MaxOutProbe: An Algorithm for Increasing the Size of Partially Observed Networks

Sucheta Soundarajan
Syracuse University
susounda@syr.edu

Tina Eliassi-Rad
Rutgers University
tina@eliassi.org

Brian Gallagher
Lawrence Livermore National Laboratory
bgallagher@llnl.gov

Ali Pinar
Sandia National Laboratories
apinar@sandia.gov

Abstract

Networked representations of real-world phenomena are often partially observed, which lead to incomplete networks. Analysis of such incomplete networks can lead to skewed results. We examine the following problem: given an incomplete network, which b nodes should be probed to bring the largest number of new nodes into the observed network? Many graph-mining tasks require having observed a considerable amount of the network. Examples include community discovery, belief propagation, information propagation, etc. For instance, consider someone who has observed a portion (say 1%) of the Twitter retweet network via random tweet sampling. She wants to estimate the size of the largest connected component of the fully observed retweet network. To improve her estimate, she uses her limited budget to reduce the incompleteness of the network. In this work, we propose a novel algorithm, called MAXOUTPROBE, which uses a budget b (on nodes probed) to increase the size of the observed network in terms of the number of nodes. We compare MAXOUTPROBE to the best existing probing approaches across various incomplete networks observed via different sampling methods. Across a range of conditions, MAXOUTPROBE demonstrates consistently high performance in comparison to existing methods.

1 Introduction

Suppose that one has observed an incomplete portion $\hat{G}$ of some larger complete network G.[1] To learn more about the structure of G, one can probe nodes from $\hat{G}$, where each such probe reveals all the neighbors of the selected node from G. The question we address here is: which nodes from $\hat{G}$ should be probed to observe as many nodes as possible from G? This question is related to previous works on graph sampling and crawling. However, unlike much of the work in graph sampling, we are not attempting to generate a sample from scratch. Instead, we are studying how one can enhance or improve an existing incomplete network observation, without control over how it was generated or observed. Our work is motivated by problems where one only has a partial observation of the complete network; but needs the most accurate and informative picture of the complete network. For example, suppose a network administrator who has partially observed a computer network through traceroutes. Which parts of the observed network should she more closely examine to get the best (i.e., most complete) view of the entire network? With a limited query budget, how should this further exploration be done? Alternatively, suppose that one has obtained a sample of the Twitter

[1] Throughout this paper, when we use the term *complete*, we are referring to the completeness of the data (i.e., no information is missing), rather than to a clique structure.

network from another researcher. The sample was collected for some other purpose, and so may not contain the most useful structural information for one's purposes. How should one best supplement this sampled data?

We present a novel algorithm, called MAXOUTPROBE, whose goal is to select b nodes from $\hat{G}$ for probing, such that the greatest number of new nodes are brought into $\hat{G}$. MAXOUTPROBE achieves this goal by ranking each node u based on its estimate of have many neighbors u has outside of $\hat{G}$. It then selects the top b nodes in this ranking. In particular, MAXOUTPROBE consists of two steps: (1) *Estimation* and (2) *Selection*. First, during the *Estimation Step*, for each node u in $\hat{G}$, MAXOUTPROBE estimates u's true degree in G; it also estimates G's average clustering coefficient. Second, during the *Selection Step*, MAXOUTPROBE uses these two quantities to estimate the number of nodes outside $\hat{G}$ to which u is adjacent. The b nodes that are predicted to have the most neighbors outside the incomplete network are selected for probing.

We evaluate MAXOUTPROBE on six network datasets, using incomplete networks observed by four popular sampling method, and show that MAXOUTPROBE selects nodes for probing that are more valuable than those selected by comparison methods.

The **contributions** of our paper are as follows: **(1)** We present MAXOUTPROBE, a two-step algorithm that first estimates graph statistics, and then selects probes in accordance with those statistics. Unlike existing literature on estimating network characteristics, we do not assume knowledge of how the incomplete network was observed, and present a method for estimating the relevant statistics without such background knowledge. **(2)** We show that with respect to the task of bringing as many nodes as possible into the incomplete network, the graphs obtained by probing nodes according to MAXOUTPROBE are substantially better than those obtained by probing according to the other competing strategies.

