldsymbol{w}) = \mathbb{E}_{X\sim\boldsymbol{w}}\left[g(X)\right]$

We give the main idea here; see the supplement for a proof and pseudocode for EstVal. Take any seed set X. Note that the influence within each C_i depends only on nodes in $X \cap C_i$, which we write as X_{C_i}. So, g can be rewritten as $g(X) = \sum_{i=1}^{L} \mathbb{E}[f(X_{C_i}, C_i)]$. If we knew X_{C_i}, then we could calculate $\mathbb{E}\left[f(X_{C_i}, C_i)\right]$ by simulating draws from the SBM for the unobserved portions of C_i. Concretely, let H_i be the subgraph observed in community C_i, with estimated size $\hat{S}_i$. We simulate the rest of C_i by adding $\hat{S}_i - |H_i|$ new nodes, with edges between them and H_i randomly generated from the SBM. This is sufficient to choose the best seed within H_i, as in Line 2. For Line 7, we need to estimate g. The obstacle is not knowing which of the $v_1 ... v_T$ lie in the same community (since a node will contribute less influence if there is another seed from the same community). However, we do know (approximately) how many other times each community is sampled, and the (approximate) weight that those samples will receive, so g can be estimated by averaging some careful simulations. Via a standard Hoeffding bound (Kempe et al. 2015), $O(\frac{n^2}{\epsilon^2}\log\frac{1}{\epsilon})$ simulations per EstVal call guarantee error ϵ with high probability.

5 Experiments

We now present experiments comparing ARISEN with several baselines on an array of networks. We focus on networks with about 100-1000 nodes because this is the size of real-world social groups of interest to us. The first network is *homeless*: Two networks (a and b) gathered from homeless youth in Los Angeles and used to study HIV prevention with 150-200 nodes each. Second, *india*: Three networks of the household-level social contacts of villages in rural India. Gathered by Banerjee et al. (2014) to study diffusion of information about microfinance programs, with 250-350 nodes each. Third, *netscience*[1]: a collaboration network of network science researchers with 1461 nodes. Fourth, *SBM*: a synthetic SBM graphs with 1000 nodes. There are 10 communities with size from 350 to 30 nodes ($p_w = 6 \cdot 10^{-3}$, $p_b = 2{\cdot}10^{-5}$). We approximate the optimal value by running TIM (Tang, Xiao, and Shi 2014), a state of the art influence maximization algorithm, on each full network. For each real network, p_w and p_b are estimated from a different network in the same category (for netscience, we use another collaboration network, astro-ph[1]). For SBM, we use another network from the same distribution. We present a cross-section of results across the datasets but the general trends hold for all networks. Exhaustive results are in the supplement.

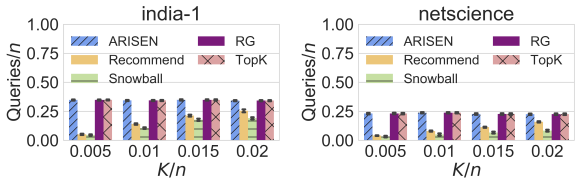

Figure 5: Query complexity as K varies.

Table 1: ARISEN's % influence gain with 25% fewer seeds.

Network/baseline	Rec.	Snowball	RG	TopK
homeless-a	24.2	9.9	20.7	91.1
india-1	0.03	6.6	25.7	29.1
netscience	4.8	63.9	35.4	43.4

We consider four benchmarks. First, *random greedy* (RG). RG uses the same query budget as ARISEN, but queries nodes uniformly randomly. It then runs TIM on the graph composed of the edges these queries reveal. Hence, RG uses a sophisticated seed selection technique, but not ARISEN's sampling procedure. Second, *TopK*. TopK uses ARISEN's random walk sampling (lines 1-3), but seeds the K samples with highest estimated community size instead of using InitializeWeights and RefineWeights. RG and TopK jointly test the importance of ARISEN's sophisticated methods for sampling the network and selecting seed nodes, respectively. Third, *recommend*, which for each of the K nodes, first queries a random node and then seeds their highest degree friend. Fourth, *snowball*, which starts from a random node and seeds that node's highest degree neighbor. It then seeds the highest degree neighbor of the first seed, and so on. Recommend and snowball are the most common strategies in the field (Valente and Pumpuang 2007).

