Spread Sampling for Graphs: Theory and Applications

Yu Wang* Bortik Bandyopadhyay* Aniket Chakrabarti*
David Sivakoff# Srinivasan Parthasarathy*
* Department of Computer Science and Engineering, # Department of Statistics
The Ohio State University, Columbus, Ohio, USA
wang.5205@osu.edu,srini@cse.ohio-state.edu

ABSTRACT

This paper proposes a novel scalable node sampling algorithm for large graphs that can achieve better *spread* or diversity across communities intrinsic to the graph without requiring any costly pre-processing steps. The proposed method leverages a simple iterative sampling technique controlled by two parameters: *infection rate*, that controls the dynamics of the procedure and *removal threshold* that affects the end-of-procedure sampling size. We present theoretical analyses of the sampling probability for this method on the celebrated Erdős–Rényi graph model, and of the community diversity on the Stochastic Block Model. Efficiently finding small samples with high diversity from large graphs has a number of practical applications such as online survey and community detection. Our method achieves very high community diversity with extremely low sampling budget on both synthetic and real-world graphs, with either balanced or imbalanced communities. We leverage the proposed sampling technique on community detection and show it outperforms the baselines of the same type.

CCS CONCEPTS

• **Mathematics of computing** → **Graph theory**; *Graph algorithms*; • **Theory of computation** → **Sketching and sampling**;

ACM Reference Format:
Yu Wang* Bortik Bandyopadhyay* Aniket Chakrabarti* David Sivakoff#
Srinivasan Parthasarathy*. 1997. Spread Sampling for Graphs: Theory and Applications. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97).* ACM, New York, NY, USA, Article 4, 8 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Networks are an expressive tool to represent relational data in various domains: from sociology to biology, from political science to online marketing. Given the ubiquitousness of the Internet, we are able to collect relational data at an extremely large scale. A huge amount of data restrains us from conducting complicated analysis and sampling is often touted as a means to combat the inherent complexity of analyzing large networks [16].

Network sampling is broadly classified into node and edge sampling strategies. In this paper, we study node sampling, which seeks to sample nodes from a network to perform inference. The idea of sampling subjects from different groups is called *stratified sampling* [18]. Suppose we want to take a survey covering as many professions as possible on an online social network. A typical stratification-driven design is to first divide the user population into different strata (groups) using occupational information and then sample individuals from each group. However, due to individual privacy preferences, some users might choose not to share their professional details, whereby direct stratification approach based on occupation data is infeasible.

In a network context, people with the same profession interact more frequently, leading to assortative communities [5]. To solve this profession sampling problem, one could first detect communities using network topology, and then sample from each community. However, community detection is a very time-consuming procedure even with state-of-the-art implementation [2, 21]. Chain-based sampling [7] methods sample connected subgraphs and hence are more likely to be stuck in one community, resulting in less community diversity in the sample. Uniform node selection sampling [16] has better community coverage than chain-based sampling, but it tends to under-sample small communities when community sizes are imbalanced. Also from an end-to-end application's performance perspective, to continue benefiting from sampling approaches, the sampling budget (i.e, number of nodes to be sampled) has to be kept low, which reduces the chance of high community coverage under practical settings on large graphs.

We propose a new graph sampling method that can achieve better community diversity than existing algorithms for low sampling budget even in graphs with imbalanced communities, resulting in a more representative node set in the sample in terms of community diversity than other methods. Under appropriate user-chosen parameter configuration, the proposed method penalizes sampling neighboring nodes, cliques or near-clique structures, and hence allows for better overall community diversity of the network without any costly pre-processing step required for a typical stratified sampling approach. We provide a theoretical analysis of the sampling probability of this method, compare the community diversity of this method against baselines, and show its efficacy in practical applications like community detection seeding and maximum independent set discovery.

2 RELATED WORK

The work by [11] systematically studies node sampling on social networks. It proposes the concept of model-based sampling and design-based sampling. Our proposed method is a variant of design-based sampling technique. Node selection sampling (a kind of uniform node sampling) is studied by [16]. This method assumes we

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06…$15.00
https://doi.org/10.475/123_4

know the IDs of all the nodes in the network. This is straightforward for small networks; for large real-world social networks, like Facebook and LinkedIn, one could get such knowledge under the following circumstances: 1) to obtain the administrator privilege of the network [8]; 2) to crawl a network in advance and then analyze the crawled network as done by [17].

Traditional chain-based sampling methods are biased towards high degree hub nodes. Hence in-sample correction [7] and post-sample correction [10] are proposed to achieve a near-uniform sampling from chain-based methods. A crawling-based approach by [19] samples a connected subgraph with good community structure. It performs pretty well, in terms of community diversity, on graphs with size-balanced communities. However, its performance deteriorates on graphs with imbalanced communities. And its running time is too long on large networks. Our proposed sampling strategy performs well on both types of the graphs, and runs in linear time.

We choose community detection to test the efficacy and efficiency of our proposed sampling method. Modularity maximization [2, 6] and n-cut maximization [14] based approaches were surveyed by Fortunato [5]. The personalized PageRank based approach [1] is a variant of the n-cut maximization method. The work by [15] compared several seed expansion based community detection methods and concluded Personalized PageRank [1] works best. They find that seeding by uniform sampling results in a better recall than high-degree sampling in community detection. We show that seeding by our proposed spread sampling achieves an even better result than uniform sampling.

3 METHOLOGY

3.1 Designing Spread Sampling

We wish to obtain a sample that spreads out, in terms of community diversity, over the graph with a limited budget. Intuitively, a good spread-out sample set contains nodes that belong to different communities and hence are less likely to be neighbors of each other in the original graph. With this insight, we design an iterative sampling approach alternating between two steps: 1) uniformly sample from candidate nodes; 2) remove neighbors of already sampled nodes. The first step leads to a near-uniform sample while the second step spreads out the sampled nodes. During the sampling process in Algorithm 1, three sets are maintained: the sample set **(S)** contains all the sampled nodes; the removal set **(R)** contains nodes that have certain sampled neighbors; the candidate set **(C)** contains the remaining nodes which are not in **S** and **R**. The cardinality of both **S** and **R** monotonically increases as the iteration proceeds, while it decreases for **C**. To obtain a better "spread out" (to be quantified in Sect. 3.4) sample with a limited budget, the removal set **R** keeps track of the set of nodes for which enough neighbors (typically a user-controlled parameter) are already in the sample to ensure that such nodes are not included in the final sample.