2 Problem Definition

We are given an incomplete, partially observed graph $\hat{G}$ that is a part of a larger, fully observed graph G. We wish to gain broader observability of G; that is, we wish to observe as many nodes as possible from G. To aid in this task, we are allowed to probe a specified number b additional nodes in $\hat{G}$ and gain more information about the probed nodes. Thus, the problem at hand is to select the b nodes which maximize the number of nodes observed from G using only the structure provided in $\hat{G}$. Let $\hat{G}'$ represent the augmented graph−i.e., $\hat{G}$ with the information obtained from the probes.

In this work, *probing* a node u is defined as learning all of u's neighbors (e.g., querying Facebook for a list of all friends of a user or learning all e-mail contacts of an individual). Moreover, there is no 'master list' of nodes in G that allows an algorithm to probe nodes it has not yet seen. Thus, only nodes that already exist in $\hat{G}$ can be probed.

Figure 1 illustrates the probing process. Red nodes in $\hat{G}$ have already been fully explored. Yellow nodes are present in $\hat{G}$, but have not yet been fully explored: these are the candidates for probing. Green nodes are not yet present in $\hat{G}$, and we have no knowledge of them. By probing yellow nodes, we learn about the existence of the green node. We refer to the red nodes as 'fully explored', and the yellow nodes as 'unexplored.'

3 Proposed Method: MaxOutProbe

Given a budget of b nodes to probe, MAXOUTPROBE selects which b unprobed nodes in $\hat{G}$ (i.e., yellow nodes in Figure 1) are adjacent to many nodes outside $\hat{G}$ (i.e., green nodes in Figure 1). MAXOUTPROBE does not make assumptions about how $\hat{G}$ was observed.

MAXOUTPROBE has two steps: Estimation and Selection. During the Estimation Step, a small amount of budget is used to probe nodes from $\hat{G}$. The resulting information is used to estimate necessary statistics of G. During the Selection Step, the statistics estimated during the Estimation Step are used to score each node in $\hat{G}$ with respect to how many neighbors outside of $\hat{G}$ that node may have. The highest scoring b nodes are then selected for probing.

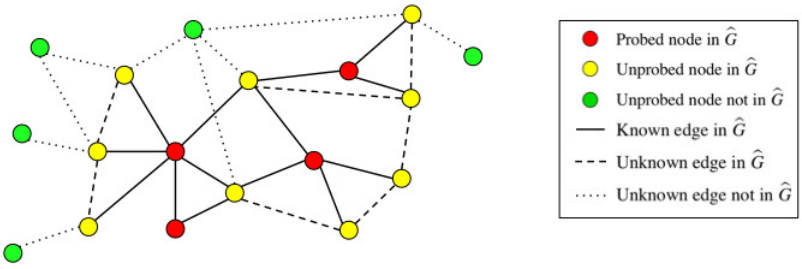


Figure 1: Overview of probing. Red nodes are already fully explored. The problem is to select yellow nodes (unprobed but in $\hat{G}$) that are adjacent to many green nodes (unprobed and not in $\hat{G}$).

Let a *candidate node* be a node in $\hat{G}$ that was not fully observed during the process of observing $\hat{G}$ (i.e., such a node is a candidate for probing). For each candidate node u in $\hat{G}$, MAXOUTPROBE estimates d_u^{out}, the number of u's neighbors that lie outside of $\hat{G}$, as follows:

$$d_u^{out} = d_u - d_u^{in} = d_u - d_u^{known} - d_u^{unknown} \tag{1}$$

where:

- d_u is u's true degree in G. This quantity is unknown and must be estimated.
- d_u^{in} is the number of nodes in $\hat{G}$ that u is adjacent to in G. This includes:
  - d_u^{known}, the number of nodes in $\hat{G}$ that we already know to be adjacent to u. This quantity can be directly calculated.
  - $d_u^{unknown}$, the number of nodes in $\hat{G}$ that u is connected to in G, but not in $\hat{G}$ (i.e., the connections to u that have not been observed). This quantity must be estimated.

Once d_u^{out} has been calculated, MAXOUTPROBE selects the b nodes with the highest such values (where b is the probing budget).

We next describe how MAXOUTPROBE estimates each node u's true degree d_u as well as the graph's average clustering coefficient C, which is used to estimate $d^{unknown}$ for each node. Section 3.2 describes how MAXOUTPROBE puts these various pieces together to obtain estimates for d_u^{out}.

