Star sampling with and without replacement

Jonathan Stokes and Steven Weber
Drexel University
Department of Electrical and Computer Engineering
Philadelphia, Pennsylvania

ABSTRACT

Star sampling (SS) is a graph search mechanism wherein each sample consists of a vertex (the star center) and its one-hop neighbors (the star points). We consider the use of star sampling to find any vertex in a specified target set in a large graph, where the figure of merit is the expected number of samples until a vertex in the target set is encountered, either as a star center or as a star point. We analyze this performance measure on three related star sampling paradigms: SS with replacement (SS-R), SS without center replacement (SS-C), and SS without star replacement (SS-S). Exact expressions for the average number of samples under SS-R and SS-C are easily obtained. Much of the paper is focused on deriving an approximate expression for the performance of SS-S. Experiments are run on both "synthetic" graphs, i.e., Erdős-Rényi (ER) graphs, as well as three "real-world" graphs. The two contributions of the paper are: *i*) the analytical approximation for SS-S is seen to be quite accurate for both types of graphs, *ii*) we observe, perhaps surprisingly, there is little performance difference across the three sampling paradigms. This performance insensitivity of SS-R relative to SS-S may be understood as the result of two competing factors: removing stars reduces the number of vertices outside the target set, but also removes the number of neighbors of the target set.

KEYWORDS

Star sampling; graph search; graph sampling; performance analysis.

ACM Reference format:
Jonathan Stokes and Steven Weber. 2017. Star sampling with and without replacement. In *Proceedings of KDD Mining and Learning with Graphs Workshop, Halifax, Nova Scotia, Aug. 2017 (KDD MLG Workshop 2017),* 8 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Large graphs, e.g., graphs of social networks, are often too large or too dynamic to be held in local memory, and as such finding vertices with a given property requires the searcher *query* the graph, requesting either a random vertex (e.g., as in sampling) or a particular vertex (e.g., as in guided search). One query option, a *star sample* (SS) in the context of a graph $G = (V, E)$, refers to a vertex $v \in V$, which we henceforth term the *star center*, and its one-hop neighbors $\Gamma(v)$, which are henceforth termed the *star points*. Star sampling arises naturally under the assumption that the property of a vertex being sampled for is revealed whether the vertex or a vertex in its neighborhood is sampled. As a means to find any vertex in a target subset $T \subset V$, where T is those vertices holding the property of interest, star sampling refers to a family of graph search mechanisms wherein star centers are repeatedly picked from the graph, a process which terminates if the star center or any of the star points lies in T.

In our simulations we consider the specific graph search problem of finding one or more vertices of a target degree. For example, in the context of a social network graph one may want to identify an individual with a large number of contacts. As we are searching for a vertex with a particular degree, we make the assumption that the graph query, say $\text{query}_G(v)$ returns *i*) the labels of the neighbors of v, and *ii*) the degrees of each neighbor. We consider each query as requiring unit (time or processing) cost, although it is also possible and reasonable to consider a computation model wherein the cost is linearly proportional to the degree.

We consider three related versions of star sampling:

- *Star sampling with replacement (SS-R)*: the star center is selected uniformly at random from V;
- *Star sampling without center replacement (SS-C)*: the star center is selected uniformly at random from the set of remaining vertices, and the star center (along with its adjacent edges) is removed from the graph after the query;
- *Star sampling without star replacement (SS-S)*: the star center is selected uniformly at random from the set of remaining vertices, and the entire star (center and points and all adjacent edges) are removed from the graph after the query.

The natural performance measure of a search algorithm is the (expected value of the) number of queries until a member of the target set is found. Our motivation in considering these variants is to understand their relative performance, in a manner similar to the elementary case of sampling balls from an urn. When seeking any one of k marked balls out of a total of n balls in an urn, sampling with replacement requires on average n/k samples; this follows immediately from the observation that the number of draws, say N, is a geometric random variable (RV) with success probability k/n, and expectation $\mathbb{E}[\mathsf{N}] = n/k$. In contrast, sampling without replacement requires a random number of draws, $\bar{\mathsf{N}}$, with expectation $\mathbb{E}[\bar{\mathsf{N}}] = (n+1)/(k+1)$. Thus, the performance ratio of the expected number of samples with vs. without replacement is $\mathbb{E}[\mathsf{N}]/\mathbb{E}[\bar{\mathsf{N}}] = (k+1)/k$, which shows sampling without replacement improves the average search time by a factor of two, relative to sampling with replacement, in the particular case of $k = 1$.

With this elementary example as motivation, we ask what performance improvement is achieved by star sampling without replacement relative to star sampling with replacement? The two primary contribution of this paper are:

This work was supported by the National Science Foundation #IIS-1250786. S. Weber is the contact author: sweber@coe.drexel.edu.

KDD MLG Workshop 2017, Halifax, Nova Scotia
© 2017 ACM. 978-x-xxxx-xxxx-x/YY/MM...$15.00
DOI: 10.1145/nnnnnnn.nnnnnnn

- Analytical approximations of the performance of the three star sampling variants; comparisons of these approximations with simulation results show the approximations to be quite accurate.
- The observation that the performance of the three star sampling variants are more or less identical; this may be explained by the fact that although SS-S has the benefit (relative to SS-R) of reducing the number of vertices outside the target set, it also has the cost of reducing the number of neighbors of the target set. This benefit and cost apparently approximately cancel each other out, at least in the graphs we have studied.

The rest of this paper is organized as follows. Basic notation and definitions are given in §2. §3 shows by example that no ordering on the performance of the three star sampling variants holds for all graphs. An approximate evolution of the degree distribution in time, as star samples are removed from the graph, is leveraged in the main result of the paper, Prop. 4.6, which gives an approximation of the number of samples required under SS-S. Numerical and simulation results, for both synthetic and "real-world" graphs, are given in §5, related work is discussed in §6, and a conclusion is offered in §7.

2 NOTATION, DEFINITIONS, SIMPLE FACTS

Let $a \equiv b$ denote a and b are equal by definition. Let $[n]$ denote $\{1, \ldots, n\}$ for $n \in \mathbb{N}$. Random variables are denoted in a sans-serif font, e.g., $\mathsf{x}, \mathsf{N}, \mathsf{u}$, expectation is denoted $\mathbb{E}[\cdot]$, and probability is denoted $\mathbb{P}(\cdot)$. If U is a set then $\mathsf{u} \sim \mathrm{Uni}(U)$ denotes a random member of U selected uniformly at random. We use the following graph notation:

- *Order, size, edges.* An undirected and simple graph of order n is denoted $G = (V, E)$, with vertex set $V = [n]$ and edge set E; size is denoted by $m \equiv |E|$. An undirected edge is denoted ij or $\{i, j\}$.
- *Neighborhoods.* Let $\Gamma(v)$ be the (one hop) neighbors of v, $\Gamma^c(v) \equiv \Gamma(v) \cup \{v\}$ the extended neighborhood of v, and $\mathcal{N}(v) \equiv \{uv \in E\}$ the edge neighborhood of v, i.e., the edges adjacent to v. Observe $\Gamma^c(v)$ is a star sample with star center v and star points $\Gamma(v)$. For $T \subseteq V$, let $\Gamma(T) \equiv \bigcup_{v \in T} \Gamma(v) \setminus T$ denote neighbors of T not including T; let $\Gamma^c(T) \equiv \bigcup_{v \in T} \Gamma^c(v)$ denote T and its neighbors.
- *Degrees.* Let $d(v) \equiv |\Gamma(v)|$ be the degree of v, let $\mathbf{d} \equiv (d(v), v \in V)$ be the degree sequence of the graph, and $D \equiv \bigcup_{v \in V} d(v)$ be the set of degrees found in G, with $\phi \equiv \max D$ the maximum degree. Partition V by D into $(V_k, k \in D)$, with $V_k \equiv \{v \in V : d(v) = k\}$ the vertices of degree k, and $n_k \equiv |V_k|$ the number of degree k vertices. Let $\mathbf{n} \equiv (n_k, k \in D)$ be the degree counts, and let $\mathbf{w} \equiv (w_k, k \in D)$ be the vertex degree distribution, with $w_k \equiv n_k/n$. Let $\mu \equiv \sum_{k \in D} k w_k = \mathbb{E}[d(\mathsf{v})]$ be the expected degree of a randomly selected vertex, for $\mathsf{v} \sim \mathrm{Uni}(V)$.
- *Stubs.* Viewing each edge $e \in E$ as a pair of edge "stubs", set $S = [2m]$ as the set of $2m$ stubs, and set d_s as the degree of the vertex for stub s. Let $(d_s, s \in S)$ be the stub degree sequence of the graph, and partition this set by D into $(S_k, k \in D)$, with S_k the stubs tied to degree k vertices, and $m_k \equiv |S_k|$ the number of such stubs. Let $\mathbf{m} \equiv (m_k, k \in D)$ be the stub degree counts, and let $\mathbf{q} \equiv (q_k, k \in D)$ be the stub degree distribution, with $q_k \equiv m_k/(2m)$. Let $\nu \equiv \sum_{k \in D} k q_k = \mathbb{E}[d_{\mathsf{s}}]$ be the expected degree of a randomly selected stub, i.e., $\mathsf{s} \sim \mathrm{Uni}(S)$.

