2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

MaxReach: Reducing Network Incompleteness through Node Probes

Sucheta Soundarajan[♢] Tina Eliassi-Rad[†] Brian Gallagher[‡] Ali Pinar[§]
[♢]Syracuse University [†]Northeastern University [‡]Lawrence Livermore Nat'l Laboratory [§]Sandia Nat'l Laboratories
[♢]susounda@syr.edu [†]eliassi@ccs.neu.edu [‡]bgallagher@llnl.gov [§]apinar@sandia.gov

***Abstract*—Real-world network datasets are often incomplete. Subsequently, any analysis on such networks is likely to produce skewed results. We examine the following problem: given an incomplete network, which b nodes should be probed to bring as many new nodes as possible into the observed network? For instance, consider someone who has observed a portion (say 1%) of the Twitter network. How should she use a limited budget to reduce the incompleteness of the network? In this work, we propose a novel algorithm, called MAXREACH, which uses a budget b to increase the number of nodes in the observed network. Our experiments, across a range of datasets and conditions, demonstrate the efficacy of MAXREACH.**

I. INTRODUCTION

Suppose that one has observed $\tilde{G}$, an incomplete portion of some larger network G. To learn more about the structure of G, one can *probe* nodes from $\tilde{G}$, revealing more information about the selected nodes. Which nodes in $\tilde{G}$ should be probed if the goal is to observe as many nodes as possible in G? Unlike much of graph sampling work, we are not generating a sample from scratch, but are improving existing samples. Our work is motivated by problems where one has partially observed the complete network; but needs an accurate global picture. For example, suppose that one has obtained a sample of the Twitter network from another researcher. How should one best supplement/enrich this sampled data?

We present MAXREACH, a novel algorithm for selecting which nodes from a partially observed network should be probed in order to observe as many nodes as possible from the underlying network. MAXREACH estimates each node's true degree in G as well as the number of nodes to which it is connected in $\tilde{G}$, then selects nodes to probe.

Across a variety of networks and probing scenarios, MAXREACH consistently outperforms comparison strategies. Figure 1 depicts the results of MAXREACH and three alternative strategies on a uniformly random edge sample from the Enron e-mail network. Over a large range of probing budgets, MAXREACH dramatically outperforms the other strategies.

We make the following contributions:

- We present MAXREACH, a novel algorithm for selecting nodes from an incomplete network to probe in order to maximize the number of observed nodes.
- Our experiments demonstrate that MAXREACH outperforms the best baseline by a large margin (e.g., Figure 1).

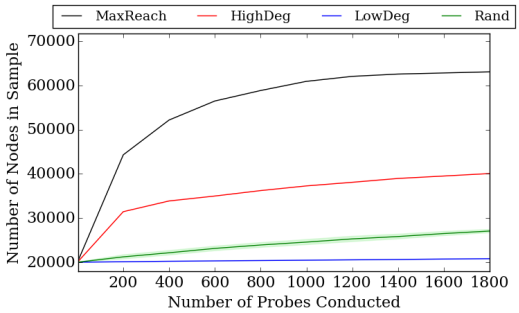

Fig. 1. MAXREACH results on Enron e-mail network. $\tilde{G}$ was generated by a 10% random edge sample. Here, probing a node returns all of its edges in G. MAXREACH outperforms the baselines by large margins. Shading indicates one standard deviation. Similar results were observed on other networks.

II. PROBLEM DEFINITION

We are given an incomplete, partially observed graph $\tilde{G}$ that is a part of a larger, fully observed graph G. Our goal is to observe as many nodes as possible in G. We are given a *probing budget*, which we use to *probe* observed nodes and gain more information about those nodes (discussed below).

A. Probing Scenarios

We consider a broad variety of probing scenarios. We assume that the degree of a node is not known in advance. Although many APIs can return the degree of a node, we intend for our techniques to be used even when one does not have access to such information. For example, one may be obtaining data without use of an API, such as by mailing surveys, observing web traffic through machines, etc. If the degree of the node is known, then it can easily be incorporated into our method.

Suppose that node u is selected for probing. We consider the following probing scenarios, which determine the information returned when the probe is conducted:

All-Neighbor Probing: All of node u's edges are returned. For example, the Facebook Graph API allows one to get a list of all of a user's friends (with appropriate permissions and subject to privacy settings) with one request.

k-Neighbor Probing, No Replacement (k-NR): k edges adjacent to u are returned (for a fixed k). These edges are selected uniformly at random from all unobserved edges of u. For example, the Twitter API allows for one to request a user's followers, and returns $5,000$ results at a time.