3.1 MAXOUTPROBE's Estimation Step

MAXOUTPROBE estimates each candidate node u's true degree d_u and the graph's clustering coefficient C. Note that although there is a large body of work on estimating graph statistics, those works typically assume knowledge about how the graph was observed/generated (see Section 5). In contrast, MAXOUTPROBE makes no such assumption. Instead to estimate the necessary statistics of G, MAXOUTPROBE performs a series of initial probes.

Estimating Node Degrees. We empirically observe that for incomplete networks, there exists some scale factor f such that multiplying u's sample degree by $\frac{1}{f}$ gives a good approximation of d_u. Thus, if MAXOUTPROBE estimates f, it can estimate the true degree d_u of every node in the incomplete network by using that node's observed degree. To perform this estimate, given a total probing budget of b, MAXOUTPROBE begins by identifying the b candidate nodes from $\hat{G}$ with the highest degrees in $\hat{G}$. A small number of these nodes are randomly selected for probing,[2] and their true degrees in G are learned. By taking the average of the ratios of each selected node's true degree to its degree in $\hat{G}$, MAXOUTPROBE estimates f.

Estimating Average Clustering Coefficient. Recall that the clustering coefficient of a node is the fraction of its neighbors that are connected, divided by the maximum possible such value. For a candidate node u, MAXOUTPROBE needs to know how many nodes that are presently two steps away

[2]In our experiments, we select 100 such nodes, but this value can vary depending on the desired accuracy and total probing budget available.

from u in $\hat{G}$ are likely to be connected to u in G. Thus, rather than estimating the global average clustering coefficient or the average clustering coefficient for a candidate node, MAXOUTPROBE must estimate the average clustering coefficient for nodes that are adjacent to the candidate nodes. It is the clustering coefficient of these neighboring nodes that governs the number $d_u^{unknown}$. To make this estimation, MAXOUTPROBE uses the nodes selected for probing during the estimation of f, as described above. For each of these selected nodes u, MAXOUTPROBE considers each node v that was two steps away in $\hat{G}$. MAXOUTPROBE then observes whether, after probing u, whether u connected to v. By averaging these results over all selected nodes u, MAXOUTPROBE estimates the average clustering coefficient.

Remarks. MAXOUTPROBE's estimates for the degree of a candidate node and the clustering coefficient of the graph are unbiased for two popular forms of observing graphs: random node sampling and random edge sampling. Due to brevity, we have omitted these proofs.

3.2 MAXOUTPROBE's Selection Step

MAXOUTPROBE uses the estimates described above to select the b candidate nodes that have the most unobserved neighbors. As before, u is a (candidate) node from $\hat{G}$ that was not fully explored during the observation process. Let f_E and f_N, respectively, denote the fraction of edges and nodes from G that are present in $\hat{G}$. To estimate $d_u^{unknown}$, MAXOUTPROBE uses the knowledge that real-world social and information networks tend to exhibit high clustering (i.e., nodes tend to connect to friends-of-friends). More precisely, let w be an unexplored node that is two hops away from a candidate node u, such that w and u are not connected in $\hat{G}$ (but they may be connected in G). w forms an open wedge with u, because w and u share at least one neighbor. Let W_u be the number of such nodes w; these are the nodes in $\hat{G}$ to which u is most likely connected. In the Estimation Step, MAXOUTPROBE estimated the average clustering coefficient C of the network, which defines the fraction of wedges that are triangles (that is, the ratio of 3 times the number of triangles in the network to the number of length-2 paths in the network). Given this value, MAXOUTPROBE estimates that u is connected to $C \times W_u$ nodes in $\hat{G}$, in addition to its known neighbors in $\hat{G}$.[3]

Putting this all together, MAXOUTPROBE obtains the estimate for the number of neighbors that u has outside of $\hat{G}$:

$$d_u^{out} = d_u - d_u^{known} - d_u^{unknown} = d_u - d_u^{known} - (C \times W_u) \tag{2}$$

d_u^{known} and W_u can be calculated exactly from $\hat{G}$. The challenge thus lies in estimating d_u and C, which was described in the Estimation Step above. MAXOUTPROBE computes d_u^{out} for all candidate nodes in $\hat{G}$ and selects the b candidate nodes with the highest d^{out} values.