Fix the initial graph $G_0 = (V_0, E_0)$ and fix the target set $T \subseteq V_0$. Recalling §1, we consider three star sampling variants. Unadorned notation denotes SS-R, a bar denotes SS-C, and a tilde denotes SS-S.

- *Star sampling with replacement (SS-R)*: generate an iid sequence $(\mathsf{v}_t, t \in \mathbb{N})$ of RVs, with each $\mathsf{v}_t \sim \mathrm{Uni}(V_0)$.
- *Star sampling without center replacement (SS-C)*: generate the random sequence $(\bar{\mathsf{v}}_t, t \in \mathbb{N})$, and define the associated random graph sequence $(\bar{\mathsf{G}}_t, t \in \mathbb{N})$, with $\bar{\mathsf{G}}_t = (\bar{\mathsf{V}}_t, \bar{\mathsf{E}}_t)$ the graph after sample t. In particular, each star center is drawn uniformly at random from the previous graph, i.e., $\bar{\mathsf{v}}_t \sim \mathrm{Uni}(\bar{\mathsf{V}}_{t-1})$, and the graph is updated to reflect deletion of the center node: $\bar{\mathsf{V}}_t = \bar{\mathsf{V}}_{t-1} \setminus \bar{\mathsf{v}}_t$ and $\bar{\mathsf{E}}_t = \bar{\mathsf{E}}_{t-1} \setminus \bar{\mathcal{N}}_{t-1}(\bar{\mathsf{v}}_t)$.
- *Star sampling without star replacement (SS-S)*: generate the random sequence $(\tilde{\mathsf{v}}_t, t \in \mathbb{N})$, and define the associated random graph sequence $(\tilde{\mathsf{G}}_t, t \in \mathbb{N})$, with $\tilde{\mathsf{G}}_t = (\tilde{\mathsf{V}}_t, \tilde{\mathsf{E}}_t)$ the graph after sample t. In particular, each star center is drawn uniformly at random from the previous graph, i.e., $\tilde{\mathsf{v}}_t \sim \mathrm{Uni}(\tilde{\mathsf{V}}_{t-1})$, and the graph is updated to reflect deletion of the star:

$$\tilde{\mathsf{V}}_t = \tilde{\mathsf{V}}_{t-1} \setminus \tilde{\Gamma}^c_{t-1}(\tilde{\mathsf{v}}_t), \ \tilde{\mathsf{E}}_t = \tilde{\mathsf{E}}_{t-1} \setminus \bigcup_{v \in \tilde{\Gamma}_{t-1}(\tilde{\mathsf{v}}_t)} \tilde{\mathcal{N}}_{t-1}(v). \qquad (1)$$

That is, the new edge set $\tilde{\mathsf{E}}_t$ is obtained by removing the edge neighborhood $\tilde{\mathcal{N}}_{t-1}(v)$ for each neighbor $v \in \tilde{\Gamma}_{t-1}(\tilde{\mathsf{v}}_t)$ of $\tilde{\mathsf{v}}_t$.

The performance measures for the three variants are defined below.

Definition 2.1. Performance of the three SS variants is defined as the expected number of samples until a star, either the star center or one of the star points, intersects the target set T, i.e., $\mathbb{E}[\mathsf{N}]$, $\mathbb{E}[\bar{\mathsf{N}}]$, and $\mathbb{E}[\tilde{\mathsf{N}}]$, respectively, where

$$\begin{array}{lccl} \text{SS-R:} & \mathsf{N} & \equiv & \min\{t : \Gamma^c_{t-1}(\mathsf{v}_t) \cap T \neq \emptyset\} \\ \text{SS-C:} & \bar{\mathsf{N}} & \equiv & \min\{t : \bar{\Gamma}^c_{t-1}(\bar{\mathsf{v}}_t) \cap T \neq \emptyset\} \\ \text{SS-S:} & \tilde{\mathsf{N}} & \equiv & \min\{t : \tilde{\Gamma}^c_{t-1}(\tilde{\mathsf{v}}_t) \cap T \neq \emptyset\} \end{array} . \qquad (2)$$

Recall $\Gamma^c_0(T)$ contains T and its neighbors in G_0. Observe the equivalence: a star sample $\Gamma^c_0(v)$ intersects T if and only iff $v \in \Gamma^c_0(T)$. This observation yields the performance of SS-R and SS-C.

Fact 1 (Performance of star sampling with replacement (SS-R)). *The RV* $\mathsf{N} \sim \mathrm{geo}(|\Gamma^c_0(T)|/n)$ *in (2) is a geometric RV with success probability* $p \equiv |\Gamma^c_0(T)|/n$, *and expectation* $\mathbb{E}[\mathsf{N}] = n/|\Gamma^c_0(T)|$.

Fact 2 (Performance of star sampling without center replacement (SS-C)). *The RV* $\bar{\mathsf{N}}$ *has* $\mathbb{E}[\bar{\mathsf{N}}] = (n+1)/(|\Gamma^c_0(T)| + 1)$.

Proof. Observe SS-C with target set T on a graph of order n is equivalent to sampling without replacement from an urn with n balls of which $k = |\Gamma^c_0(T)|$ are marked. □

Fact 3 (SS-C outperforms SS-R). *The expected number of star samples with replacement (SS-R) exceeds the expected number of star samples without center replacement (SS-C):* $\mathbb{E}[\mathsf{N}] \geq \mathbb{E}[\bar{\mathsf{N}}]$.

Proof. This follows immediately from Fact 1 and Fact 2. A more intuitive proof, however, is as follows. Let $(\bar{p}_t, t \in \mathbb{N})$ denote the probability of success under SS-C in trial t, where $\bar{p}_t = |\Gamma^c(T)|/(n - t + 1)$ on account of one vertex from $V \setminus \Gamma^c(T)$ being removed after each failed star sample. Then $\bar{p}_t \geq p$ (recall Fact 1), which implies $\mathbb{E}[\mathsf{N}] \geq \mathbb{E}[\bar{\mathsf{N}}]$. □

3 ORDERING THE THREE VARIANTS

The purpose of this section is to provide examples demonstrating that, in contrast with SS-C and SS-R (c.f., Fact 3), there is no guaranteed ordering of SS-C and SS-S, or SS-R and SS-S. We provide examples demonstrating these facts.

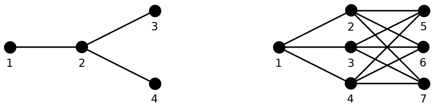

Figure 1: Left: graph $G^{(1)}$, for which SS-C outperforms SS-S. Right: $G^{(2)}$, for which SS-R outperforms SS-S.