Symbol	Meaning
G	Underlying network, not fully observed
$\tilde{G}$	Partially bserved network, initially and while probing
N	Number of nodes in G
p	Fraction of edges or nodes sampled to produce $\tilde{G}$
q_i, $\hat{q_i}$	Probability that a randomly selected node from G has degree equal to i in G, MAXREACH's estimate of q_i
$C(d)$	Mean clustering coefficient in G for nodes of degree d
d_u, $\hat{d}_u$	Degree of node u in G, MAXREACH's estimate of d_u
$\tilde{d}_u$	Observed degree of node in $\tilde{G}$

TABLE I
Notation used in describing MAXREACH.

k-Neighbor Probing, With Replacement (k-WR): The k-WR probing scenario is like the k-NR scenario, but edge selection is done with replacement; that is, the k returned edges may contain duplicates or previously observed edges. For example, the Twitter API returns 100 retweets at a time; however, multiple retweets can correspond to the same network edge.

Connection Charge Probing, No Replacement (Conn-NR): For each probe, one must pay a cost c to initiate a probe as well as a cost r per edge requested, and the user must specify k, the number of edges requested, for a total cost of $c + rk$.

Connection Charge Probing, With Replacement (Conn-WR): Conn-WR is identical to Conn-NR, except that the returned edges are selected uniformly at random with replacement.

B. Generation of $\tilde{G}$ and Assumptions

We assume that the original creators of $\tilde{G}$ produced it either by sampling p fraction of the edges from G uniformly at random, or by sampling p fraction of the nodes from G uniformly at random. In the latter case, when a node is sampled, it and all of its neighbors are included.

We assume that we know the number of nodes and edges in G, the sampling process that was used to generate $\tilde{G}$, and, if $\tilde{G}$ was generated by a Random Node sample, the identities of the nodes that were sampled during the sampling process.

III. PROPOSED METHOD: MAXREACH

We present MAXREACH, a novel algorithm for selecting which nodes from a partially observed network $\tilde{G}$ to probe in order to maximize the total number of nodes observed.

For each node u in $\tilde{G}$, MAXREACH estimates node u's true degree d_u and the number of neighbors d_u^{out} that it has outside $\tilde{G}$ (Section IV-C). Using these values, MAXREACH assigns a probing scenario dependent score to node u (Section V-A). For example, under All-Neighbor Probing, MAXREACH assigns each node u a score of d_u^{out}. Figure 2 presents an overview of MAXREACH, and Table I contains our notation.

MAXREACH contains the following steps:

- The *Setup Stage* infers global network characteristics, as well as each node u's true degree d_u in G and the number of neighbors that u has in $\tilde{G}$. Using these two values, MAXREACH assigns each node a score. See Section IV.
- In the *Probing Stage*, MAXREACH selects nodes for probing using their scores, obtains new information about those nodes, and updates their scores. See Section V.

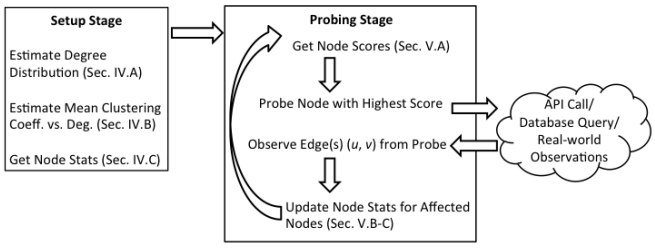

Fig. 2. Overview of MAXREACH. The Setup Stage estimates global network properties. Next, probing is conducted in an iterative manner. Node statistics are updated as new information is obtained.

IV. MAXREACH SETUP STAGE: INFERRING GRAPH STATISTICS

MAXREACH uses the observed network $\tilde{G}$ to estimate the degree distribution of the underlying network G, as well as the relationship between degree and mean clustering coefficient.[1]

A. Estimating Degree Distribution

To estimate the degree distribution of the underlying network G, MAXREACH uses (1) the observed node degrees in $\tilde{G}$, (2) whether $\tilde{G}$ was generated by random node sampling or random edge sampling, (3) the number of nodes and edges in G, and (4) the fraction p and identities of nodes or edges sampled from G to produce $\tilde{G}$ (as described in Section II-B).