The sampling algorithm has two parameters viz: **single step infection rate (q)** and **removal threshold (k)**, which effectively controls between the two steps to generate sample sets with desired spread property based on the desired application. A (near) uniform sample is generated by either setting a very high removal threshold or a high single-step infection rate. On the contrary, a better spread-out sample with less adjacent nodes (hence reduced local cliques) is generated by setting a small removal threshold and a small single-step infection rate.

Algorithm 1 Spread Sampling (SS method)

Input: infection rate p, removal threshold k, a connected undirected graph G, target sample size;
Output: A set S of sampled nodes.
1: Initialize candidate set $C = G$;
2: **while** C is not empty and $|S|$ smaller than target size **do**
3: for each node u in C, sample it with probability p, and add the sampled nodes into S;
4: $C = C - S$; { remove sampled nodes from candidate set}
{Below: if a candidate node has at least k neighbors sampled, remove it from candidate set}
5: $B_k = \{v \in C \mid |N(v) \cap S| \geq k\}$;
6: $R = R \cup B_k$; { R is the removal set}
7: $C = C - R$;
8: **end while**
9: **return**

3.2 Degree-Inverse Sampling Probability

3.2.1 Theoretical Analysis. We are interested in the following question: *for a node with degree d, what is the probability that this node is sampled*? Our strategy is to define a recursion for the sampling probability at each step, the solution of which leads to the overall sampling probability of each node. Erdős-Rényi graph is a well-studied random graph model. It is simple for theoretical analysis, and well-known community models like SBM [13] and BTER [23] are both its variants. For a sparse ER graph $ER(n, p)$, to make analysis tractable we assume the following: *a) exactly one node is sampled at each step/iteration, b) a node is removed if at least one of its neighbors are sampled, c) there is no sample budget constrain.* Since there is no sample budget constraint, the while-loop in Algorithm 1 exits only when there is no more candidate node.

We introduce the notations in Table 1 and proceed to present the sampling probability analysis framework in Algorithm 2. To compute the sampling probability of a particular node i with degree d, we only need to focus on two mutually exclusive events (Figure 1): a) the node is sampled; b) one of the node's neighbors is sampled causing the node to be removed. Iterations after these two events have no impact on the sampling probability, and therefore can be ignored (if-condition in line 3 of Algorithm 2). We calculate the marginal probability of the node being sampled *at each step*, and sum them up to get the (overall) probability of the node being sampled by the algorithm. The rationale of line 6 is: in t-th iteration, node i has m_{t-1} active neighbors and for a fixed node ξ_t ($\xi_t \neq i$), it has probability p to connect each of i's neighbors on an ER graph. Similar argument holds for line 7. Assuming the sampling probability of node i at t-th iteration is p_t, the desired (overall) sampling probability of node i is $P_i \equiv$P(node i is sampled) $= \sum_{t=1}^{\infty} p_t$.

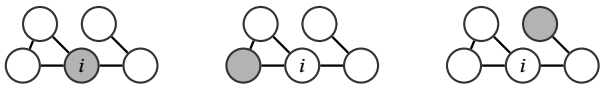
Table 1: Notation Table for Algorithm 2

Notation	Explanation
t	iteration counter
n	network size
p	connection rate of an Erdos-Renyi ER(n, p) graph
i	ID of the node of interest
d_i	degree of node i
M_t	active neighbors of node i at the end of iteration t, and the beginning of iteration $t+1$
m_t	$\|M_t\|$
U_t	candidate nodes at the end of iteration t
u_t	$\|U_t\|$
ξ_t	node sampled at iteration t
n_m	number of common neighbors of ξ_t and i
n_n	number of neighbors of ξ_t disconnected from i
$\sim$	distributed as
Bino	Binomial distribution
$\bar{d}$	d-bar (d with an overline), mean degree of the graph
n_d	number of degree d nodes
P_d	sampling probability of degree d nodes
S	the set of sampled nodes

Algorithm 2 Reduced version of Algorithm 1 for analysis

Input: the same as Algorithm 1
Output: the same as Algorithm 1

1: **while** node i is not sampled and not removed **do**
2: pick up ξ_t u.a.r. from U_{t-1};
3: **if** $\xi_t \in i \cup M_{t-1}$ **then**
4: stop;
5: **end if**
6: $n_m \sim$ Bino(m_{t-1}, p); // number of common neighbors of ξ_t and i.
7: $n_n \sim$ Bino($u_{t-1} - m_{t-1}$ − node $i - \xi_t$, p); // number of neighbors of ξ_t disconnected from i due to ξ_t's being sampled.
8: $U_t = U_{t-1} - n_m - n_n$; // update candidate nodes.
9: $M_t = M_{t-1} - n_m$; // update active neighbors.
10: **end while**
11: **return**



(a) Node i being sampled, and the procedure terminates. **(b) One of node i's neighbors being sampled, and node i is removed.** **(c) Other nodes being sampled.**

Figure 1: Possible outcomes in line 2 per iteration in Algorithm 2, can be classified into three events. At each iteration t, the probability of events {(a) or (b)} is $\pi'_t = \frac{m_{t-1}+1}{u_{t-1}}$. Hence the probability that the procedure stops at iteration t is $\pi_t = \pi'_t P$(The procedure does not stop before iteration t). See Proposition 3.1 for details.

Proposition 3.1. *We show that for a node i with degree d in a sparse and large ER graph, its sampling probability P_d by Algorithm 2 is proportional to $1/(d+1)$.*

$$P_d \propto \frac{1}{d+1}$$

Notations are explained in Table 1, and $d_i = d$.