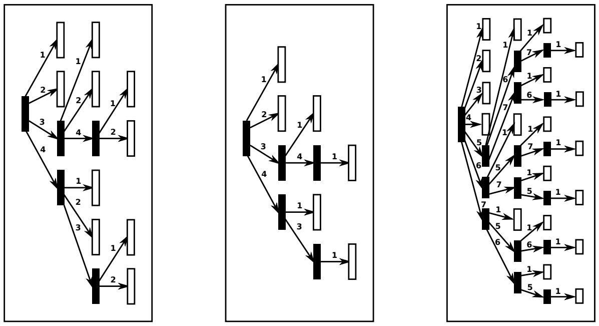

Figure 2: Left: outcome tree for $SS-C$ on $G^{(1)}$. Center: outcome tree for $SS-S$ on $G^{(1)}$. Right: outcome tree for $SS-S$ on $G^{(2)}$. White blocks are terminating states.

3.1 SS-C may outperform SS-S

The expected performance under SS-C and SS-S on a given graph may be analyzed by making outcome trees, as illustrated in Fig. 2 for the graphs in Fig. 1. Each level of the tree corresponds to a time instant $t \in \mathbb{Z}_+$, with the root node, corresponding to $t = 0$, the initial graph $G_0 = (V_0, E_0)$. Each vertex in the tree at level t corresponds to a particular graph possible at time t, say $\bar{G}_t$ for SS-C and $\tilde{G}_t$ for SS-S. Each labeled edge in the tree, connecting a graph at time t with a graph at time $t+1$, corresponds to a choice of the star center at sample $t \in \mathbb{N}$. For each vertex v in the tree, corresponding to, say, a graph $G(v) = (V(v), E(v))$, there is a collection of edges in the tree, emanating from v, one edge for each vertex in $G(v)$, corresponding to the possible star centers that may be chosen from $G(v)$. Each of these edges has a probability of $1/|V(v)|$, as each vertex in $V(v)$ is equally likely to be selected.

Leaf vertices in the tree are terminating states, representing the fact that the target set T has been hit for the first time. Let $\bar{\mathcal{L}}$ and $\tilde{\mathcal{L}}$ denote the leaves in the outcome trees for a given graph under SS-C and SS-S, respectively. Each leaf has a unique path to the root node, and the probability of the leaf is the product of the probabilities assigned to the edges comprising that path. Define $\bar{P} \equiv (\bar{P}_L, L \in \bar{\mathcal{L}})$ and $\tilde{P} \equiv (\tilde{P}_L, L \in \tilde{\mathcal{L}})$ as the probability distributions on the leaves of the outcome trees under SS-C and SS-S, respectively. Finally, observe each leaf L has a depth, denoted $\bar{N}_L, \tilde{N}_L \in \mathbb{N}$, i.e., a length of the path from the root, and this corresponds to the number of samples until the target set was hit. It follows that

$$\mathbb{E}[\bar{\mathsf{N}}] = \sum_{L \in \bar{\mathcal{L}}} \bar{N}_L \bar{P}_L, \ \mathbb{E}[\tilde{\mathsf{N}}] = \sum_{L \in \tilde{\mathcal{L}}} \tilde{N}_L \tilde{P}_L. \tag{3}$$

Fact 4 (SS-C may outperform SS-S). *There exist graphs for which star sampling without replacement of center outperforms star sampling without replacement of star, i.e.,* $\mathbb{E}[\bar{\mathsf{N}}] < \mathbb{E}[\tilde{\mathsf{N}}]$.

Proof. Fix $G^{(1)}$ in Fig. 1 and fix the target set $T = \{1\}$. The outcome tree shown in the left figure in Fig. 2, corresponding to running SS-C on $G^{(1)}$, has

$\bar{M}_L$	$\bar{N}_L$	$\bar{P}_L$	$\bar{M}_L \bar{N}_L \bar{P}_L$
2	1	1/4	1/2
4	2	1/12	2/3
4	3	1/24	1/2

(4)

where $\bar{M}_L$ is the number of leaf vertices of type $(\bar{N}_L, \bar{P}_L)$. Adding up the right column gives $\mathbb{E}[\bar{\mathsf{N}}] = 5/3$. The outcome tree shown in the middle figure in Fig. 2, corresponding to running SS-S on $G^{(1)}$, has

$\tilde{M}_L$	$\tilde{N}_L$	$\tilde{P}_L$	$\tilde{M}_L \tilde{N}_L \tilde{P}_L$
2	1	1/4	1/2
2	2	1/8	1/2
2	3	1/8	3/4

(5)

Summation gives $\mathbb{E}[\tilde{\mathsf{N}}] = \frac{7}{4}$. Thus $\frac{5}{3} = \mathbb{E}[\bar{\mathsf{N}}] < \mathbb{E}[\tilde{\mathsf{N}}] = \frac{7}{4}$. □

3.2 SS-R may outperform SS-S

Fact 5 (SS-R may outperform SS-S). *There exist graphs for which star sampling with replacement outperforms star sampling without replacement of star, i.e.,* $\mathbb{E}[\mathsf{N}] < \mathbb{E}[\tilde{\mathsf{N}}]$.

Proof. Fix $G^{(2)}$ in Fig. 1 and fix the target set $T = \{1\}$. Note $|\Gamma^c(T)| = 4$ and $n = 7$, and thus the performance under SS-R is, by Fact 1, $\mathbb{E}[\mathsf{N}] = 7/4$. The outcome tree shown in the right figure in Fig. 2, corresponding to running SS-S on $G^{(2)}$, has

$\tilde{M}_L$	$\tilde{N}_L$	$\tilde{P}_L$	$\tilde{M}_L \tilde{N}_L \tilde{P}_L$
4	1	1/7	4/7
3	2	1/21	5/21
6	3	1/42	3/7
6	4	1/42	4/7

(6)

Summation gives $\mathbb{E}[\tilde{\mathsf{N}}] = \frac{38}{21}$. Thus $\frac{7}{4} = \mathbb{E}[\mathsf{N}] < \mathbb{E}[\tilde{\mathsf{N}}] = \frac{38}{21}$. □

Remark 1. *In both examples above we see the performance of SS-S to be worse than SS-C and SS-R, respectively. These results may be counter-intuitive, as one might expect SS-S to outperform both SS-C and SS-R, on account of the fact that SS-S removes more vertices outside the target set than the other two. However, as these examples show, these vertices outside the target set include the* neighbors *of the target set, and removing them may, as in these examples, hurt the expected performance, as the target set is harder to "hit" with a randomly selected star when it has fewer neighbors.*

4 SS-S PERFORMANCE ANALYSIS

Fix the (initial) graph $G_0 = (V_0, E_0)$ and pick a target set $T \subset V$. Recall from §2 that $\Gamma_0(T)$ denotes neighbors of T in G_0 not in T, while $\Gamma_0^c(T)$ denotes T and its neighbors. The objective in this section is to derive an approximation for the performance under SS-S, $\mathbb{E}[\tilde{\mathsf{N}}]$ (c.f. Def. 2.1), given in Prop. 4.6.

4.1 Degree l 1-neighbors of a degree k vertex

Our starting point in deriving approximations to the degree evolution is to assume an arbitrary graph for which we have knowledge only of the degree set D and the "stub" counts $\mathbf{m} = (m_k, k \in D)$.

For any graph $G = (V, E)$, define the RV $\mathsf{h}_{l|k}^{(1)} \equiv |V_l \cap \Gamma(\mathsf{v})|$, where $\mathsf{v} \sim \mathrm{Uni}(V_k)$, i.e., $\mathsf{h}_{l|k}^{(1)}$ is the number of degree l one-hop neighbors of a degree k vertex, when that vertex is selected uniformly at random from all degree k vertices in the graph.

Given limited knowledge of G, we *define* an approximation for the distribution of $\mathsf{h}_{l|k}^{(1)}$ by assuming all edges are wired uniformly at random, subject to the constraints imposed by the degree count $\mathbf{n}$. We employed a similar approximation in our recent work [30].