Random Edge Sample. Suppose that $\tilde{G}$ was generated by sampling p fraction of edges uniformly at random from the edges in G. Then estimating the underlying degree distribution of G reduces to solving the following problem:

$$\begin{bmatrix} B(0,1) & B(0,2) & ... & B(0,D) \\ B(1,1) & B(1,2) & ... & B(1,D) \\ B(2,1) & B(2,2) & ... & B(2,D) \\ & & ... & \\ B(\tilde{D},1) & B(\tilde{D},2) & ... & B(\tilde{D},D) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ ... \\ c_D \end{bmatrix} = \begin{bmatrix} \tilde{c_0} \\ \tilde{c_1} \\ ... \\ \tilde{c_{\tilde{D}}} \end{bmatrix}, \quad (1)$$

where $B(i,j)$ is the probability that a node with degree j in G has degree i in $\tilde{G}$. These values come from the binomial distribution and are calculated as: $B(i,j) = \binom{j}{i} p^i (1-p)^{j-i}$.

D and $\tilde{D}$ represent the maximum node degrees in G and $\tilde{G}$. D is not known, so MAXREACH sets $D = 2\frac{\tilde{D}}{p}$ to be a reasonable upper bound on this degree. c_i and $\tilde{c_j}$ represent, respectively, the number of nodes with degree i in G and the number of nodes with degree j in $\tilde{G}$. Because we know the number of nodes in G, we set $\tilde{c_0}$ to be the number of nodes not present in $\tilde{G}$. (Assuming all nodes have degree at least 1 in G; if not, the expression above can be trivially modified.)

We wish to estimate the c_i values, but this problem is underdetermined. One can instead solve a constrained least-squares problem using convex optimization, such as is described in [13], but this can be slow. Instead, MAXREACH uses an iterative estimation procedure to estimate the c_i values.

[1] See Section II-A for a discussion of why we do not assume that the degree of each node can be obtained through the API.

MAXREACH first sets the probability that a randomly selected node has true degree i in G as $\hat{q}_i = \frac{1}{D}$ (that is, the initial degree distribution is set to be the uniform distribution). Next, MAXREACH performs several iterations.[2] In each iteration, for each node u in $\tilde{G}$ and for each possible true degree i, the probability that node u has true degree i is calculated as:

$$Pr(d_u = i) = \frac{\hat{q}_i B(i, \tilde{d}_u)}{\sum_j \hat{q}_j B(j, \tilde{d}_u)} \quad (2)$$

Then by summing over all nodes in $\tilde{G}$ (recall that we assume that MAXREACH knows the number of nodes in G, so this sum includes those nodes which are not directly observed, and thus have degree 0 in $\tilde{G}$), and dividing by the total number of nodes, MAXREACH updates $\hat{q}_i$, the estimated probability that a randomly selected node from G has true degree i. These probabilities are then used in the next iteration.

Through this process, MAXREACH estimates the degree distribution of the underlying network.

MAXREACH obtains high-quality degree estimates, as measured by the K-L divergence between the estimated and true degree distributions. To compare, we estimate degrees naively by multiplying the observed degree of each node by $\frac{1}{p}$, which is the expected value of the node's true degree. MAXREACH obtains a mean K-L divergence of 0.2 (stdev of 0.16) from the true distribution, 24x-430x better than the naive method.

Random Node Sample. A random node sample contains two types of nodes: those selected during sampling, and their neighbors. If MAXREACH used only the selected nodes, then the tail end of the distribution would be ignored, because networks tend to have few high degree nodes.

MAXREACH instead focuses on the nodes that were *not* sampled. For such a node, its expected observed degree matches the case when p fraction of the edges were sampled (as in a Random Edge sample). By considering only these nodes, MAXREACH performs a process identical to the above.

B. Estimating Mean Clustering Coefficient vs. Degree

MAXREACH estimates the relationship between clustering coefficient and degree (this relationship exists as shown in [11]). The clustering coefficient of a node is the fraction of its neighbor-pairs that are connected; i.e., for a node u, it is the number of triangles (u, v, w) that u participates in divided by the number of wedges $(u, v), (u, w)$ centered on u.

MAXREACH estimates $C(d)$, the mean clustering coefficient for nodes with degree d in G. To do this, for each node u in $\tilde{G}$, MAXREACH estimates its true degree d_u in G using the procedure above. Next, MAXREACH estimates that node's clustering coefficient in G, as below.