4 Experiments

Our experiments demonstrate that MAXOUTPROBE outperforms a variety of baseline algorithms with respect to the task of maximizing the number of nodes brought into the observed network, across incomplete networks observed/generated by different popular sampling methods.

Datasets. Table 1 describes the six real-world networks used in our experiments. Four are communications networks, and the other two are co-occurrence networks. The first co-occurrence network is an Amazon co-purchasing network with the nodes denoting books; and the second is a Youtube co-watching network with the nodes representing videos.

Sampling Methods. We generate incomplete networks using sampling methods, each corresponding to real application scenarios. For each method, we sample 10% of the edges from the original network. We consider the following sampling methods.

RandNode: We sample nodes at random. The sample contains those nodes plus all of their neighbors. We assume that a list of the selected nodes is available. We select one node at a time until our sample reaches 10% of the total number of edges in G_{orig}.

[3] Although u's individual clustering coefficient would be more valuable here than the global clustering coefficient C, MAXOUTPROBE has incomplete information about u, and thus uses C as an approximation for the per-node clustering coefficients.

Type	Network G	# of Nodes	# of Edges	G's Clust. Coeff.	# of Conn. Comp.
Communications	Enron Emails	84K	326K	0.08	950
Communications	Yahoo! IM	100K	595K	0.08	360
Communications	Twitter Replies	261K	309K	0.002 Comm.	11,315
Communications	Twitter Retweets	40K	46K	0.03 Comm.	3,896
Co-occurrences	Amazon Books	270K	741K	0.21	3840
Co-occurrences	Youtube Videos	167K	1M	0.007	1

Table 1: Statistics for the networks considered in our experiments. Clust. Coeff. is short for clustering coefficient. Conn. Comp. is short for connected components.

Category	Sub-category	Name	Description
Exploit	Degree	**HighDeg, LowDeg**	Select the highest or lowest degree nodes.
	Structural Hole	**HighDisp, LowDisp**	Select the highest or lowest dispersion nodes.
		CrossComm	Pick nodes with the highest fraction of neighbors outside of their community (as identified with the Louvain method [5]).
	Clustering	**HighCC, LowCC**	Select the highest or lowest clustering coefficient nodes.
Explore		**Random**	Randomly select nodes from the sample.

Table 2: Baseline probing strategies. We categorize strategies as explore or exploit, and further subdivide as degree-based, structural hole-based, or random. Dispersion is an edge-based measure of how well a node's neighbors are connected to each other [4]. For each node, we average the dispersion of each of its adjacent edges.

RandEdge: Random edges are selected uniformly at random. Because edges, rather than nodes, are sampled, no list of sampled nodes is available. The Twitter Gardenhose, for example, is a random 10% of the tweet stream (where each retweet represents an edge).

Random Walk (RW) and **Random Walk w/ Jump (RWJ)**: These sampling methods begin at a random node and repeatedly transition to a random neighbor. In the RWJ method, at each step there is a chance (we use 15%) of jumping to a random node. A single edge at a time is observed. Thus, no list of sampled nodes is available. These are common sampling methods for large networks [12].

Baseline and Competing Approaches. Table 2 describes a variety of baseline and competing methods, each with a simple scoring function for selecting nodes.

These methods each rank all of the nodes in the incomplete graph $\hat{G}$ (except those that have already been fully explored). Some of the methods use *exploit* strategies, which strategically select nodes, while others utilize *explore* strategies, which randomly select nodes.

High degree probing relies on the intuition that high-degree nodes in $\hat{G}$ are connected to many nodes outside of $\hat{G}$. The structural hole methods target nodes that "fill" structural holes, thus connecting different parts of the graph [6]. On a similar intuition, the CROSSCOMM algorithm selects nodes on the border of two communities.

Random probing selects random nodes from within $\hat{G}$ for probing.