Proof sketch. $\pi_t \equiv$P(procedure stops at iteration t)
$p_t \equiv$P(node i is sampled at iteration t) $= \frac{\pi_t}{m_{t-1}+1}$ (by definition, Figure 1)
$\pi_t = \frac{m_{t-1}+1}{u_{t-1}}[1 - \sum_{j=1}^{t-1}\pi_j] \approx \frac{m_{t-1}}{u_{t-1}}[1 - \sum_{j=1}^{t-1}\pi_j] \approx \frac{d}{n}(1-\frac{d}{n})^{t-1}$ (using Edgeworth Expansion [9] to approximate a ratio distribution by its expectation)
$\sum_{t=1}^{\infty}\pi_t = 1$ (the sampling procedure cannot run forever)
$p_t = \frac{\pi_t}{m_{t-1}+1} \approx \frac{\pi_t}{d+1}$ (large sparse ER graph[1])
$P_d \equiv P$(node i of degree d is sampled)$= \sum_{t=1}^{\infty}$P(node i is sampled at iteration t)$\equiv \sum_{t=1}^{\infty} p_t = \frac{1}{d+1}\sum_{t=1}^{\infty}\pi_t = \frac{1}{d+1}\cdot 1$
Hence, $P_d \propto \frac{1}{d+1}$ □

The degree-inverse sampling probability agrees with our intuition: after each uniform sampling step of the procedure, a high degree node has more chance to be a neighbor of a sampled node and hence has more chance to be removed. In long run, high degree nodes are all removed and the candidate set has only low degree nodes, resulting in the degree-inverse sampling rate.

Thus, for a **sparse ER** graph, we have $P_i \propto \frac{1}{d_i+1}$. Given $P_i = \frac{c}{d_i+1}$ where c is an unknown graph-dependent parameter, the end-of-procedure sampling size $|S|$ could be obtained by summation: $|S| = \sum_d n_d P_d$. We can show that the end-of-procedure sample size for a large and sparse ER graph is $\lim_{n\to\infty}|S| = \frac{c}{p}[1 - e^{-\bar{d}}]$ i.e, inversely proportional to the mean degree (see Corollary 3.2). It can be shown that when $k = 1$ and q is exactly one node as in Algorithm 2, the obtained sample nodes make up a maximal independent set, and hence the sample size is upper-bounded by the size of the maximum independent set [4]. Moreover, we empirically find that the sampling probability by spread sampling follows a (truncated) power law on dense ER graphs and graphs with community structure like SBM, and BTER, which will be elaborated in Section 3.2.2.

Corollary 3.2. *For a large, sparse Erdos-Renyi graph ER(n, p),*

$$E[|S|] = \sum_{d=0}^{n-1} E[n_d]P_d = \frac{c}{p}[1-(1-p)^n],$$

where $P_d = \frac{c}{d+1}$ is the sampling probability of a degree-d node, and c is a graph-dependent parameter.
Moreover, $\lim_{n\to\infty} E[|S|] = \frac{c}{p}[1 - e^{-\bar{d}}]$.

Proof. $P_d \propto \frac{1}{d+1}$ by Proposition 3.1. For a particular graph, we can represent it as $P_d = \frac{c}{d+1}$, where c is a graph-dependent parameter.

Hence, among all n_d degree d nodes, $n_d P_d$ nodes are included in the sample. And therefore, the total expected sample size is $E[|S|] = \sum_{d=0}^{n-1} E[n_d P_d] = \sum_{d=0}^{n-1} E[n_d]P_d$ $(0 \le d < n)$.
For ER(n, p), $\mathrm{E}[n_d] = n \cdot \binom{n-1}{d} p^d (1-p)^{n-1-d}$
$E[|S|] = \sum_{d=0}^{n-1}\mathrm{E}[n_d]P_d = \sum_{d=0}^{n-1}\frac{c}{d+1}\cdot n\cdot\binom{n-1}{d}p^d(1-p)^{n-1-d}$
$\overset{k\equiv d+1}{=\!=\!=} \frac{c}{p}\sum_{k=1}^{n}\cdot\frac{n!}{k!(n-k)!}p^k(1-p)^{n-k} = \frac{c}{p}[1-(1-p)^n]$

[1]For a large sparse ER graph, the probability that node i's 2-hop neighbors are sampled is low, and hence m_t decays slowly. Our experiments suggest the average decay rate m_t/m_{t-1} of a sparse ER graph with 1k nodes is above 0.99 for the first 100 iterations, and the larger the graph is, the higher this number is.

Hence $E[|S|] = \frac{c}{p}[1-(1-p)^n]$

$\lim_{n\to\infty} E[|S|] \overset{\text{sparse ER, } p=\bar{d}/n}{=\!=\!=} \lim_{n\to\infty} \frac{c}{p}[1-(1-\frac{\bar{d}}{n})^n] = \frac{c}{p}[1-e^{-\bar{d}}]$ □

3.2.2 Empirical Sampling Probability. The sampling procedure Algorithm 1 stops under two conditions: 1) target sample size is reached; 2) no more nodes can be sampled. Hence there are two corresponding sampling probabilities: truncated sampling probability and end-of-procedure sampling probability. We report both the probabilities in Figures 2 and 3. Recall that we are interested in the sampling probability - for a node with degree d, what is the probability that it is sampled, which can be estimated by:

$$\hat{P}(\text{a degree } d \text{ node sampled}) = \frac{\text{number of degree } d \text{ nodes in the sample}}{\text{number of degree } d \text{ nodes in the graph}}$$

We list the size, the number of communities (when applicable) and the clustering coefficient of the graphs we tested in Table 2.