Random Edge Sample. Suppose p fraction of edges from G were sampled to produce $\tilde{G}$. For a node u in G, for each wedge $(v, u), (u, w)$ centered on u, there is a p^2 probability that the wedge is preserved in $\tilde{G}$. For each triangle (u, v, w), there is a p^3 probability that the triangle is preserved in $\tilde{G}$. Thus, to estimate u's clustering coefficient in G, we divide its observed clustering coefficient by p.

By estimating each node's true degree (as above) and performing this calculation, MAXREACH obtains $C(d)$, the mean clustering coefficient for each estimated true degree d.

Note that MAXREACH can only estimate $C(d)$ for values d such that some node in $\tilde{G}$ actually has estimated true degree d. For other values d', we estimate $C(d')$ as being equal to $C(d)$ for the value of d closest to d' on which C is defined.

Random Node Sample. To estimate the $C(d)$ values when $\tilde{G}$ was produced by random node sampling, MAXREACH groups the nodes into two sets: those selected during sampling (whose neighborhoods are known), and their neighbors.

First consider nodes in the first category. To calculate the clustering coefficient for such a node u, MAXREACH keeps a wedge count $\tilde{W}$ and a triangle count $\tilde{T}$. MAXREACH iterates over all neighbors v of u such that v was also selected during sampling. In each iteration, MAXREACH increments $\tilde{W}$ by $d_u - 1$, which is the number of wedges (v, u, w) that v participates in with u, and increments $\tilde{T}$ by the number of triangles (v, u, w) that v participates in with u. $\tilde{T}/\tilde{W}$ gives an unbiased estimate of node u's true clustering coefficient.

For nodes in the second category, MAXREACH uses a process similar to the case of Random Edge sampling. For a node u in this category, a triangle (u, v, w) from G survives in $\tilde{G}$ only if v and w were both selected during sampling. Similarly, wedge (v, u, w) from G is present in $\tilde{G}$ only if v and w were both sampled. This occurs with probability p^2, so the clustering coefficient of node u in $\tilde{G}$ is an unbiased estimate of its clustering coefficient in G.

By estimating the true clustering coefficients of each node in $\tilde{G}$, and by estimating each node's true degree in G, MAXREACH is able to estimate the $C(d)$ values.

C. Node Statistics

The score of a node depends on the particular probing scenario being considered, and uses the following values:

- $\hat{d}_u$ is u's estimated true degree in G.
- d^{out} is the estimated number of u's neighbors outside $\tilde{G}$.
- d_u^{in} is the estimated number of nodes in $\tilde{G}$ that u is adjacent to in G. This includes:
  - d_u^{known}, the number of nodes in $\tilde{G}$ that we already know to be adjacent to u.
  - $d_u^{unknown}$, the estimated number of nodes in $\tilde{G}$ that u is connected to in G, but not in $\tilde{G}$ (i.e., the connections to u that have not been observed).

MAXREACH estimates d_u^{out} as:

$$d_u^{out} = \hat{d}_u - d_u^{in} = \hat{d}_u - d_u^{known} - d_u^{unknown}. \quad (3)$$

Using the estimated degree distribution, MAXREACH obtains estimates the degree $\hat{d}_u$ for each node u through application of Bayes' Theorem. Suppose node u in $\tilde{G}$ has observed degree $\tilde{d}_u$. Then the expected true degree $\hat{d}_u$ of u in G is:

$$\hat{d}_u = E[d_u] = \frac{\sum_i i\hat{q}_i B(i, \tilde{d}_u)}{\sum_j \hat{q}_j B(j, \tilde{d}_u)}, \quad (4)$$

[2] In our experiments, five iterations proved sufficient.

where the sum is over all possible degrees i, and $\hat{q}_i$ is the estimated probability that a random node has degree i in G.

d_u^{known} is equal to $\tilde{d}_u$. $d_u^{unknown}$ is estimated using the estimated clustering coefficients of its known neighbors. Suppose that node u has neighborhood N_u in $\tilde{G}$. Each of u's neighbors v has an estimated true degree $\hat{d}_v$ and estimated clustering coefficient $C(d_v)$. Let N_v be the neighborhood of v in $\tilde{G}$. MAXREACH uses the clustering coefficient of node v to estimate the number of v's neighbors to which u is connected.