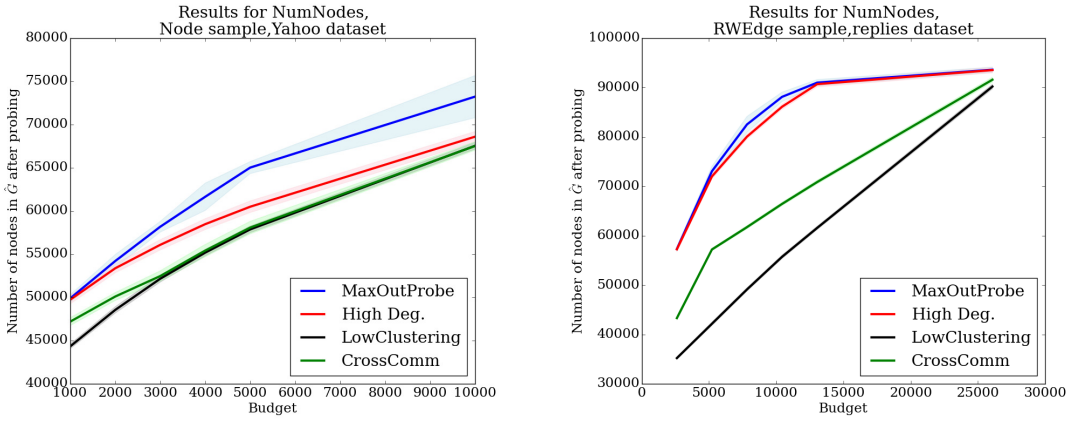
Experimental Setup.

For each network listed in Table 1, we generate 20 incomplete networks using each of the four sampling methods described above. Each incomplete network contains 10% of the edges from the original network. For each probing approach, we conduct probes at budgets $b \in \{1\%, 2\%, 3\%, 4\%, 5\%\}$ of the number of nodes in G.

After conducting probes on an incomplete network $\hat{G}$, we obtain an augmented sample graph $\hat{G}'$. To evaluate the quality of $\hat{G}'$, we simply count how many nodes it has.[4]

Results. First, we evaluate MAXOUTPROBE and each of the baseline approaches on each of the incomplete networks described above. We observe that averaged over the 20 incomplete networks,

[4]Since each $\hat{G}$ graph starts with the same number of nodes, counting the number of nodes in $\hat{G}'$ tells us how many new nodes were observed.



(a) (b)

Figure 2: (a) Results of MAXOUTPROBE and several baseline probing strategies on 20 incomplete networks generated by random node sampling on the Yahoo! IM network. Shading indicates one standard deviation. Observe that MAXOUTPROBE is the best across all considered budgets. (b) Results of MAXOUTPROBE and several baseline probing strategies on 20 incomplete networks generated by random walk sampling on the Twitter Replies network. Shading indicates one standard deviation. Observe that MAXOUTPROBE and High Degree probing perform similarly because the Twitter Replies network has a very low clustering coefficient (of 0.002).

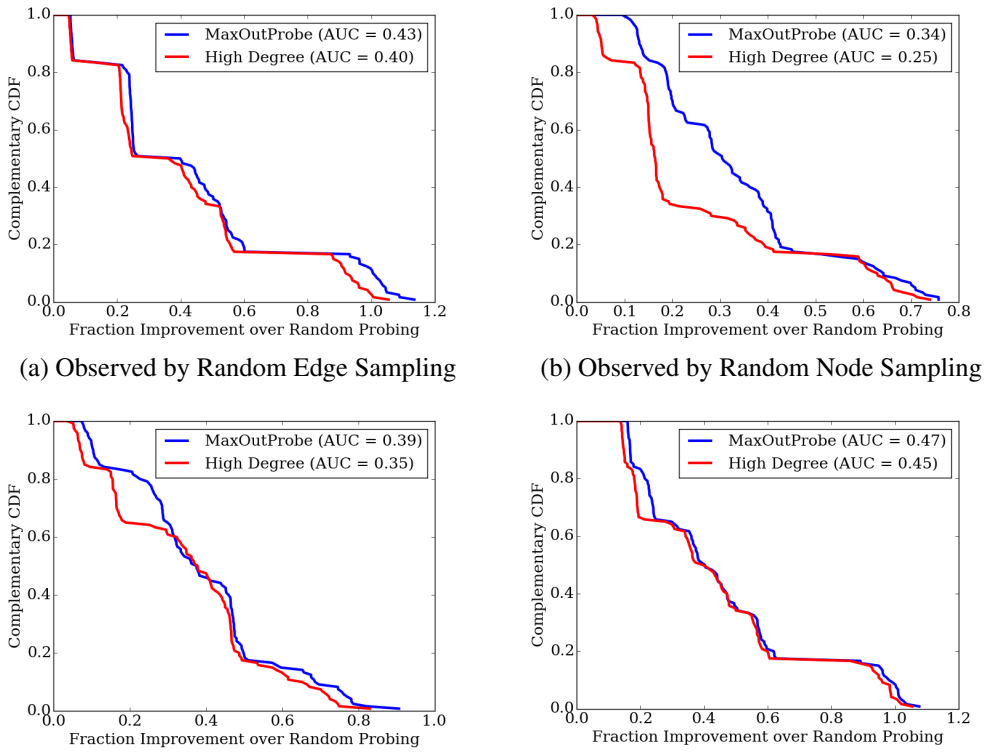
for every sampling method and every network, MAXOUTPROBE matches or outperforms the best baseline strategies. For example, see Figure 4(a), which shows the performance of several strategies on an incomplete network generated by the RandNode sampling process on the Yahoo! IM network (for the sake of space, we present only the best baseline approaches).