Figure 3 shows that ARISEN obtains substantially higher influence spread than the baselines, often exceeding the best baseline by 20-50%. The x axis varies q. Each point gives

[1]http://www-personal.umich.edu/ mejn/netdata/

the fraction of OPT achieved for that q, averaged over 50 runs. E.g., the point at $q = 0.2$ for SBM indicates that ARISEN's value was $0.8 \cdot OPT$. We take $K = 0.01n$, focusing on when few seeds are available (as in previous work (Chen, Wang, and Wang 2010)). All differences ($q \in [0.01, 0.7]$) are statistically significant (t-test with Bonferroni correction, $p < 10^{-7}$). The gap between ARISEN and the baselines is particularly high in the difficult case of small but nonzero q. When q is close to 0, all algorithms perform close to OPT since little is possible. When q is very high, influence maximization is easy and nearly any algorithm performs well (Chen, Wang, and Wang 2010). Thus, Figure 4 presents results where K is varied with $q = 0.15$ fixed (since this is the hard case). We see that ARISEN uniformly outperforms the baselines, particularly when K is small. As K becomes larger, the baselines improve (again because the problem becomes easier). However, they are still outperformed by ARISEN.

In particular, we conclude from RG's poor performance that ARISEN's random walk based query scheme substantially improves on uniformly sampling an equivalent number of nodes. The comparison with TopK confirms that ARISEN's weighted seed selection is also necessary since simply seeding the largest communities does poorly. In combination, this demonstrates that ARISEN's major elements are both needed to ensure good empirical performance.

Figure 5 examines each algorithm's query cost (each selects the same number of seeds). The supplement lists R and T values; here we just focus on the total queries. ARISEN uses more queries than recommend and snowball, and an equal number to RG and TopK. However, recommend and snowball use more queries as K increases, with query cost close to ARISEN for $K = 0.02 \cdot n$. ARISEN's query cost is uniformly in the range $0.20 \cdot n - 0.35 \cdot n$, a relatively small portion of the network in absolute terms. This query budget is justified by ARISEN's larger influence spread, which makes more efficient use of seed nodes. Intervening to seed a node is often much more costly than querying its edges, as in the HIV domain where an intervention is a day-long class. Table 1 shows the percent by which ARISEN's influence spread exceeds each baseline when the baseline uses $K = 0.02 \cdot n$ but ARISEN uses 25% fewer seeds. ARISEN outperforms all of the baselines, often by over 20%. Hence, ARISEN delivers higher influence with fewer costly seeds.

6 Theoretical analysis

The previous section showed that ARISEN obtains close to optimal influence spread on an array of real world networks. We complement these results with a theoretical analysis which formally demonstrates how ARISEN makes use of community structure in the stochastic block model.

We aim to show that ARISEN's expected influence is close to OPT. We analyze the weights produced by INITIALIZEWEIGHTS; applying REFINEWEIGHTS can only increase $\mathbb{E}[g(X)]$ (Proposition 1). Note that INITIALIZEWEIGHTS, and hence our theoretical guarantees, does not require the algorithm to know q. However, REFINEWEIGHTS uses q, if is available, to improve empirical performance. We often use the following connection between the joint behavior of the SBM/ICM on the one hand, and the connected components of an Erdős-Rényi random graph on the other. The ICM can be seen as removing each edge independently with probability $1 - q$. A node is influenced if afterwards it lies in the same connected component as a seed node (Kempe, Kleinberg, and Tardos 2003). Since each community is itself an Erdős-Rényi graph, the connected components induced by the ICM in each community are distributed exactly as those in an Erdős-Rényi graph with connection probability $p_w q$. A well-known result characterizes the component sizes:

Lemma 1 ((Janson, Luczak, and Rucinski 2011)). *Consider the Erdős-Rényi graph $\mathcal{G}(n, p)$. If $np < 1$, then with probability $1 - o(1)$, its largest connected component has size at most $\frac{3}{(1-np)^2} \log n$. If $np > 1$, then with probability $1-o(1)$, its largest component has size $(1 + o(1))\beta n$. β is the solution to $\beta + e^{-\beta np} - 1 = 0$.*

We denote by $\beta(x)$ the fraction of nodes contained in the largest connected component of $\mathcal{G}(x, p_w q)$ (assuming that $x p_w q > 1$ and the event in Lemma 1 occurs). $\beta(|C_i|)$ gives the fraction of C_i that can be reached by a cascade.

Model parameters: As discussed earlier, we must place some restrictions on p_w and p_b to model real-world networks. While it is often possible to prove approximation guarantees for ARISEN in other settings, we focus on a particular parameter range which produces networks with community structure. First, we assume that each community is internally connected, with few between-community edges.