Random Edge Sample. Suppose that $(u,v),(v,w) \in \tilde{G}$, and $(u,w) \notin \tilde{G}$. Then MAXREACH estimates the probability that u and w are neighbors in G, given the estimated clustering coefficient $C(d_v)$ of v and the fact that no edge between the two was observed in $\tilde{G}$, as $Pr((u,w) \in G) = (1-p)C(d_v)$.

By summing over all such nodes v and neighbors w (which are not known neighbors of u), MAXREACH estimates $d_u^{unknown}$, the number of nodes in $\tilde{G}$ to which u is connected in G but not in $\tilde{G}$, and from Eq. 3, gets d_u^{out}.

Random Node Sample. Suppose that $(u,v),(v,w) \in \tilde{G}$, and $(u,w) \notin \tilde{G}$. Furthermore, suppose that u and w were not selected during the sampling process that generated $\tilde{G}$ (because if they were, MAXREACH would know that they are not connected). Then using the clustering coefficient $C(d_v)$ of v, MAXREACH estimates a $C(d_v)$ probability that $(u,w) \in G$.

By summing these probabilities over all neighbors v of u, and over all neighbors w (where w was not selected during sampling), MAXREACH estimates $d_v^{unknown}$, and thus d_u^{out}.

V. MAXREACH PROBING STAGE: NODE SCORES AND UPDATES

Using the node statistics above, MAXREACH assigns each node a score dependent on the probing scenario. The goal of MAXREACH is to select a node for probing that will maximize the number of new nodes observed through probing.

As information is obtained, scores are updated. Suppose MAXREACH probes node u, and one or more edges (u,v) are observed. We refer to the nodes adjacent to a returned edge as 'affected'. MAXREACH updates the degree estimates for affected nodes, as well as their $d_v^{unknown}$ values.

A. Node Scoring

All-Neighbor Probing: In this probing scenario, when a node is probed, all of its edges are observed. Thus, the score that MAXREACH assigns node u is simply its d_u^{out} value.

k-Neighbor Probing, No Replacement: MAXREACH assigns node u a score of $k\frac{d_U^{out}}{\hat{d}_u - d_u^{known}}$. The numerator is the number of edges leaving $\tilde{G}$, and the denominator is the total number of unobserved edges. This score represents the estimated number of neighbors outside of $\tilde{G}$ that will be observed in k draws.

k-Neighbor Probing, With Replacement: MAXREACH assigns node u a score equal to $Dist(k, d_u)\frac{d_u^{out}}{\hat{d}_u}$, where $Dist(k, \hat{d}_u)$ is the number of distinct elements expected to be observed in k draws from a population of size $\hat{d}_u$. $Dist(k, \hat{d}_u)$ is calculated as a well-known variation on the birthday paradox, and can be calculated as $k[1-(1-\frac{1}{\hat{d}_u})^k]$.

Connection Charge Probing, No Replacement: In this probing scenario, MAXREACH requests some number k of returned edges, and must pay r units for each edge requested, as well as a connection charge of c. From k edges, the estimated number of edges leading outside $\tilde{G}$ is $k\frac{d_u^{out}}{\hat{d}_u - d_u^{known}}$, so the average value per unit cost is $\frac{k}{c+rk}\frac{d_u^{out}}{\hat{d}_u - d_u^{known}}$. The maximum of this occurs when $k = \hat{d}_u$. MAXREACH assigns node u a score of $\frac{\hat{d}_u}{c+r\hat{k}}\frac{d_U^{out}}{\hat{d}_u - d_u^{known}}$. When a node u is selected for probing under this scenario, $\hat{d}_u$ probes are requested.

Connection Charge Probing, With Replacement: This probing scenario is the same as the above, but we are not guaranteed to get a different edge back with each request. MAXREACH finds the positive value of k that maximizes

$$\frac{Dist(k, \hat{d}_u)}{c+rk}\frac{d_u^{out}}{\hat{d}_u}, \tag{5}$$

where $Dist(k, \hat{d}_u) = k[1-(1-\frac{1}{\hat{d}_u})^k]$.The score of each node u is determined using this number of requested edges.