In some cases, MAXOUTPROBE performs roughly the same as High Degree probing (see, e.g., Figure 4(b), for results on incomplete samples generated by random walk sampling on the Twitter Replies network). This occurs when the incomplete network has a very low estimated clustering coefficient. On these incomplete networks, MAXOUTPROBE estimated average clustering coefficients ranges from 0.0 to 0.03. With such a low clustering coefficient, MAXOUTPROBE reduces to effectively picking the nodes with the highest observed degrees.

Due to space constraints, we cannot present plots for each network and sampling method. We thus aggregate results across datasets, comparing MAXOUTPROBE to High Degree probing (which is typically the best baseline appraoch).

For each incomplete network and each considered budget, we conduct probes using MAXOUTPROBE and High Degree probing, and count the number of nodes in each resulting $\hat{G}'$. Because we cannot meaningfully compare these numbers across datasets or sampling methods, we also conduct the same number of probes using Random probing, and then calculate the percent by which MAXOUTPROBE and High Degree probing improved over Random probing (i.e., how much larger is the $\hat{G}'$ produced by MAXOUTPROBE or High Degree probing versus that produced by Random probing?). By comparing to Random probing, we are able to make sensible comparisons across datasets.

Figure 4 shows complementary CDFs of these aggregate results at the 0.05 probing budget on incomplete networks generated, respectively, by Random Edge sampling, Random Node sampling, Random Walk sampling, and Random Walk with Jump sampling. Similar results were observed at other probing budgets. The x-axis represents the percent improvement over Random probing, with values greater than 0 indicating that the approach performed better than Random probing. The y-axis represents the fraction of such results that produced at least a certain fraction improvement over random. We see that in all cases, MAXOUTPROBE has a higher area under the curve than High Degree probing, indicating greater improvement. The area under the curve for MaxOutProbe ranges from 4% to 36% higher than that of High Degree probing.



(a) Observed by Random Edge Sampling

(b) Observed by Random Node Sampling

(c) Observed by Random Walks

(d) Observed by Random Walks with Jumps

Figure 3: Complementary CDFs showing results of MAXOUTPROBE and High Degree probing as compared to Random probing across all datasets, on incomplete networks observed by various sampling methods. In all cases, MAXOUTPROBE has a higher area under the curve, indicating a greater improvement over Random probing than High Degree probing.

5 Related Work

Our work is most related to graph sampling and crawling. There is a rich literature on sampling graphs. For instance, previous literature has examined the study of community detection from graph samples—e.g., by using minimum spanning trees (MSTs) [20], or by expanding a graph sample [14]. Avrachenkov et al. [3] use queries to locate high-degree nodes. O'Brien and Sullivan [16] use local information to estimate the core number of a node. Hanneke and Xing [9] and Kim and Leskovec [11] infer characteristics of a larger graph given a sample.

Maiya and Berger-Wolf [15] study the problem of online sampling for centrality measures. Cho et al. [7] study the problem of determining which URLs to examine in a Web crawl. In contrast, we study the problem of selecting which nodes from an existing incomplete network one should probe, rather than constructing a sample from scratch. Also, unlike many sampling methods, we are not down-sampling a network to which we have complete access, but using a limited number of probes. Additionally, we select probes in a batch, not incremental, manner. This difference is critical when obtaining data is time-consuming.