Table 2: Graphs Used in Experiments

Graph	$\|V\|$	$\|E\|$	$\|C\|$	CC
ER($1k$, 0.4)	1k	100k	N/A	0.3992
ER($10k$, 0.001)	10k	50k	N/A	0.0011
PL($1k$, 4, 0.01)	1k	4k	N/A	0.0343
PL($1k$, 4, 0.8)	1k	4k	N/A	0.3720
SBM10k balanced	10k	594k	100	0.0165
SBM10k imbalanced	10k	103k	500	0.1265
BTER syn (fit power-law)	1k	33k	10	0.2696
BTER fb (fit FB below)	4k	86k	250	0.5964
Email [17]	1k	25k	42	0.3994
FB [17]	4k	88k	N/A	0.6055
DBLP [17]	317k	1M	13k	0.6324

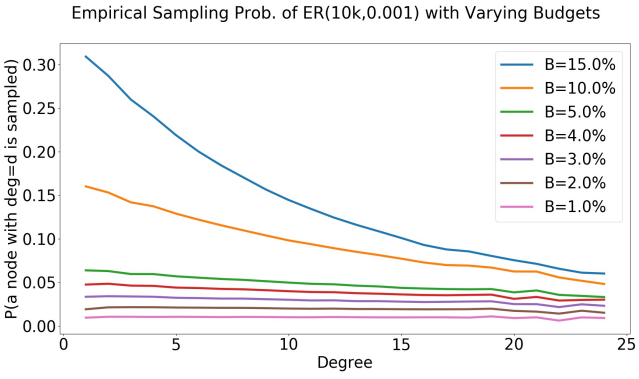
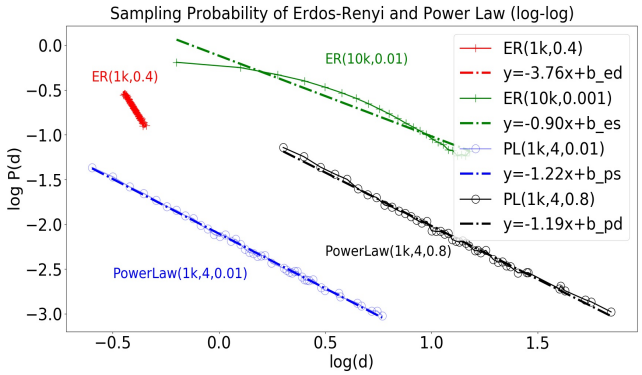
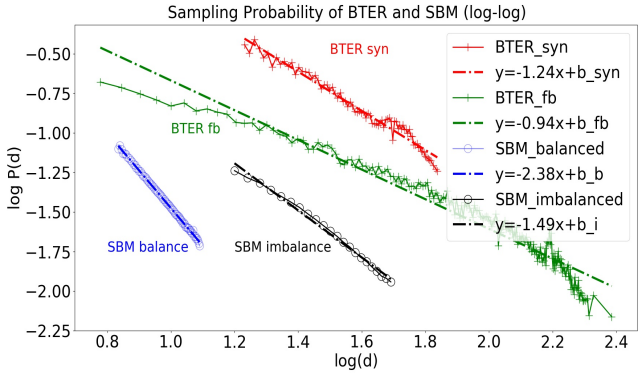


Figure 2: Empirical sampling probability with varying budgets on ER(10k,0.001). Removal threshold equals one, and exactly one node is sampled per step. At small budgets (< 5%), the nodes are nearly uniformly sampled; as the sample budget increases, low degree nodes gain much more chance than others to be sampled.

We report results on graphs without community structure: Erdős-Rényi graphs (dense and sparse) and Power Law graphs (dense and sparse); graphs with community structure: SBM graphs (balanced and imbalanced communities) and BTER graphs (fit a power-law



(a) Sampling probability by SS on sparse/dense ER graph and sparse/dense power-law graph follows power-law. Plus sign represents empirical probability on ER graphs, while circle represents empirical probability on power-law graphs. The dashed straight lines are least-square fitted reference power-law distribution. (straight line on log-log plot is power-law.)



(b) Sampling probability by SS on SBM with (im)balanced communities and BTER fitted from synthetic/real-world graphs follows power-law.

Figure 3: Sampling probability by SS on various graphs follows power-law.

sequence, and FB graph). For small budget (< 5% in Figure 2), the sampling probability for different degrees do not differ too much and hence achieves near-uniform sampling. As the sample budget increases, low degree nodes get much more chance than hub nodes to be sampled. This agrees with our intuition: spread sampling alternates between uniform sampling and node removal. Small sampling budget results in early termination and hence fewer nodes being removed. Therefore, nodes with different degrees have the similar probability of being sampled. As the budget increases, more (high-degree) nodes are removed, as explained after the proof of Proposition 3.1. This phenomenon exists on all types of graphs, and we only report a sparse ER graph.

If we have no sampling budget (or the budget is very high and the procedure cannot reach), the sampling procedure stops only when there is no more nodes to be sampled, and we call it the end-of-procedure status. The sampling probability follows a power

law on all the aforementioned graphs (Figure 3). Knowing the sampling probability helps the design of the Horvitz-Thompson estimator [12].

3.3 High Community Diversity

3.3.1 Theoretical Analysis. We use *community coverage ratio* to quantify how well spread-out the sampled nodes are. We define community coverage ratio[2] for each sample set S as the fraction of communities represented by the sampled nodes in S. It can be formulated as:

$$CoverageRatio(S) = \frac{|\bigcup_{i\in S} c_i|}{|C|},$$

where S is the sample, c_i is the community of node i and C is the set of all the communities. We use its expectation over all possible sample sets to evaluate a sampling method on a graph[3]:

$$\mathrm{E}_S[CoverageRatio] = \mathrm{E}_S[\frac{|\bigcup_{i\in S} c_i|}{|C|}] = \sum_{t\in[0,1]} \mathrm{P}(CoverageRatio = t)\cdot t,$$

A special case in the last formula is when t equals to 1, which means all the communities are represented by the sample. We call this probability the *Full Coverage Probability*. We first show theoretically that SS on SBM graphs has higher full coverage probability than uniform sampling, then empirically compare SS against several baselines in terms of both expected coverage ratio and full coverage probability.