B. Updating Estimated Degrees

Probing Without Replacement. In Section IV-C, MAXREACH estimated the true degree d_u of every node u by using Bayes' Theorem, given the estimated degree distribution of G and the observed degree $\tilde{d}_u$ of u in $\tilde{G}$ before probing. A similar method is used to update node u's estimated degree, except that in Eq. 4, the sum is only taken over degrees $i \geq d_u^{obs} + d_u^{probing}$, where d_u^{obs} is the total observed degree of node u).

This update is performed for all affected nodes.

Probing With Replacement. MAXREACH uses the number of times each node has been selected for probing, and how many duplicate edges were observed from those probes.

Let k_r be the total number of (possibly non-unique) edges of u that have been observed by probing u, and let k be the number of unique edges of u that have been observed when probing u. MAXREACH uses the following equation:

$$E[d_u] = \frac{\sum_i i\hat{q}_i S(k_r,k) N_k i^{-k_r} B(i, \tilde{d}_u)}{\sum_j \hat{q}_j S(k_r,k) N_k j^{-k_r} B(j, \tilde{d}_u)}. \tag{6}$$

$S(k_r,k)$ is the Stirling number of the second kind,[3] and N_k is defined as $(N)(N-1)(N-2)...(N-k+1)$. $S(k_r,k)N_k d^{-k_r}$ is the probability of observing k distinct elements in k_r draws from a population of size d.

As with probing without replacement, the sum is taken over all i greater than or equal to the current observed degree of node u, and this update is performed for all affected nodes.

C. Updating Estimated Connections Into and Out of $\tilde{G}$

When a new edge (u,v) is observed, various values can be updated. d_u^{known} and d_v^{known} represent the observed degrees of u and v, so both of these are incremented. $d_u^{unknown}$ and $d_v^{unknown}$ are recalculated using the process in Section IV-C.

[3]This is the number of ways to divide A elements into k non-empty subsets.

Type	Network	#Nodes	#Edges	Trans.	# CC
Comms	Enron Emails	84K	326K	0.08	950
Comms	Yahoo! IM	100K	595K	0.08	360
Comms	Twit. Replies	261K	309K	0.002	11K
Comms	Twit. Retweets	40K	46K	0.03	4K
Co-occ.	Amazon	270K	741K	0.21	4K
Co-occ.	DBLP	317K	1M	0.31	1

TABLE II
Datasets used in our experiments. 'Trans.' is transitivity, and 'CC' is # of connected components.

D. Removing a Node from Consideration

When we know that a node has no further edges to observe, it is not probed again. Similarly, when $\tilde{G}$ was generated by a Random Node sample, some nodes were fully explored, and are never selected for probing.

VI. Experimental Setup

Our experiments demonstrate that MaxReach outperforms baseline algorithms with respect to maximizing the number of nodes brought into the network.

There are five aspects of the experimental setup: the dataset (Table II), the sampling method used to generate the incomplete network $\tilde{G}$ (Section VI-A), the probing scenario (Section VI-C), the probing budget (Section VI-C), and the baseline strategies (Section VI-B).

A. Sample Generation

In real applications, MaxReach is given an incomplete network. Here, we generate incomplete networks using Random Node and Random Edge sampling (Section II-B).

B. Comparison Strategies

We use three baseline strategies: High Degree, Low Degree, and Random probing. High Degree and Low Degree probing, respectively, assign each node a score equal to its degree or the inverse of its degree, and the highest-scoring node is probed. Random probing selects a random node.[4]

When probing without replacement, a strategy learns that a node has no further edges when a probe returns fewer edges than expected. Similarly, in the All-Neighbor probing scenario, none of the strategies probe the selected node again.

When probing with replacement, scores are modified to reflect the probability of getting a new edge. We multiply each node u's score by p_u, the ratio between the expected number of unobserved edges and the MLE of the total number of edges d_u of node u, calculated as described in [9].

With Connection Charge probing, the strategies estimate each node's degree, as in Section IV. When probing without replacement, each strategy requests edges equal to a node's estimated unobserved degree. When probing with replacement, each strategy requests edges according to Eq. 5.

[4]When the initial $\tilde{G}$ was generated by a Random Node sample, the strategies do not request nodes that were fully explored during sampling.

C. Probing Scenarios and Budgets

We consider the probing scenarios from Section II-A. Note that the probing scenario is a function of the API, and is generally not determined by the user. We calculate a maximum probing budget, and then conduct probes ranging from 0 to this maximum budget, and consider quantiles over this range.