Definition 4.1. The approximate distribution of $\mathsf{h}_{l|k}^{(1)}$ is defined as

$$\mathbb{P}(\mathsf{h}_{l|k}^{(1)} = h) \approx p_{l|k}^{(1)}(h) \equiv \frac{\binom{m_l}{h}\binom{2m-m_l}{k-h}}{\binom{2m}{k}}, \; h \in \{0, \ldots, \min\{k, m_l\}\}. \tag{7}$$

We emphasize that we define this approximation; we do not prove it to be accurate. The approximation is justified only in so far as it leads to estimators of the degree evolution under SS-S that appear to align well with simulation results. The intuition behind Def. 4.1 is from balls in an urn, i.e., (7) is the probability that a random k-sample from an urn containing $2m$ balls, of which m_l are marked, contains h marked balls. Under the approximate distribution in Def. 4.1, the expected number of degree l neighbors of a randomly selected degree k vertex is:

$$\mathbb{E}[\mathsf{h}_{l|k}^{(1)}] \approx E_{l|k}^{(1)} \equiv \sum_{h=0}^{\min\{k, m_l\}} h \, p_{l|k}^{(1)}(h). \tag{8}$$

We have the following result for the case when the number of degree l stubs, m_l, exceeds k, the degree of the center vertex.

Proposition 4.2. *If $m_l \geq k$ then $E_{l|k}^{(1)} = kq_l$.*

Proof sketch, when $m_l \geq k$, the expected fraction of neighbors of degree l ($E_{l|k}^{(1)}/k$) equals the fraction of degree l edge stubs ($q_l = m_l/(2m)$), as is intuitive under the assumptions.

4.2 Degree l 2-neighbors of a degree k vertex

For any graph $G = (V, E)$, define the RV $\mathsf{h}_{l|k}^{(2)} \equiv |V_l \cap \Gamma(\Gamma(\mathsf{v}))|$, where $\mathsf{v} \sim \mathrm{Uni}(V_k)$. Note $\Gamma(\Gamma(\mathsf{v}))$ is the set of two-hop neighbors of v, and thus $\mathsf{h}_{l|k}^{(2)}$ is the number of degree l two-hop neighbors of a degree k vertex, when that vertex is selected uniformly at random from all degree k vertices in the graph. We define the following approximation:

Definition 4.3. The approximate expectation of $\mathsf{h}_{l|k}^{(2)}$ is defined as

$$\mathbb{E}[\mathsf{h}_{l|k}^{(2)}] \approx E_{l|k}^{(2)} \equiv \sum_{j \in D} E_{l|j}^{(1)} E_{j|k}^{(1)}. \tag{9}$$

This approximation asserts that the expected number of degree l two-hop neighbors is found from the expected number of one-hop neighbors by decomposing all one-hop neighbors by their degree. The approximation is quite crude in that it ignores the double counting that results when a given two-hop neighbor is adjacent to multiple one-hop neighbors. Again, we provide no analytical proof of the validity of this approximation, other than to show that it leads to estimators of the degree evolution under SS-S that align well with our simulation results. The intuition behind (9) is that the $E_{j|k}^{(1)}$ degree j one-hop neighbors of the center node k will each connect with $E_{l|j}^{(2)}$ degree l neighbors. Recall $\nu \equiv \sum_{j \in D} jq_j$ is the average degree of a randomly selected stub. The following holds.

Fact 6. *If $m_j \geq j$ for each $j \in D$ then $E_{l|k}^{(2)} = kq_l\nu$.*

Proof. Under the assumption we may leverage Prop. 4.2:

$$E_{l|k}^{(2)} = \sum_{j \in D} (jq_l)(kq_j) = kq_l \sum_{j \in D} jq_j = kq_l\nu. \tag{10}$$

□

4.3 Change in number of degree l vertices

Consider an arbitrary graph $G = (V, E)$ from which a star sample is drawn with star center of degree k. Under SS-S, we remove the star center and the star points. The approximate expected number of degree l vertices that are removed from G by selecting a degree k vertex uniformly at random is, by Def. 4.1 and Prop. 4.2:

$$F_{l|k}^{(1)} \equiv \begin{cases} E_{l|k}^{(1)}, & l \neq k \\ E_{k|k}^{(1)} + 1, & \text{else} \end{cases} \tag{11}$$

Observe $E_{k|k}^{(1)} + 1$ is the approximate expected number of degree k vertices in a star sample with star center of degree k. Define

$$F_{l|k}^{(2)} \equiv E_{l+1|k}^{(2)} - E_{l|k}^{(2)}. \tag{12}$$

Observe $F_{l|k}^{(2)}$ is the approximate expected change in the number of degree l vertices that are two-hop neighbors of a randomly selected degree k vertex, say v, when the star-sample centered at that vertex is removed, assuming each such vertex has a single connection to $\Gamma(\mathsf{v})$. In particular: two-hop neighbors of degree $l+1$ become degree l vertices and those of degree l become become degree $l-1$ vertices. Combining (11) and (12) leads to the following approximation:

$$\begin{aligned} F_{l|k} &\equiv F_{l|k}^{(1)} + F_{l|k}^{(2)} \\ &= \begin{cases} E_{l+1|k}^{(2)} - E_{l|k}^{(2)} + E_{l|k}^{(1)}, & l \neq k \\ E_{k+1|k}^{(2)} - E_{k|k}^{(2)} + E_{k|k}^{(1)} + 1, & \text{else} \end{cases} \end{aligned} \tag{13}$$

The unconditioned approximate expected change in the number of degree l vertices after removing a randomly selected star sample is:

$$F_l \equiv \sum_{k \in D} F_{l|k} w_k. \tag{14}$$

We have the following result.

Proposition 4.4. *If $m_l \geq k$ for each $k \in D$ then the approximate expected change in the number of degree l vertices from removing a* *star sample with star center of degree k is*

$$F_{l|k} = \begin{cases} kv(q_{l+1} - q_l(1-1/v)), & l \neq k, \quad l < \phi \\ kv(q_{k+1} - q_k(1-1/v)) + 1, & l = k, \quad l < \phi \\ -kvq_l(1-1/v), & l \neq k, \quad l = \phi \\ -kvq_k(1-1/v) + 1, & l = k, \quad l = \phi \end{cases} \tag{15}$$

and the (unconditioned) approximate expected change in the number of degree l vertices after removing a randomly selected star sample is:

$$F_l = \begin{cases} \mu v(q_{l+1} - q_l(1-1/v)) + w_l, & l < \phi \\ -\mu v q_l(1-1/v) + w_l, & l = \phi \end{cases} \tag{16}$$

Finally, $\sum_{l \in D} F_l \approx \mu + 1$.

The proof follows from Prop. 4.2 and Fact 6. Observe that $\sum_{l \in D} F_l$ is the approximate expected change in the number of vertices after removing a randomly selected star sample. As is intuitive, this approximation equals one plus the average vertex degree.

4.4 Temporal degree evolution under SS-S

Recall that SS-S induces a sequence of random graphs, $(\tilde{\mathsf{G}}_t, t \in \mathbb{N})$, with $\tilde{\mathsf{G}}_{t+1}$ induced from $\tilde{\mathsf{G}}_t$ (and $\tilde{\mathsf{G}}_1$ induced from the initial graph G_0) by removing the star, with the star center selected uniformly at random from $\tilde{\mathsf{V}}_t$. Define the RV $\tilde{\mathsf{w}}_{l,t} \equiv |\tilde{\mathsf{V}}_{l,t}|/|\tilde{\mathsf{V}}_t|$, i.e., the fraction of degree l vertices in $\tilde{\mathsf{V}}_t$, and the RV $\tilde{\mu}_t \equiv \sum_{k \in \tilde{D}_t} k\tilde{\mathsf{w}}_{k,t}$, i.e., the average degree in $\tilde{\mathsf{G}}_t$.