Consider an SBM graph with C equal-sized communities, and each community has n_C nodes. Hence the graph size is $C \cdot n_C$. The within community connection rate is larger than across community connection rate: $p_{\text{in}} > p_{\text{out}}$. For a fixed sampling budget C, the number of communities, we are interested in the probability that all sampled nodes are from distinct communities by sampling method **M**, which is the "full (community) coverage probability", denoted P_{FCP}. We prove that SS has larger P_{FCP} than uniform sampling.

For uniform sampling, the community distribution of the sampled nodes follows a C-variate multi-nomial distribution, hence $P_{\text{FCP,uniform}} = \binom{C}{1,\cdots,1}(1/C)^C = \frac{C!}{C^C}$.

For SS, we name the communities as $C_1, C_2, \cdots, C_C$. For simplicity, we again assume the infection rate is exactly one node and the removal threshold is one node as in Algorithm 2. We consider such an event: the first sampled node comes from C_1, the second sampled node comes from C_2, ..., and the last sampled node comes from C_C. And the probability of this event is P_{atom}. Then $P_{\text{FCP,SS}} = C!P_{\text{atom}}$. Now we show $P_{\text{atom}} > 1/C^C$, and hence $P_{\text{FCP,SS}} > P_{\text{FCP,uniform}}$.

The **intuition** is that for each to-be-covered community, which means we haven't sampled any node from that community, its size is larger than the global average, and hence got a higher chance to be sampled in the uniform sampling stage of SS.

Proposition 3.3. *$P_{atom} > 1/C^C$, and hence $P_{FCP,SS} > P_{FCP,uniform}$.*

Proof. Let $n_{C_i}^{(j)}$ be the size of community C_i after j nodes sampled, in particular $n_{C_i}^{(0)} = n_C, \forall i \in \{1, 2, \cdots, C\}$. Hence $P_{\text{atom}} =$

$$\frac{n_{C_1}^{(0)}}{\sum_{i=1}^C n_{C_i}^{(0)}} \times \frac{n_{C_2}^{(1)}}{\sum_{i=1}^C n_{C_i}^{(1)}} \times \cdots \times \frac{n_{C_C}^{(C-1)}}{\sum_{i=1}^C n_{C_i}^{(C-1)}} = \prod_{j=0}^{C-1} \frac{n_{C_{j+1}}^{(j)}}{\sum_{i=1}^C n_{C_i}^{(j)}}.$$

$$n_{C_i}^{(j)} = \begin{cases} n_C(1-p_{\text{out}})^j, & j < i \\ [n_C(1-p_{\text{out}})^{i-1} - 1](1-p_{\text{in}})(1-p_{\text{out}})^{j-i}, & j \geq i \end{cases}$$

$$= \begin{cases} n_C(1-p_{\text{out}})^j, & j < i \\ n_C(1-p_{\text{out}})^j(\frac{1-p_{\text{in}}}{1-p_{\text{out}}}) - (1-p_{\text{in}})(1-p_{\text{out}})^{j-i}, & j \geq i \end{cases}$$

Hence for $j \geq i$, $n_{C_i}^{(j)} \leq n_C(1-p_{\text{out}})^j(\frac{1-p_{\text{in}}}{1-p_{\text{out}}})$.

$$P_{\text{atom}} = \prod_{j=0}^{C-1} \frac{n_{C_{j+1}}^{(j)}}{\sum_{i=1}^j n_{C_i}^{(j)} + \sum_{i=j+1}^C n_{C_i}^{(j)}} = \prod_{j=0}^{C-1} \frac{n_C(1-p_{\text{out}})^j}{\sum_{i>j} n_C(1-p_{\text{out}})^j + \sum_{i\leq j} n_{C_i}^{(j)}}$$

$$\geq \prod_{j=0}^{C-1} \frac{n_C(1-p_{\text{out}})^j}{\sum_{i>j} n_C(1-p_{\text{out}})^j + \sum_{i\leq j} n_C(1-p_{\text{out}})^j(\frac{1-p_{\text{in}}}{1-p_{\text{out}}})}$$

$$= \prod_{j=0}^{C-1} \frac{1}{C-j+j\cdot(\frac{1-p_{\text{in}}}{1-p_{\text{out}}})} > \prod_{j=0}^{C-1} \frac{1}{C} = 1/C^C$$

Therefore, $P_{\text{FCP,SS}} = C!P_{\text{atom}} > C!/C^C = P_{\text{FCP,uniform}}$. □

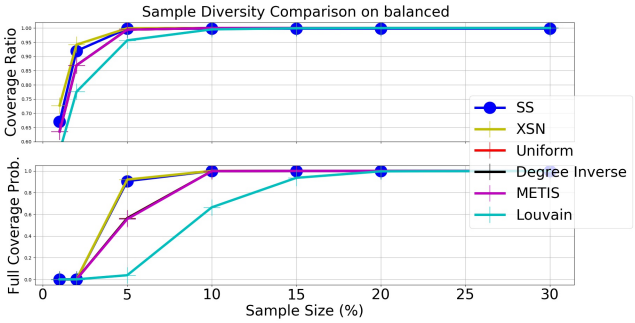
3.3.2 Empirical Results. We compare 5 baseline sampling methods: uniform sampling, expanding snow-ball (XSN [19]), degree-inverse sampling, Louvain[2]+ stratification sampling, and METIS[14]+ stratification sampling. XSN is a greedy sampling method to sample a community-diversified connected component, and performs very well on graphs with balanced communities. We compare against degree-inverse because our method has degree-inverse property for small removal threshold. We show that simply sampling nodes with degree-inverse probability cannot achieve as high community coverage as our method. For the stratification sampling, we first run state-of-the-art community detection methods, then sample the nodes with probability proportional to the sizes of the detected communities.[4] Random-walk-based sampling [16] methods, including multi-walkers [20], have inferior community coverage, and hence we do not include those results.