Assumption 1. (a) For all communities C_i, it holds that $p_w \geq \frac{\log |C_i|}{|C_i|}$. (b) $p_b < \frac{1}{n}$.

Assumption 1(a) is necessary for each community to be internally connected (Janson, Luczak, and Rucinski 2011); the idea of a community is not meaningful if it may contain entirely disconnected subgroups. Assumption 1(b) is also necessary for meaningful community structure. Without it, we can see via Lemma 1 that the graph will contain a giant connected component consisting just of between-community edges. We make a complementary assumption that influence mostly spreads within communities:

Assumption 2. (a) $p_w q |C_i| > 1$ (b) Let $c_{max} = \max_i p_b q \cdot (n - |C_i|)|C_i|$. We require $c_{max} < 1$.

Assumption 2(a) implies that it is possible for an influence cascade to reach a linear portion of the community. Otherwise, if $p_w q |C_i| < 1$, at most $O(\log |C_i|)$ nodes can be influenced by any constant number of seeds (via Lemma 1). We focus on when it is possible for influence maximization to have large results, not when only a vanishingly small fraction of nodes can possibly be reached. In Assumption 2(b) c_{max} is the average number of additional communities influenced by a single seed node. $c_{max} < 1$ says that while between-community spread is possible, the *average* seed node influences just its own community. Otherwise, a cascade starting in one community can reach a linear portion of the graph. While it is clearly possible to give guarantees for this case (even for choosing seeds completely at random), small between-community influence is both more challenging and more relevant to applications.

Isolated communities: $p_b = 0$

For simplicity, we start with disconnected communities (fixing $p_b = 0$). Here, $g(X)$ is exactly the influence of a seed set, so we just have to show that ARISEN obtains a high value of g. We begin with a simplified version of our main result which captures the intuition behind the proof. Suppose that the top K communities each have equal size μ, and occupy a linear portion of the network – for concreteness, $\mu K \geq 0.01n$. We have

Theorem 2 (Simplified case)**.** *Under the above conditions, ARISEN can be implemented with approximation ratio* $(1 - \frac{1}{e^{0.99}} - 0.01 - \frac{1}{K} - o(1))\beta(\mu)$ *using* $O(\log^6 n)$ *queries.*

The query cost is chosen so that the random walk based estimates of each community's size are accurate with high probability. We emphasize that only a polylogarithmic number of nodes need be queried, an exponential improvement over exhaustive surveys. We now motivate the two components of the approximation ratio. The first term is nearly $1-1/e$, up to error terms which decrease as n and K become large. We show that each of the top K communities is seeded with probability close to $1 - 1/e$. The proof tracks the intuition outlined when INITIALIZEDWEIGHTS was described: each community receives total weight close to 1, giving it probability close to $\frac{1}{K}$ of being hit by each of K seeds. The second term, $\beta(\mu)$, is the fraction of each of the top K communities which can be influenced by a seed node (via Assumption 2(a)). These nodes form a giant connected component under the ICM. Consider a given seed node u_i, which is a uniformly random node in some C_i. With probability $\beta(\mu)$, u_i lies in this component; hence it influences at least $\beta(\mu)^2\mu$ nodes in expectation. The best that OPT can do is to influence the entire connected component with certainty, giving an influence spread of $\beta(\mu)\mu$. The ratio between these terms is $\frac{\beta^2(\mu)\mu}{\beta(\mu)\mu} = \beta(\mu)$. Essentially, $\beta(\mu)$ expresses the difficulty of finding the influential nodes in each community and increases as the product $\mu p_w q$ becomes larger.

We now state our full result, which applies when the top K communities have unequal (possibly sublinear) sizes. We compensate by setting T and R, the number of samples, to account for this imbalance. We do so by introducing parameters ρ and ϵ. ρ reflects how large the top K communities are compared to n, while ϵ reflects the desired accuracy. We then set $T = O\left(\frac{1}{\epsilon^3\rho}\log\frac{1}{\epsilon\rho}\right)$ and $R = O\left(\frac{1}{\epsilon^2}\log^2\left(\frac{T}{\epsilon}\right)\log^6 n\right)$, chosen to ensure that each community's size is estimate accurately and hence achieve the desired approximation guarantee. In Theorem 2 we fixed $\epsilon, \rho = 0.01$, leading to $T = \Theta(1), R = O\left(\log^6 n\right)$ and $TR = O\left(\log^6 n\right)$ total queries.