Proposition 4.5. *The approximate expected fraction of vertices of degree l in random graph $\tilde{\mathsf{G}}_t$ is, with $\tilde{F}_{l,t}$ given in Prop. 4.4,*

$$\mathbb{E}[\tilde{\mathsf{w}}_{l,t}] \approx \tilde{w}_{l,t} \equiv \frac{n_{l,0} - \sum_{t'=1}^{t-1} \tilde{F}_{l,t'}}{n_0 - \sum_{t'=1}^{t-1} \tilde{\mu}_{t'}}, \tag{17}$$

and the approximate expected vertex degree in random graph $\tilde{\mathsf{G}}_t$ is

$$\mathbb{E}[\tilde{\mu}_t] \approx \tilde{\mu}_t \equiv \sum_{k \in \tilde{D}_t} k\tilde{w}_{k,t}. \tag{18}$$

Proof. Define the sequences of RVs $((\tilde{\mathsf{n}}_{l,t}, \tilde{\mathsf{n}}_t, \tilde{\mathsf{F}}_{l,t}, \tilde{\mathsf{F}}_t), t \in \mathbb{N})$:

- $\tilde{\mathsf{n}}_{l,t} \equiv |\tilde{\mathsf{V}}_{l,t}|$ is the number of degree l vertices in $\tilde{\mathsf{G}}_t$;
- $\tilde{\mathsf{n}}_t \equiv |\tilde{\mathsf{V}}_t|$ is the number of vertices in $\tilde{\mathsf{G}}_t$;
- $\tilde{\mathsf{F}}_{l,t} \equiv |\tilde{\mathsf{V}}_{l,t-1}| - |\tilde{\mathsf{V}}_{l,t}|$ is the change in the number of degree l vertices between $\tilde{\mathsf{G}}_{t-1}$ and $\tilde{\mathsf{G}}_t$;
- $\tilde{\mathsf{F}}_t \equiv |\tilde{\mathsf{V}}_{t-1}| - |\tilde{\mathsf{V}}_t|$ is the change in the number of vertices between $\tilde{\mathsf{G}}_{t-1}$ and $\tilde{\mathsf{G}}_t$;

Observe the recurrences $\tilde{\mathsf{n}}_{l,t} = \tilde{\mathsf{n}}_{l,t-1} - \tilde{\mathsf{F}}_{l,t}$ and $\tilde{\mathsf{n}}_t = \tilde{\mathsf{n}}_{t-1} - \tilde{\mathsf{F}}_t$ with initial condition $\tilde{\mathsf{n}}_{l,0} = n_{l,0}$ and $\tilde{\mathsf{n}}_0 = n_0$ have solution

$$\tilde{\mathsf{n}}_{l,t} = n_{l,0} - \sum_{t'=1}^{t-1} \tilde{\mathsf{F}}_{l,t'}, \ \tilde{\mathsf{n}}_t = n_0 - \sum_{t'=1}^{t-1} \tilde{\mathsf{F}}_{t'} \tag{19}$$

and thus, by linearity of expectation,

$$\begin{aligned} \mathbb{E}[\tilde{\mathsf{n}}_{l,t}] &= n_{l,0} - \sum_{t'=1}^{t-1} \mathbb{E}[\tilde{\mathsf{F}}_{l,t'}] = n_{l,0} - \sum_{t'=1}^{t-1} \tilde{F}_{l,t'} \\ \mathbb{E}[\tilde{\mathsf{n}}_t] &= n_0 - \sum_{t'=1}^{t-1} \mathbb{E}[\tilde{\mathsf{F}}_{t'}] = n_0 - \sum_{t'=1}^{t-1} \tilde{F}_{t'} \end{aligned} \tag{20}$$

Approximating the expectation of a ratio as a ratio of expectations:

$$\mathbb{E}[\tilde{\mathsf{w}}_{l,t}] \equiv \mathbb{E}\left[\frac{|\tilde{\mathsf{V}}_{l,t}|}{|\tilde{\mathsf{V}}_t|}\right] = \mathbb{E}\left[\frac{\tilde{\mathsf{n}}_{l,t}}{\tilde{\mathsf{n}}_t}\right] \approx \frac{\mathbb{E}[\tilde{\mathsf{n}}_{l,t}]}{\mathbb{E}[\tilde{\mathsf{n}}_t]} = \tilde{w}_{l,t}. \tag{21}$$

□

Notice, the recurrences in (17) and (18) are well defined in that $\tilde{F}_{l,t'}$ is expressible in terms of $\tilde{F}_{l,t'-1}$ if $\mathbf{w}_0$ and n_0 are given.

4.5 Simulation validation of degree evolution

We now present simulation results to evaluate the accuracy of the various approximations used throughout this section. First, Fig. 3 shows the temporal evolution of the degree distribution under SS-S. In particular, it presents the degree distribution $\tilde{w}_{k,t}$, i.e., the approximate fraction of vertices of degree k in $\tilde{\mathsf{G}}_t$ from Prop. 4.5 along with the Monte-Carlo approximation of $\mathbb{E}[\tilde{\mathsf{w}}_{k,t}]$. The simulations are averaged over 100 independent trials on each of 10 independent Erdős-Rényi (ER) random graphs, used as the initial graphs, with parameters $n = 100$ (left) and $n = 500$ (right), and edge probability 1/20. The plots show the approximation $\tilde{w}_{k,t} \approx \mathbb{E}[\tilde{\mathsf{w}}_{k,t}]$ holds reasonably well for $k \in \{2,5,8\}$ and $t \in [30]$(left) and $t \in [55]$(right).

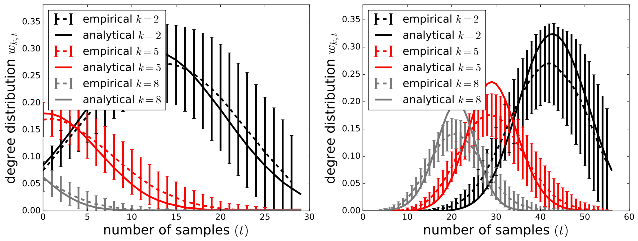

Figure 3: Temporal degree distribution evolution under SS-S.

4.6 SS-S Performance estimates

We now leverage Prop. 4.5 to derive an approximation of the SS-S performance, $\mathbb{E}[\tilde{\mathsf{N}}]$, in Def. 2.1, the main result of the paper.

Proposition 4.6. *The approximate performance of SS-S is*

$$\mathbb{E}[\tilde{\mathsf{N}}] \approx \tilde{N} \equiv \sum_t t\tilde{p}_t \prod_{t'=1}^{t-1} (1 - \tilde{p}_{t'}) \tag{22}$$

where, with F_t in Prop. 4.5,

$$\tilde{p}_t \equiv \frac{|T| + |\tilde{\Gamma}_0(T)|\left(1 - \frac{1}{n_0}\sum_{t'=1}^{t-1} \tilde{F}_{t'}\right)}{n_0 - \sum_{t'=1}^{t-1} \tilde{F}_{t'}}. \tag{23}$$

Proof. Recall T is the target set. The random graph $\tilde{\mathsf{G}}_t$ under SS-S at time t produces the following RVs: *i)* $\tilde{\Gamma}_t(T)$ are neighbors of T (not including T); *ii)* $\tilde{\Gamma}^c_t(T) = \tilde{\Gamma}_t(T) \cup T$; *iii)* $\tilde{\Gamma}_t(\tilde{\Gamma}_t(T))$ are two-hop neighbors of T; *iv)* $\tilde{\mathsf{n}}_t \equiv |\tilde{\mathsf{V}}_t|$ is the order of $\tilde{\mathsf{G}}_t$.

Observe the RV $\tilde{\mathsf{p}}_t \equiv \frac{|T| + |\tilde{\Gamma}_t(T)|}{\tilde{\mathsf{n}}_t}$ is the probability a random star sample from $\tilde{\mathsf{G}}_t$ will hit T, conditioned on the first $t-1$ samples missing T. Thus $\tilde{\mathsf{P}}_t \equiv \tilde{\mathsf{p}}_t \prod_{t'=1}^{t-1}(1 - \tilde{\mathsf{p}}_{t'})$ is the (unconditioned) probability of a first hit at sample t, and SS-S performance is:

$$\mathbb{E}[\tilde{\mathsf{N}}] = \sum_t t\mathbb{E}[\tilde{\mathsf{P}}_t]. \tag{24}$$

Defining $\tilde{p}_t \equiv \mathbb{E}[\tilde{\mathsf{p}}_t]$ and $\tilde{P}_t \equiv \mathbb{E}[\tilde{\mathsf{P}}_t]$, and approximating the RVs $(\tilde{\mathsf{p}}_t, t \in \mathbb{N})$ as independent, we obtain the approximation in (22).