We report results on graphs with balanced and imbalanced (ground-truth) communities. Algorithms differ with each other on small budgets, and all algorithms achieve high community coverage as the sampling budget increases. SS achieves a comparable result to XSN, and is much better than other baselines on graphs with balanced communities (Figure 4a). SS achieves significantly better results than baselines on imbalanced communities (Figures 4b to 4c). Any sampling method will eventually achieve 100% coverage as the sampling budget increases (100% sampling budget will, of course, cover all the communities). For a fixed sampling method, the saturation point (100% coverage) is earlier (smaller sampling budget required) for graphs with balanced and few communities than graphs with imbalanced and many communities, which implies that it is more difficult to achieve high diversity on graphs with imbalanced communities than on graphs with balanced communities. Figure 4 shows that the SS method indeed achieves better community coverage than the baselines.

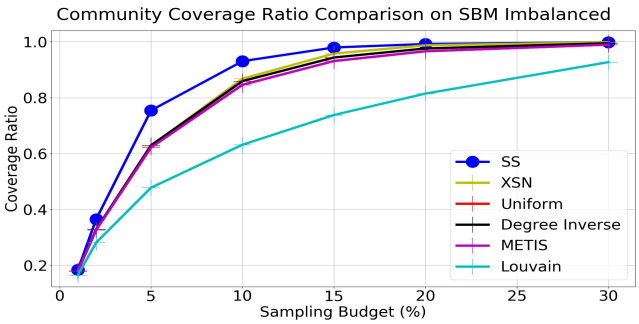
[2]By design, our method achieves 100% expansion quality, ratio of the neighborhood size of the sample to the number of unsampled nodes, as defined in [19] when the infection rate is exactly one node and removal threshold is one.

[3]We use summation instead of integral on the fraction t because t is a ratio computed from a discrete event.

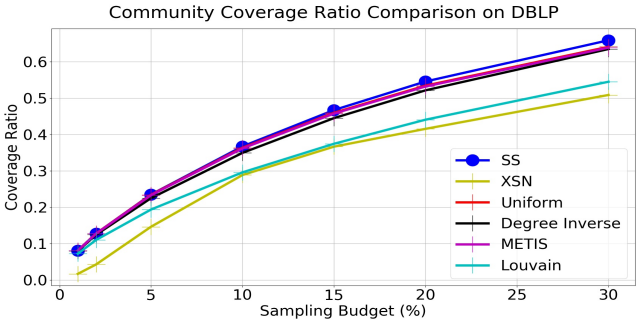
[4]Evaluation is always performed on the ground truth communities.



(a) Sample diversity comparison on SBM10k Balanced. (100 communities of size 100) At very small sample budget (1%, 2%), SS is comparable to XSN, and both have much higher probability to cover all the communities than the baselines. As the sample budget increases, all methods catch up with the front runners. Full coverage probability is more discriminative than coverage ratio.



(b) Community Coverage Ratio on SBM10k Imbalanced. At very small sample budget (1%, 2%), SS covers significantly more communities than the baselines.



(c) DBLP (13.7k real-world communities) graph has many communities of which the sizes are highly imbalanced. No sampling method can cover all the communities even at 30% sampling budget. SS consistently outperforms all the baselines on all budgets.

Figure 4: Community diversity comparisons. SS has good performance on both balanced communities, on which XSN performs well, and imbalanced communities, on which METIS+uniform is the second best.

Our SS method not only achieves better community coverage than baselines, but also runs faster than both XSN and community-detection-based methods. We compare the time efficiency of SS against XSN and Louvain/METIS + stratification[5]. SS has time complexity $O(|V|)$ (Section 3.4) and hence is much faster than Louvain ($O(|V|\log|V|)$ and METIS ($O(|V|\log|C|)$). We report the running time of sample budget being 30% for all the algorithms (since this is the slowest scenario of SS among our experiments, $k = 1, q = 0.005$ in Algorithm 1). We run all the algorithms 100 times on various graphs and report the corresponding total running time. Experiments in Table 3 show that SS is indeed much faster than the baselines.

Table 3: Running Time Comparison

	Python		C/C++		
Graph	Louvain (ratio to SS)	SS	METIS (ratio to SS)	XSN (ratio to SS)	SS
SBM-b	3.2 h (96X)	**2 min**	17s (10X)	51s (32X)	**1.6s**
SBM-i	1.5 h (60X)	**1.5 min**	11s (2X)	34s (5X)	**6.2s**
amazon	35 h (52X)	**40min**	**1.3 min** (0.3X)	10h (162X)	3.7 min
dblp	60 h (7X)	**8.3 h**	11 min (3.3X)	9h (164X)	**3.3 min**
lj	N/A	N/A	3.1h (2X)	DNF	**1.7h**

3.4 Complexity Analysis

According to Algorithm 1, there are $|C_t|$ candidate nodes in the while-loop at the $t-$th iteration, $|C_{t+1}| = (1-q)|C_t|$ for infection rate q. The bottleneck is line 4, candidate nodes update: for each candidate node, we need to scan all its neighbors to determine the validity of its candidacy. This procedure incurs $|C_t|\bar{d}$ queries per iteration, where $\bar{d} = 2|E|/|V|$ is the mean degree. Hence, the overall time complexity is $O(\sum_{C_t=0}^{C_t=n}|C_t|\bar{d}) \overset{\text{geometric sum w/ ratio } 1-q}{=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=} O(|C_0|\bar{d}) = O(n\bar{d}) = O(|E|) \overset{\text{sparse [3], } |E|=O(|V|)}{=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=\!=} O(|V|)$. Therefore, SS has **linear** time complexity. In particular, for infection rate being exactly one node as in Algorithm 2, $O(\sum|C_t|\bar{d}) = O(\bar{d}n^2) \overset{\text{sparse}}{=\!=\!=\!=\!=} O(|V|^2)$, as shown in Figure 5. We do not store edge information. All the edge information is retrieved from the graph. Hence the space complexity is $O(|V|)$.

3.5 Impact of Sampling Parameters

The spread sampling method has two parameters: single-step infection rate q, and removal threshold k. Our theoretical analyses of sampling probability and community coverage study the special case of $k = 1$ and q being exactly one node for tractability. Section 3.4 shows that infection rate being exactly one node leads to quadratic time complexity while fractional infection rate has only linear time complexity. In practice, we choose q and k according to our purpose/application. We empirically (Figure 6) find the infection rate has little impact on community diversity. Hence we can get high community diversity sampling with linear time complexity in practice.