Our result requires two technical conditions on ρ and ϵ. Let μ now be the *average* size of the top K communities, $\mu = \frac{1}{K}\sum_{i=1}^{K}|C_i|$. The first condition is $\rho \leq \mu/n$. For instance, if $K = 3$ and the largest three communities occupy $\frac{3}{4}$ of the graph in total, we need $\rho \leq \frac{1}{4}$. This ensures that there are enough samples to detect the largest K communities since we set T higher when ρ is small. ρ should be large if we prefer to use few samples (and risk failing if the top K communities are very small) and small to guarantee performance even when all communities are small.

The second technical condition is that $\epsilon^5\rho|C_i| = poly(n)$ for all C_i, which says that the attainable degree of precision becomes smaller when there are very small communities which are hard to distinguish.

Let $\beta_{min} = \beta((1-\epsilon)|C_K|)$ and $\beta_{max} = \beta(|C_1|)$ where C_1 is the largest community and C_K the kth largest. β_{min} (β_{max}) measures the fraction of the smallest (largest) of the top K communities that can be influenced. We have:

Theorem 3. *Suppose that* $\rho \leq \frac{\mu}{n}$ *and choose* $\epsilon < \frac{3}{8}$ *such that* $\epsilon^5\rho|C_i| = poly(n)\ \forall C_i$. *Then ARISEN can be implemented using* $O\left(\left(\frac{1}{\epsilon^5\rho}\right)\log^3\left(\frac{1}{\epsilon\rho}\right)\log^6 n\right)$ *queries with approximation ratio* $(1 - e^{-(1-\epsilon)} - \epsilon - \frac{1}{K} - o(1))\frac{\beta_{min}^2}{\beta_{max}}$.

One difference from Theorem 2 is the term $\frac{\beta_{min}^2}{\beta_{max}}$, caused by imbalanced community sizes. When the top K communities have similar sizes, $\frac{\beta_{min}^2}{\beta_{max}}$ converges to $\beta(\mu)$. Besides handling unequal community sizes, Theorem 3 lets us use more queries to extend the approximation guarantee to when all communities have sublinear size. E.g., when $\mu K = \Theta(\sqrt{n})$, we need $O(\sqrt{n}\log^9 n)$ queries.

General case: $p_b > 0$

We now generalize to handle edges between communities. There are two new challenges. First, a random walk may leave its starting community, which could invalidate our estimate of that community's size. However, since p_b is small by Assumption 1(b), there are few between-community edges. Further, the random walks are short since R (the number of steps) is polylogarithmic in n. Hence, we show that all walks stay in their starting communities with high probability.

Second, we must account for between-community influence spread. ARISEN does not try to select nodes that bridge multiple communities, so it may get unlucky and not benefit at all from between-community influence. However, it always does at least as well as when $p_b = 0$ since any additional influence can only help. On the other hand, OPT may be able to exploit $p_b > 0$ by using its full knowledge of the network to find seeds that can each influence multiple communities. We bound the extent to which this is possible. Specifically, Assumption 2(b) ($c_{max} < 1$) says that between-community influence is small. Under this condition, we show that OPT can increase from the $p_b = 0$ case by at most a factor of $\frac{12\log\frac{n}{\mu}}{1-c_{max}}$. When the top K communities comprise a linear portion of the network ($\mu = \Theta(n)$), this is constant with respect to n. Formally, we obtain:

Theorem 4. *Under the same conditions as Theorem 3, but with* $p_b > 0$*, ARISEN achieves an approximation ratio of*

$$\left(\frac{1 - c_{max}}{12\log\frac{n}{\mu}}\right)\frac{\beta_{min}^2}{\beta_{max}}\left(1 - e^{-(1-\epsilon)} - \epsilon - \frac{1}{K} - o(1)\right)$$

We note that the constant $\frac{1}{12}$ can likely be improved; the main take-away is the dependence on the network structure. We conclude that ARISEN provably exploits community structure in the SBM, providing context for its strong empirical performance on real world networks.

Acknowledgments: This research was supported by MURI Grant W911NF-11-1-0332. Wilder was supported by a NSF Graduate Fellowship.

References

Abbe, E., and Sandon, C. 2015. Community detection in general stochastic block models: Fundamental limits and efficient algorithms for recovery. In *FOCS*, 670–688. IEEE.