It remains to approximate $\tilde{p}_t$. Approximate the expectation of the ratio as the ratio of expectations and leverage Prop. 4.5:

$$\begin{aligned} \tilde{p}_t \quad &\equiv \quad \mathbb{E}[\tilde{\mathsf{p}}_t] = \mathbb{E}\left[\frac{|T| + |\tilde{\Gamma}_t(T)|}{\tilde{\mathsf{n}}_t}\right] \approx \frac{|T| + \mathbb{E}[|\tilde{\Gamma}_t(T)|]}{\mathbb{E}[\tilde{\mathsf{n}}_t]} \\ &\approx \quad \frac{|T| + \mathbb{E}[|\tilde{\Gamma}_t(T)|]}{n_0 - \sum_{t'=1}^{t-1} \tilde{F}_{t'}}. \end{aligned} \tag{25}$$

We approximate $\mathbb{E}[|\tilde{\Gamma}_t(T)|]$ also using Prop. 4.5, thinning $\tilde{F}_{t'}$ by the ratio $|\tilde{\Gamma}_0(T)|/n_0$:

$$\mathbb{E}[|\tilde{\Gamma}_t(T)|] \approx |\tilde{\Gamma}_0(T)| \left(1 - \frac{1}{n_0} \sum_{t'=1}^{t-1} \tilde{F}_{t'}\right). \tag{26}$$

□

Assuming $m_l \geq k$ for each pair $(k, l) \in D$, Alg. 1 recursively calculates $\tilde{F}_{t,l}$ given a graph of order n. Let $\hat{\mathbf{w}}_{0,l} = Bin(k; n-1, s)$ for $l \in [0, \theta]$ be the expected probability at $t = 0$ of selecting a degree l node and $\hat{\mathbf{n}}_{0,l} = n\hat{\mathbf{w}}_{0,l}$ be the expected number of degree l nodes. Compute $\mu_0 = \sum_{l=0}^{\phi} l\hat{\mathbf{w}}_{0,l}$ the expected average degree, $c_0 = \sum_{l=0}^{\phi} l^2\hat{\mathbf{n}}_{0,l}$ the numerator of $\nu_0 = \frac{c_0}{n_0\mu_0}$ the expected average stub degree, and let $\hat{\mathbf{q}}_{0,l} = \frac{l n_{0,l}}{n_0, \mu_0}$ for $l \in [0, \phi]$. If $m_l \ngeq k$ for some pair $(k, l) \in D$ a similar recursive algorithm can compute $\tilde{F}_{l,t}$ using equation (14). For $n \gg \phi$ the complexity of Alg. 1 is $O(2n\phi)$.

Algorithm 1 Calculate $\tilde{F}_{t,l}$ for all t while $\tilde{n}_t > \phi + 1$

```
**require:** $\hat{\mathbf{F}} = (\tilde{F}_{t,l}) \in \mathbb{R}^{n \times n}$, $\hat{\mathbf{w}}_0$, $\hat{\mathbf{n}}_0$, $n$, $\phi$, $\mu_0$, $c_0$, $\tilde{n}_0 = n$, $t = 0$
**while** $\tilde{n}_t > \phi + 1$ **do**
    $\mu_t = 0$, $c_t = 0$, $\hat{\mathbf{w}}_t \in \mathbb{R}^n$, $\hat{\mathbf{n}}_t \in \mathbb{R}^n$, $\hat{\mathbf{q}}_t \in \mathbb{R}^n$        ▷ init. val.
    **if** $t \neq 0$ **then**
        $\tilde{n}_t = \tilde{n}_{t-1} - (\mu_{t-1} + 1)$        ▷ update no. of nodes
        **for** $l \in [0, \phi]$ **do**
            $\hat{\mathbf{n}}_{t,l} = \hat{\mathbf{n}}_{t-1,l} - \hat{\mathbf{F}}_{t-1,l}$        ▷ update no. of deg. $l$ nodes
            $\hat{\mathbf{w}}_{t,l} = \frac{\hat{\mathbf{n}}_{t,l}}{\tilde{n}_t}$        ▷ update deg. $l$ node prob. dist.
            $\hat{\mathbf{q}}_{t,l} = \frac{k\hat{\mathbf{n}}_{t,l}}{\tilde{n}_t\mu_t}$        ▷ update deg. $l$ stub prob. dist.
            $\mu_t = \mu_t + l\hat{\mathbf{w}}_{t,l}$        ▷ update ave. node deg.
            $c_t = c_t + l^2\hat{\mathbf{n}}_{t,l}$        ▷ update num. of ave. stub deg.
    $\nu_t = \frac{c_t}{\tilde{n}_t\mu_t}$        ▷ calc. ave. stub deg.
    **for** $l \in [0, \phi]$ **do**
        $\tilde{F}_{t,l} = F_l(\mu_t, \nu_t, \hat{\mathbf{w}}_{t,l}, \hat{\mathbf{q}}_{t,l})$, def. in eqn. (16)        ▷ calc. $\tilde{F}_{t,l}$
    $t = t + 1$
**return F**
```

Fig. 4 compares the empirical and analytical parameters computed recursively for an ER graph under the assumption $m_l \ngeq k$ for some pair $(l, k) \in D$ with $n = 500$, $s = 0.02$, and a unique max degree node v^* where $d_{v^*} = \phi$, over 500 sampling trials of $SS - R$, $SS - C$, and $SS - S$. The left figure shows n_t, the number of nodes in the graph at sample t. The middle figure shows the number of nodes that are in set $\Gamma^c(T)$, set T and the nodes neighboring T, conditioned on T being unsampled. The right figure shows p_t the probability of sampling a node in set $\Gamma^c(T)$, conditioned on T being unsampled.

The parameter estimates for n_t, $|\Gamma^c(T)|$, and p_t are accurate for $SS - R$ and $SS - C$. However under $SS - S$ the estimates for n_t and $|\Gamma^c(T)|$ diverge for t large and $n_t \rightarrow \phi + 1$, see the left and middle figures of Fig. 4. These divergences result in an underestimate of $\tilde{p}_t$ for large t, see right figure of Fig. 4. Yet since $\tilde{\mathsf{P}}_t \approx \tilde{p}_t \prod_{t'=1}^{t-1}(1 - \tilde{p}_{t'})$ the weight of terms $\tilde{p}_t$ decrease as t becomes large and as the underestimate in $\tilde{p}_t$ occurs for only $t \gg \mathbb{E}[N] \geq \mathbb{E}[\tilde{N}]$, the estimate of $\mathbb{E}[\tilde{N}]$ remains fairly accurate, see §5.

5 NUMERICAL AND SIMULATION RESULTS

To evaluate the performance of the three star sampling variants we present approximate and Monte-Carlo performance estimates for initial graphs that are: *i*) "synthetic" Erdős-Rényi (ER) random graphs, and *ii*) three different "real-world" graphs. In all cases we set the target set to be the set of maximum degree vertices.

Synthetic ER graphs. Fig. 5 shows approximate (Prop. 4.6) and Monte-Carlo performance estimates for the three SS variants on 5 distinct ER initial graphs, 100 independent trials on each graph, with ER edge probability 1/50 (left) and $5/n$ (right).

Real-world graphs. Fig. 6 shows approximate (Prop. 4.6) and Monte-Carlo performance estimates for the three SS variants on three different "real-world" graphs: *i*) the High Energy Physics Theory (HepTh) collaboration graph from SNAP [22], *ii*) the General Relativity and Quantum Cosmology (GrQC) collaboration graph from SNAP [22], and *iii*) the Western States Power Grid of the United States (Power) [34]. In all three cases 1000 independent simulations were run. The properties of the three graphs are listed below, where n is order, m is size, μ is average degree, ϕ is the maximum degree, and α is assortativity.