4 APPLICATION

4.1 Overlapping Community Detection Seeding

Personalized PageRank (PPR) [1] is a well-established method for local graph partitioning: to identify a community from a small

[5]We implement SS in both Python and C++ (2 implementations). Louvain method (python-louvain v0.9) is in Python while METIS (networkx-metis v1.0) is in C with a Python wrapper.

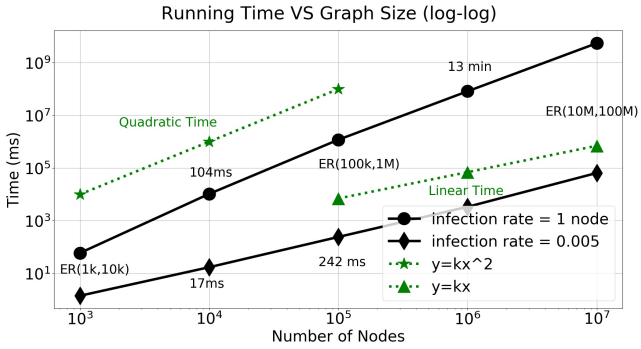


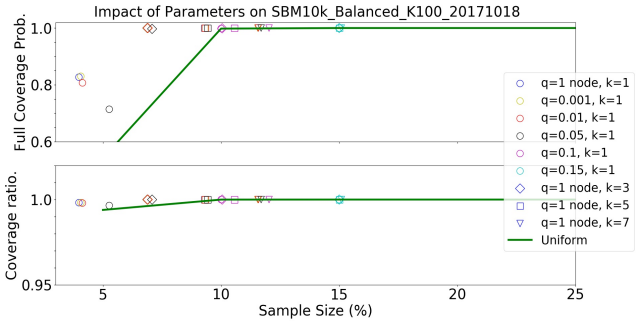
Figure 5: Scalability of spread sampling method (C++). The running time is obtained from ER graphs with mean degree 10 of varying sizes. Diamond markers correspond to infection rate being 0.005 and hence linear; circles correspond to infection rate being exactly one node as in Algorithm 2 (vertically shifted for separation). The green dotted lines are for reference.

seed set. An extensive comparison by [15] shows that PPR has the best performance among all local partitioning methods. They focus on the comparison of different methods, and claim that uniform sampling is better than high-degree node sampling. Our spread sampling, with different parameter settings, can be specialized as uniform sampling ($k > d_{\max}$) or low-degree nodes sampling (small k). We run PPR community detection following the same procedure and on the same datasets (with one more, LiveJournal) as in Sect. 2 of [15] and compare various seeding methods: SS with small k (low-degree heuristic), SS with large k, uniform sampling and high-degree heuristic.[6] For each ground truth community, we try the sampling budget to be 5% and 10%, and they return consistent results in terms of seeding methods ranking. PPR assigns each node a conductance score in $[0, 1]$. For each to-be-detected community, we sort all the nodes in descending order of the score and determine the top-C nodes as the community members, where C is the size of the ground truth community. For SS, we fix q to be exact one node per step, and vary removal threshold k from 1 to 50. Note that large k degenerate to uniform sampling. Our key finding is that low-degree heuristic works the best in overlapping community detection, and the recall of each seeding methods is reported in Table 4.[7]

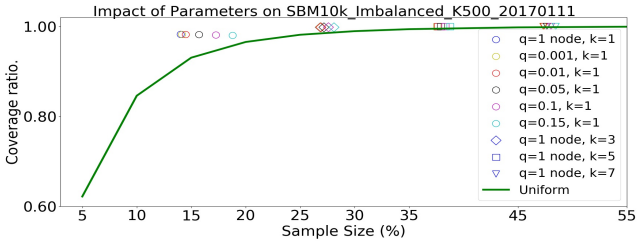
Experiments show that low-degree heuristic works best on graphs with overlapping communities: high-degree nodes belong to several communities while low-degree nodes are "core members" of a community [22, 28]. Expansion from a high degree node results in sample from different communities and hence the recall is low; expansion from "core members" is more likely to return nodes within the same community and hence a better recall. The recall on youtube graph is very low, which is probably due to its low clustering coefficient.

[6]The neighborhood inflation method [27] is also a highly cited work. We did not compare against it since according to [15], PPR performs better than the neighborhood inflation method.

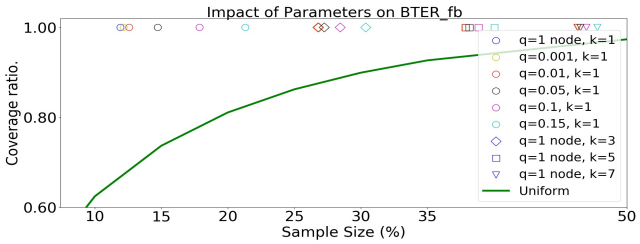
[7]Precision$\equiv \frac{\text{True Positive}}{|C_{\text{detect}}|} \overset{\text{exp. setup}}{=\!=\!=} \frac{\text{TP}}{|C_{\text{ground truth}}|} \equiv$recall



(a) SBM10k with balanced communities. Full coverage probability (up) is a more discriminative metric than the coverage ratio (bottom). Infection rates are separated by different colors while removal thresholds by different shapes. All parameter combinations outperform uniform sampling. Note that sampling configurations with the same k, removal threshold (in the same shape), tends to cluster together, validating the claim that k controls the end-of-procedure status.



(b) SBM10k with imbalanced communities. SS outperforms uniform sampling in all parameter settings, and parameter settings with the same k tends to cluster together. Uniform sampling cannot achieve non-zero full coverage probability (FCP) even with 50% sampling budeget while SS can achieve above 60% FCP even at 10% budget, so we omit the FCP plot.