Alon, N.; Feldman, M.; Lev, O.; and Tennenholtz, M. 2015. How robust is the wisdom of the crowds? In *IJCAI*, 2055–2061.

Banerjee, A.; Chandrasekhar, A. G.; Duflo, E.; and Jackson, M. O. 2014. Gossip: Identifying central individuals in a social network. Technical report, National Bureau of Economic Research.

Bond, R. M.; Fariss, C. J.; Jones, J. J.; Kramer, A. D.; Marlow, C.; Settle, J. E.; and Fowler, J. H. 2012. A 61-million-person experiment in social influence and political mobilization. *Nature* 489(7415):295–298.

Borgs, C.; Brautbar, M.; Chayes, J.; Khanna, S.; and Lucier, B. 2012. The power of local information in social networks. In *WINE*, 406–419. Springer.

Borgs, C.; Brautbar, M.; Chayes, J.; and Teng, S.-H. 2014. Multiscale matrix sampling and sublinear-time pagerank computation. *Internet Mathematics* 10(1-2):20–48.

Brautbar, M., and Kearns, M. J. 2010. Local algorithms for finding interesting individuals in large networks. In *Innovations in Theoretical Computer Science*, 188–199.

Bressan, M.; Peserico, E.; and Pretto, L. 2013. The power of local information in pagerank. In *WWW*, 179–180. ACM.

Carpentier, A., and Valko, M. 2016. Revealing graph bandits for maximizing local influence. In *International Conference on Artificial Intelligence and Statistics*, 10–18.

Chen, Y.-C.; Zhu, W.-Y.; Peng, W.-C.; Lee, W.-C.; and Lee, S.-Y. 2014. Cim: community-based influence maximization in social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5(2):25.

Chen, W.; Wang, C.; and Wang, Y. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, 1029–1038. ACM.

Fienberg, S. E., and Wasserman, S. S. 1981. Categorical data analysis of single sociometric relations. *Sociological methodology* 12:156–192.

Janson, S.; Luczak, T.; and Rucinski, A. 2011. *Random graphs*, volume 45. John Wiley & Sons.

Jung, K.; Heo, W.; and Chen, W. 2012. Irie: Scalable and robust influence maximization in social networks. In *ICDM*, 918–923. IEEE.

Kempe, D.; Kleinberg, J.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *KDD*, 137–146. ACM.

Kimura, M.; Saito, K.; Nakano, R.; and Motoda, H. 2009. Finding influential nodes in a social network from information diffusion data. In *Social Computing and Behavioral Modeling*. Springer. 1–8.

Krzakala, F.; Moore, C.; Mossel, E.; Neeman, J.; Sly, A.; Zdeborová, L.; and Zhang, P. 2013. Spectral redemption in clustering sparse networks. *PNAS* 110(52):20935–20940.

Leskovec, J.; Lang, K. J.; Dasgupta, A.; and Mahoney, M. W. 2009. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics* 6(1):29–123.

Mihara, S.; Tsugawa, S.; and Ohsaki, H. 2015. Influence maximization problem for unknown social networks. In *ASONAM*, 1539–1546. ACM.

Mossel, E.; Neeman, J.; and Sly, A. 2015. Reconstruction and estimation in the planted partition model. *Probability Theory and Related Fields* 162(3-4):431–461.

Rice, E.; Tulbert, E.; Cederbaum, J.; Adhikari, A. B.; and Milburn, N. G. 2012. Mobilizing homeless youth for HIV prevention. *Health education research* 27(2):226–236.

Squartini, T.; Picciolo, F.; Ruzzenenti, F.; and Garlaschelli, D. 2012. Reciprocity of weighted networks. *Scientific Reports*.

Tang, Y.; Xiao, X.; and Shi, Y. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *KDD*. ACM.

Valente, T. W., and Pumpuang, P. 2007. Identifying opinion leaders to promote behavior change. *Health Education & Behavior*.

Wang, Y.; Cong, G.; Song, G.; and Xie, K. 2010. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *KDD*, 1039–1048. ACM.

Wilder, B.; Yadav, A.; Immorlica, N.; Rice, E.; and Tambe, M. 2017. Uncharted but not uninfluenced: Influence maximization with an uncertain network. In *AAMAS*, 740–748.

Yadav, A.; Chan, H.; Xin Jiang, A.; Xu, H.; Rice, E.; and Tambe, M. 2016. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *AAMAS*, 740–748.