Graph	n	m	μ	α	ϕ	$n/(\phi+1)$
HepTh	9,877	25,973	5.26	0.268	65	≈ 152
Power	4,941	6,594	2.67	0.003	19	≈ 247
GrQc	5,242	14,484	5.5	0.659	81	≈ 64

Table 1: Statistics of the three "real-world" graphs.

From Fact 1, we may approximate $\mathbb{E}[\mathrm{N}]$ as $n/|\Gamma^c(T)|$, see Table 1. Recall T is the set of maximum degree vertices. If we assume that there is a unique such vertex (c.f., [28]), then $|\Gamma^c(T)| = \phi + 1$. This approximation is reasonably close for all three graphs in Fig. 6:

The findings in both the synthetic ER and the three "real-world" graphs are the same: *i*) there is little to no difference across the three star sampling variants, and *ii*) the analytical approximations are seen to be very accurate. Thus, the simple expression $n/|\Gamma^c(T)|$ appears to be a suitable approximation for the number of samples required under all three star-sampling variants.

6 RELATED WORK

Star sampling is presented as a special case of the more general concept of *snowball sampling* in [17]. *Snowball sampling* was introduced by Goodman [13] and studied by Frank [10]. *Snowball sampling* appears in [20], [2], [14]. *Star sampling* is a snowball sample where a sample consists of a center vertex $\mathrm{v} \in V$ and its immediate neighbors $\Gamma(\mathrm{v})$; *Star sampling* appears in [33].

This paper is an extension of our prior work [30], which also focused on estimating the number of star samples required to find

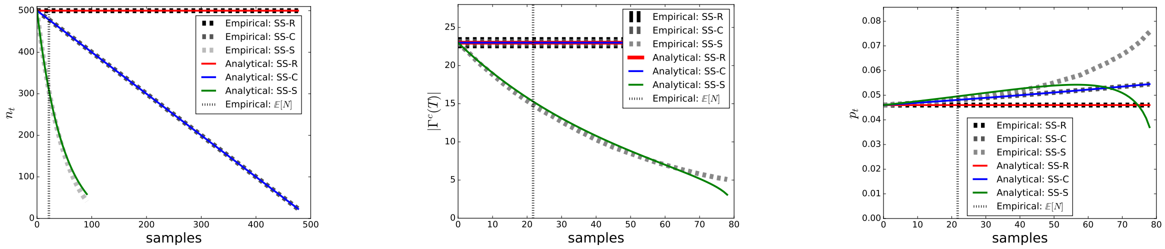

Figure 4: Assuming set T is unsampled, Left: n_t, Middle: $|\Gamma^c(T)|$, Right: p_t

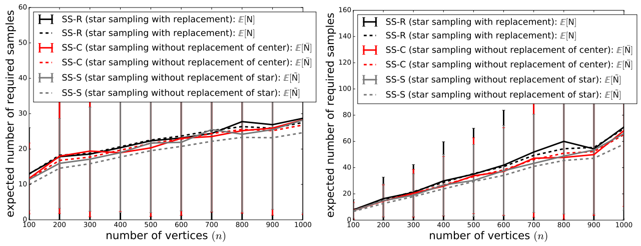

Figure 5: Approximate and simulation star sampling performance results for the three variants vs. the graph order n.

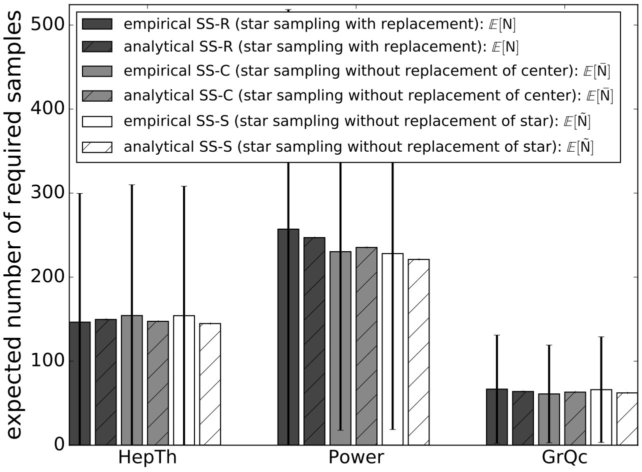

Figure 6: Approximate and simulation star sampling performance results for the three "real-world" graphs.

a target vertex. In [30] we considered the slightly more general problem of finding a degree k node or an edge with a degree l endpoint. Moreover, we attempted to analyze the performance of *star sampling with replacement* on a modified Erdős-Rényi (ER) random graph construction, such that the estimator in Def. 4.1 is exact. However, none of the analysis in this paper is present in [30].

There has been a substantial amount of work on the problem of sampling graphs. Classic graph exploration strategies include: *random sampling* of vertices or edges, *random walk sampling*, and *random jump sampling*, which alternates between a *random walk* and *random sampling*. These graph exploration strategies are general in the sense that they perform reasonably well in a broad range of problems. However, the graph sampling literature itself is divided between work *i*) attempting to derive an unbiased or uniform estimate of the vertices in a network, and *ii*) attempting to find vertices with particular properties, for instance maximum degree vertices. We refer to the former problem as the *graph sampling problem* and the latter as the *graph search problem*.

The graph sampling problem became widespread with the advent of social media. In particular, one important question it addresses is how to obtain a representative sample of social media users. To solve this problem Leskovac introduced *forest fire sampling* for temporal graphs where, similar to *breadth-first search* (BFS), a search frontier is established. However, instead of expanding this frontier to unexplored vertices as in BFS, each iteration there is a chance of the frontier retreating to re-examine previously explored vertices [21]. Riberio proposed and analyzed a related algorithm entitled *frontier sampling* [26]. Avrachenkov [5] and Jin [16] have both proposed *random walk jump* algorithms to obtain an unbiased sample of vertices and Avin [3] has proposed a *random walk* biased toward high degree unvisited vertices. Miaya has looked at the sampling bias showing that *expansion sampling* can be more effective in exploring graphs than degree *biased random walks* [25]. Although subsequently Voudigari has proposed a *degree biased breadth-first search* algorithm for the graph sampling problem [32].

More sophisticated algorithms for solving the graph sampling problem include *Metropolized random walk with backtracking*, proposed by Stutzbach [31]. However Lee has argued that Metropolis-Hastings sampling algorithms should avoid backtracking [19]. Li proposed a *Rejection controlled Metropolis-Hastings* algorithm and a *Non-backtracking generalized maximum-degree sampling* algorithm [23]. Gjorka found that *Metropolis-Hasting random walk*'s and *Re-weighted random walk*'s both out perform a simple random walk in returning a uniform sample of Facebook users [11]. While Kurant [18] has shown that *weighted random walks* can be used to carry out stratified sampling on graphs, and Chierichetti [7] gives bounds on the number of steps required to return a uniform sample of a network using *rejection sampling*, *maximum-degree sampling*, and *Metropolis-Hastings sampling*.

The performance of a *random walk* in solving a graph search problems depends on its performance in the graph cover problem, the time it takes a random walk to visit every node $v \in V$, or every node $v \in T$ for $T \subset V$. This problem gained prominence with P2P networks where the question was how to design P2P networks and search algorithms which allowed users to locate files efficiently. Ikeda has shown that given any undirected connected graph G of order n the cover time and mean hitting time of a *degree biased* *random walk* is bounded by $O(n^2 \log n)$ and $O(n^2)$ respectively [15]. Cooper has shown in sparse Erdős Rényi graphs $G(n, s)$ the cover time of a *random walk* is asymptotically $cn \log \frac{c}{c-1} \log n$ where $s = \frac{c \log n}{n}$ and $c > 1$ [8]. Cooper also shows that in power-law graphs of order n with parameter $c \geq 3$, finding all vertices of degree n^a, or greater with a *degree biased random walk* for $0 \leq a \leq 1$ and bias coefficient $b > 0$ is $\tilde{O}(n^{1-2ab(1-\epsilon)})$ with high probability [9].