(c) BTER fitted with a Facebook graph with imbalanced communities. SS significantly outperforms the baseline in all parameter settings, and parameter settings with the same k cluster together. SS achieves near 100% FCP at budget below 15% while uniform sampling cannot achieves non-zero FCP at budget 30%.

Figure 6: Impact of sampling parameters in terms of community coverage. Parameter settings with the same k cluster together, validating the intuition that k controls the end-of-procedure status. The high community coverage of SS is more significant on graphs with imbalanced communities (b), (c).

Table 4: Community Detection Recall with Varying Seed Strategies

Data	k=1	k=10	Uniform	Max Deg.
amazon	.5768±.038	**.5808±.040**	.5617±.042	.5612±.043
dblp	**.2512±.004**	.2396±.003	.2383±.004	.1479±.001
lj	**.1328±.002**	.1313±.003	.1311±.002	.1123±.001
youtube	**.0227±.004**	.0200±.003	.0138±.002	.0108±.001

5 CONCLUSIONS

We propose a simple yet elegant procedure - spread sampling (SS)- for sampling nodes within a graph. We show that spread sampling tries to sample nodes from all regions of the graph thereby improving community diversity than existing baselines, especially on the networks with imbalanced communities. We show that SS samples nodes with sampling probability following a power law distribution on both community-structured graphs and community-free graphs. We show that SS achieves high community diversity on various graphs with only linear time complexity. We apply SS on seeding-based community detection, and show it outperforms the-state-of-the-art. In the future, we will find an analytical sampling probability on SBM graphs, and apply SS to network A/B testing, community-based change point detection [25, 26] and community-based question-answer matching problem [24].

REFERENCES

[1] Reid Andersen, Fan Chung, and Kevin Lang. 2006. Local graph partitioning using pagerank vectors. In *FOCS'06*. 475–486.

[2] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

[3] Fan Chung. 2010. Graph theory in the information age. *Notices of the AMS* 57, 6 (2010), 726–732.

[4] Amin Coja-Oghlan and Charilaos Efthymiou. 2015. On independent sets in random graphs. *Random Structures & Algorithms* 47, 3 (2015), 436–486.

[5] Santo Fortunato. 2010. Community detection in graphs. *Physics reports* 486, 3 (2010), 75–174.

[6] Michelle Girvan and Mark EJ Newman. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences* 99, 12 (2002), 7821–7826.

[7] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. 2010. Walking in facebook: A case study of unbiased sampling of osns. In *Infocom, 2010 Proceedings IEEE*. IEEE, 1–9.

[8] Huan Gui, Ya Xu, Anmol Bhasin, and Jiawei Han. 2015. Network a/b testing: From sampling to estimation. In *Proceedings of the 24th International Conference on World Wide Web*. ACM, 399–409.

[9] Peter Hall. 2013. *The bootstrap and Edgeworth expansion*. Springer Science & Business Media.

[10] David J Hand. 2010. Statistical Analysis of Network Data: Methods and Models by Eric D. Kolaczyk. *International Statistical Review* 78, 1 (2010), 135–135.

[11] Mark S Handcock and Krista J Gile. 2010. Modeling social networks from sampled data. *The Annals of Applied Statistics* 4, 1 (2010), 5.

[12] D. G. Horvitz and D. J. Thompson. 1952. A Generalization of Sampling Without Replacement From a Finite Universe. *J. Amer. Statist. Assoc.* 47, 260 (1952), 663–685. http://www.jstor.org/stable/2280784

[13] Brian Karrer and Mark EJ Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical review E* 83, 1 (2011), 016107.

[14] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing* 20, 1 (1998), 359–392.

[15] Isabel M Kloumann and Jon M Kleinberg. 2014. Community membership identification from small seed sets. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1366–1375.

[16] Jure Leskovec and Christos Faloutsos. 2006. Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 631–636.

[17] Jure Leskovec and Rok Sosič. 2016. SNAP: A General-Purpose Network Analysis and Graph-Mining Library. *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, 1 (2016), 1.

[18] Sharon Lohr. 2009. *Sampling: design and analysis*. Nelson Education.

[19] Arun S Maiya and Tanya Y Berger-Wolf. 2010. Sampling community structure. In *Proceedings of the 19th international conference on World wide web*. ACM, 701–710.

[20] Bruno Ribeiro and Don Towsley. 2010. Estimating and sampling graphs with multidimensional random walks. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 390–403.

[21] Yiye Ruan, David Fuhry, Jiongqian Liang, Yu Wang, and Srinivasan Parthasarathy. 2015. Community Discovery: Simple and Scalable Approaches. In *User Community Discovery*. Springer, 23–54.

[22] Yiye Ruan and Srinivasan Parthasarathy. 2014. Simultaneous detection of communities and roles from large networks. In *Proceedings of the second ACM conference on Online social networks*. ACM, 203–214.

[23] Comandur Seshadhri, Tamara G Kolda, and Ali Pinar. 2012. Community structure and scale-free collections of Erdős-Rényi graphs. *Physical Review E* 85, 5 (2012), 056109.

[24] Jiankai Sun, Sobhan Moosavi, Rajiv Ramnath, and Srinivasan Parthasarathy. 2018. QDEE: Question Difficulty and Expertise Estimation in Community Question Answering Sites. In *ICWSM*.

[25] Yu Wang, Aniket Chakrabarti, David Sivakoff, and Srinivasan Parthasarathy. 2017. Fast change point detection on dynamic social networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2992–2998.

[26] Yu Wang, Aniket Chakrabarti, David Sivakoff, and Srinivasan Parthasarathy. 2017. Hierarchical Change Point Detection on Dynamic Networks. In *Proceedings of the 2017 ACM on Web Science Conference*. ACM, 171–179.

[27] Joyce Jiyoung Whang, David F Gleich, and Inderjit S Dhillon. 2013. Overlapping community detection using seed set expansion. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. ACM, 2099–2108.

[28] Jaewon Yang and Jure Leskovec. 2012. Structure and overlaps of communities in networks. *arXiv preprint arXiv:1205.6228* (2012).