To set the maximum budgets, we use the following methods:

All-Neighbor Probing The maximum budget is set to the number of nodes in the initial $\tilde{G}$ that can be probed.[5]

k-Neighbor Probing For each node in $\tilde{G}$, we calculate the number of unobserved edges adjacent to that node. We divide this value by k (the number of edges returned) to obtain the maximum budget. We consider values of k from 1 to 1000.

Connection Charge Probing For each node in $\tilde{G}$, we calculate the minimum cost to observe all its edges: $c + rd$, where c is the connection charge, r is the edge charge, and d is its unobserved degree. The maximum budget is the sum of these costs over all nodes in $\tilde{G}$. We set $c = 5$ and $r = 1$.

VII. Results and Discussion

Our results demonstrate that over a range of probing scenarios and datasets, MaxReach outperforms the baselines described in Section VI-B by a wide margin. As an example, Figure 1 depicts results on a Random Edge sample of the Enron network under the 5-Neighbor, With Replacement probing scenario. MaxReach is consistently better than the baselines.

A. All-Neighbor Probing

We generate histograms showing aggregate results of MaxReach over the baselines, over all networks and probing budgets. For a given $\tilde{G}$ and budget b, we calculate the log-ratio of the number of nodes added to $\tilde{G}$ by MaxReach vs. by a baseline strategy. Positive log-ratios indicate that MaxReach outperformed the baseline.[6]

A histogram of these log-ratios, over all networks and budgets, is shown in Figure 3 for $\tilde{G}$ produced by a Random Edge sample. The values in boxes state the means and standard deviations of the log-ratios. Results are similar for the case of a Random Node sample. Note that MaxReach outperforms every baseline, on every network, at every probing budget.

B. k-Neighbor Probing

We cannot present such histograms for each of the k-Neighbor probing scenarios, as we consider too many values of k. Instead, we only show the means (i.e., the numbers in boxes in Figure 3) for different values of k. Positive values indicate that MaxReach outperformed the baseline.

Figure 4 depicts results for k-Neighbor, No Replacement probing with Random Node sampling (other results were similar). All values are positive: on average, MaxReach outperforms the baselines. As k increases, the probing scenario becomes All-Neighbor probing, so the values stabilize.

[5]This is simply the number of nodes in $\tilde{G}$. If $\tilde{G}$ was produced by Random Node sampling, we do not include nodes already sampled.

[6]We use log-ratios because when MaxReach is better than a baseline, the log-ratio has the same magnitude as the case when the baseline is better than MaxReach by the same amount.

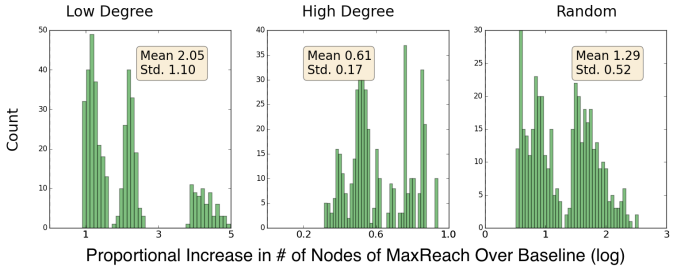

Fig. 3. Histogram of log-ratios (with mean and stdev) of nodes added by MAXREACH vs. baselines. Probing was conducted on 10% Random Edge samples, under All-Neighbor probing. MAXREACH outperforms all baselines.

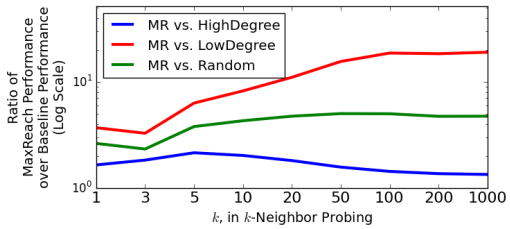

Fig. 4. Mean log-ratio of number of nodes added to $\tilde{G}$ by MAXREACH (MR) vs. baseline strategies. Probing was conducted on Random Node samples, without replacement, with k-Neighbor probing, for varying values of k. MAXREACH outperforms all baselines. See text for explanation of trends.

C. Connection Charge Probing

MAXREACH performs substantially better than all considered baselines, over different probing budgets and network datasets. For example, on Random Edge samples, MAXREACH outperforms Low Degree probing by a mean log-ratio of 1.29 (stdev of 0.9), High Degree probing by 3.11 (stdev of 1.3), and Random probing by 0.86 (stdev of 0.49).