Cooper's results match Adamic's observation that the search time of *random walks* and *degree biased random walks* scale sublinearly with the size of power-law graphs [1]. Lv [24] has also shown that for the graph search problem, random walks outperform network flooding in P2P networks; Gkantsidis [12] expanded on this work and Brautbar [6] and Avrachenkov [4] have both shown that *random walk jump* algorithms are effective in finding the high degree nodes. Our prior work [29] proposed a *self avoiding degree biased random walk jump* algorithm called SAWJ.

Random walks however are not the only approach to searching a graph for vertices with particular properties. Avrachenkov has introduced the *Two-stage algorithm* for finding high degree vertices developed under the assumption that queries of the sampled graph are limited [4]. Our own work on finding maximum degree vertices has assumed that queries of the sampled graph are not a limiting factor. Given this assumption we have shown that *biased random walks* and *star sampling* can both be effective in finding vertices of interest, c.f. our earlier work [27], [28].

7 CONCLUSION

Star sampling is a natural graph sampling paradigm, and as such it is important to optimize its design. In this paper we study three star sampling variants, involving various types of replacement, motivated by analogous sampling strategies of balls from an urn. Our analytical and simulation results demonstrate that *i*) our mathematical approximations lead to reasonably accurate performance estimators, and *ii*) there is, perhaps surprisingly, no significant difference between the three variants. Our intuitive explanation for this is that star sampling without star replacement "helps" by reducing the number of vertices outside the target set, but "hurts" by reducing, on average, the number of neighbors of the target set. This target set neighbor reduction makes it harder for a star sample to "hit" a vertex in the target set. Our future work will focus on more rigorous mathematical justifications for the various approximations employed in deriving our estimators.

REFERENCES

[1] Lada A. Adamic, Rajan M. Lukose, Amit R. Puniyani, and Bernardo A. Huberman. 2001. Search in power-law networks. *Phys. Rev. E* 64 (2001), 046135. Issue 4.

[2] Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue Moon, and Hawoong Jeong. 2007. Analysis of Topological Characteristics of Huge Online Social Networking Services. In *Proceedings of the 16th International Conference on World Wide Web (WWW '07)*. 835–844.

[3] C. Avin and B. Krishnamachari. 2008. The power of choice in random walks: An empirical study. *Computer Networks* 52, 1 (2008), 44 – 60.

[4] Konstantin Avrachenkov, Nelly Litvak, Marina Sokol, and Don Towsley. 2012. *Quick Detection of Nodes with Large Degrees*. Springer Berlin Heidelberg, 54–65.

[5] Konstantin Avrachenkov, Bruno Ribeiro, and Don Towsley. 2010. Improving random walk estimation accuracy with uniform restarts. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 98–109.

[6] Mickey Brautbar and Michael Kearns. 2010. Local Algorithms for Finding Interesting Individuals in Large Networks. In *ICS*.

[7] Flavio Chiericetti, Anirban Dasgupta, Ravi Kumar, Silvio Lattanzi, and Tamás Sarlós. 2016. On Sampling Nodes in a Network. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, 471–481.

[8] Colin Cooper and Alan Frieze. 2007. The cover time of sparse random graphs. *Random Structures & Algorithms* 30, 1-2 (2007), 1–16.

[9] Colin Cooper, Tomasz Radzik, and Yiannis Siantos. 2014. A Fast Algorithm to Find All High-Degree Vertices in Graphs with a Power-Law Degree Sequence. *Internet Mathematics* 10, 1-2 (2014), 137–161.

[10] O. Frank. 1977. Survey sampling in graphs. *Journal of Statistical Planning and Inference* 1, 3 (1977), 235 – 264.

[11] Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. 2010. Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In *Proceedings of the 29th Conference on Information Communications (INFOCOM'10)*. 2498–2506.

[12] Christos Gkantsidis, Milena Mihail, and Amin Saberi. 2006. Random walks in peer-to-peer networks: Algorithms and evaluation. *Performance Evaluation* 63, 3 (2006), 241 – 263.

[13] L. Goodman. 1961. Snowball Sampling. *Ann. Math. Statist.* 32, 1 (1961), 148–170.

[14] Pili Hu and Wing Cheong Lau. 2013. A survey and taxonomy of graph sampling. *arXiv preprint arXiv:1308.5865* (2013).

[15] S. Ikeda and I. Kubo. 2003. Impact of Local Topological Information on Random Walks on Finite Graphs. In *Proc. of the 30th Intl. Conf. on Automata, Languages and Programming*. 1054–1067.

[16] L. Jin and et al. Chen, Y. 2011. Albatross Sampling: Robust and Effective Hybrid Vertex Sampling for Social Graphs. In *Proceedings of the 3rd ACM International Workshop on MobiArch (HotPlanet '11)*. 11–16.

[17] Eric D. Kolaczyk. 2009. *Statistical analysis of network data : methods and models*. Springer, New York" London.

[18] Maciej Kurant, Minas Gjoka, Carter T. Butts, and Athina Markopoulou. 2011. Walking on a Graph with a Magnifying Glass: Stratified Sampling via Weighted Random Walks. In *Proc. of the ACM SIGMETRICS Joint Intl. Conf. on Measurement and Modeling of Computer Systems*. 281–292.

[19] Chul-Ho Lee, Xin Xu, and Do Young Eun. 2012. Beyond Random Walk and Metropolis-hastings Samplers: Why You Should Not Backtrack for Unbiased Graph Sampling. *SIGMETRICS Perform. Eval. Rev.* 40, 1 (June 2012), 319–330.

[20] Sang Hoon Lee, Pan-Jun Kim, and Hawoong Jeong. 2006. Statistical properties of sampled networks. *Phys. Rev. E* 73 (2006), 016102. Issue 1.

[21] 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 (KDD '06)*. 631–636.

[22] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).

[23] R. H. Li, J. X. Yu, L. Qin, R. Mao, and T. Jin. 2015. On random walk based graph sampling. In *2015 IEEE 31st International Conference on Data Engineering*. 927–938.

[24] Qin Lv, Pei Cao, Edith Cohen, Kai Li, and Scott Shenker. 2002. Search and Replication in Unstructured Peer-to-peer Networks. In *Proceedings of the 16th International Conference on Supercomputing (ICS '02)*. 84–95.

[25] A.S. Maiya and T.Y. Berger-Wolf. 2011. Benefits of Bias: Towards Better Characterization of Network Sampling. In *Proc. of the 17th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*. 105–113.

[26] B. Ribeiro and D. Towsley. 2010. Estimating and Sampling Graphs with Multidimensional Random Walks. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC '10)*. 390–403.

[27] J. Stokes and S. Weber. 2016. A Markov chain model for the search time for max degree nodes in a graph using a biased random walk. In *Information Sciences and Systems (CISS)*.

[28] J. Stokes and S. Weber. 2016. On random walks and random sampling to find max degree nodes in assortative Erdős Rényi graphs. In *2015 IEEE Global Communications Conference (GLOBECOM)*.

[29] J. Stokes and S. Weber. 2016. The self-avoiding walk-jump (SAWJ) algorithm for finding maximum degree nodes in large graphs. In *2016 IEEE International Conference on Big Data (Big Data)*.

[30] J. Stokes and S. Weber. 2017. On the number of star samples to find a vertex or edge with given degree in a graph. In *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. 1–6.

[31] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. 2009. On Unbiased Sampling for Unstructured Peer-to-peer Networks. *IEEE/ACM Trans. Netw.* 17, 2 (2009), 377–390.

[32] E. Voudigari, N. Salamanos, T. Papageorgiou, and E. J. Yannakoudakis. 2016. Rank degree: An efficient algorithm for graph sampling. In *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. 120–129.

[33] H. Wang and J. Lu. 2013. Detect inflated follower numbers in OSN using star sampling. In *2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2013)*. 127–133.

[34] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of small-world networks. *Nature* 393, 6684 (1998), 440–442.