D. Running Time

MAXREACH is typically slower than the three baseline methods. This is partly because it adds many new nodes to $\tilde{G}$, so more updates are needed. This can be controlled for by calculating the number of new nodes observed per second.

Under All-Neighbor probing, High Degree probing is fastest, observing up to 100x more nodes per second, but under k-Neighbor probing, MAXREACH is slower only by a factor of 3x, and is similar to the other baselines. For example, on the Twitter Replies network, under 20-Neighbor probing, With Replacement, MAXREACH observes 1100 new nodes/second, while High Degree probing observes 2400 nodes/second. Of course, per unit of budget, MAXREACH is the best.

VIII. Related Work

There is a rich literature on sampling graphs. Avrachenkov et al. [2] use queries to locate high-degree nodes, and O'Brien and Sullivan [8] use local information to estimate core numbers. Hanneke and Xing [4], and Maiya and Berger-Wolf [6] examine online sampling for centrality measures. Cho et al. [3] determine which URLs to examine in a Web crawl. In the network completion problem, one attempts to infer missing parts of a network given a small subset [7], [5] or infer network structure from diffusion information [12].

MAXREACH is based on MAXOUTPROBE, by the same authors [10]. Unlike MAXREACH, MAXOUTPROBE cannot probe nodes as they are added to the network. The Maximum Expected Uncovered Degree sampling method [1] is also related to MAXREACH. In each step, MEUD expands the node with the highest expected degree, but requires knowledge of the true degree of each node.

IX. Conclusions

We discussed the problem of determining which nodes in an incomplete network to probe in order to maximize the number of new nodes observed. We presented MAXREACH, which estimates the degree of each observed but unexplored node, as well as the clustering coefficients of nodes, in order to rank nodes for probing. Over a range of realistic probing scenarios, on networks from diverse domains, MAXREACH outperforms several baseline approaches at the task of adding as many nodes as possible to the network.

X. Acknowledgements

Soundarajan and Eliassi-Rad were supported by NSF CNS-1314603 and by DTRA HDTRA1-10-1-0120. Gallagher was supported by LLNL under Contract DE-AC52-07NA27344. Pinar was funded by the LDRD program of the Sandia National Labs, which is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the DOE's NNSA under contract DE-AC04-94AL85000.

References

[1] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley. Pay few, influence most: Online myopic network covering. In *IEEE NetSciCom Workshop*, 2014.
[2] K. Avrachenkov, N. Litvak, L. O. Prokhorenkova, and E. Sayargulova. Quick detection of high-degree entities in large directed networks. In *ICDM*, pages 20–29, 2014.
[3] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. In *WWW*, pages 161–172, 1998.
[4] S. Hanneke and E. P. Xing. Network completing and survey sampling. In *AISTATS*, pages 209–215, 2009.
[5] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SDM*, pages 47–58, 2011.
[6] A. S. Maiya and T. Y. Berger-Wolf. Online sampling of high centrality individuals in social networks. In *PAKDD*, pages 91–98, 2010.
[7] F. Masrour, I. Barjesteh, R. Forsati, and A.-H. Esfahanian. Network completion with node similarity: A matrix completion approach with provable guarantees. In *ASONAM*, pages 302–307, 2015.
[8] M. P. O'Brien and B. D. Sullivan. Locally estimating core numbers. In *ICDM*, pages 460–469, 2014.
[9] E. Samuel. Sequential maximum likelihood estimation of the size of a population. *Annals of Mathematical Statistics*, 39(3):1057–1068, 1968.
[10] S. Soundarajan, T. Eliassi-Rad, B. Gallagher, and A. Pinar. Maxoutprobe: An algorithm for increasing the size of partially observed networks. *CoRR*, 2015.
[11] C. Tsourakakis. Fast counting of triangles in large real networks without counting: Algorithms and laws. In *ICDM*, pages 608–617, 2008.
[12] M.-H. Yang, C.-K. Chou, and M.-S. Chen. Cluster cascades: Infer multiple underlying networks using diffusion data. In *ASONAM*, pages 281–284, 2014.
[13] Y. Zhang, E. D. Kolaczyk, and B. D. Spencer. Estimating network degree distributions under sampling: An inverse problem, with applications to monitoring social media networks. *Annals of Applied Statistics*, 9(1):166–199, 2015.