The closest method to ours is the MEUD (Maximum Expected Uncovered Degree) algorithm by Avrachenkov et al. [2]: a sampling method intended to bring as many nodes as possible into the sample. MEUD begins with one node; then in each step, it grows the sample by adding the node with the highest expected degree in the underlying, unobserved network. MEUD requires knowing the degree distribution of the underlying network, but for certain classes of graphs, reduces to selecting the nodes with the highest observed degrees. This is a valid approximation only for limited classes of random graphs, while our proposed method does not have such constraints. Note that if we modified the MEUD algorithm to fit our problem setting, it would be most like the High Degree Probing.

Although we are the first to conduct a study on the problem of probing incomplete graphs, researchers interested in specific network features have studied related problems. Macskassy and Provost [13] address the problem of identifying malicious entities in incomplete network data by gathering more information about suspicious entities.

Cohen et al. [8] propose a strategy for immunizing a population in an unobserved network by targeting neighbors of randomly selected nodes. Shakkottai [18] considers the problem of nodes in a sensor network attempting to find the source of information. Ragoler et al. [17] study the problem of determining the frequency at which sensor nodes should query their environments.

Lastly, theoreticians have studied subgraphs through concepts such as monotone graph properties (e.g., [1]).

6 Conclusions

We discussed the problem of determining which nodes in an incomplete network to probe in order to maximum the number of new nodes observed. We presented the MAXOUTPROBE algorithm, which is based on estimating the degree of each observed but unexplored node, as well as the graph's average clustering coefficient in order to rank nodes for probing. We demonstrated that MAXOUTPROBE outperforms several baseline and competing approaches, on incomplete networks observed/generated by four popular sampling methods over six network datasets. In cases where the graph has a very low clustering coefficient, MAXOUTPROBE's performance is similar to selecting nodes with the highest observed degree.

7 Acknowledgements

Ali Pinar's work is supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Complex Interconnected Distributed Systems (CIDS) program and the DARPA GRAPHS Program, and the Laboratory Directed Research and Development program of Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory (LLNL) under Contract DE-AC52-07NA27344. Soundarajan and Eliassi-Rad were supported by LLNL, by NSF CNS-1314603, by DTRA HDTRA1-10-1-0120, and by DAPRA under SMISC Program Agreement No. W911NF-12-C-0028.

References

[1] N. Alon and A. Shapira. Every montone graph property is testable. *SIAM J. Comput.*, 38(2):505–522, 2008.

[2] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley. Pay few, influence most: Online myopic network covering. In *IEEE NetSciCom Workshop*, 2014.

[3] 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.

[4] L. Backstrom and J. M. Kleinberg. Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook. In *CSCW*, pages 831–841, 2014.

[5] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, page 10008, 2008.

[6] R. S. Burt. The social capital of structural holes. In M. F. Guillen, R. Collins, P. England, and M. Meyer, editors, *New Directions in Economic Sociology*. Russell Sage Foundation, 2002.

[7] J. Cho, H. Garcia-Molina, and L. Page. Efficient crawling through URL ordering. In *WWW*, pages 161–172, 1998.

[8] R. Cohen, S. Havlin, and D. Ben-Avraham. Efficient immunization strategies for computer networks and populations. *Phys. Rev. Lett.*, 91(24):247901, 2003.

[9] S. Hanneke and E. P. Xing. Network completing and survey sampling. In *AISTATS*, pages 209–215, 2009.

[10] W. Hoeffding. Probability inequalities for sums of bounded random variables. *J. American Statistical Association*, 58(30):13–30, 1963.

[11] M. Kim and J. Leskovec. The network completion problem: Inferring missing nodes and edges in networks. In *SDM*, pages 47–58, 2011.

[12] J. Leskovec and C. Faloutsos. Sampling from large graphs. In *KDD*, pages 631–636, 2006.

[13] S. A. Macskassy and F. Provost. Suspicion scoring based on guilt-by-association, collective inference, and focused data access. In *Int'l Conf. on Intell. Analysis*, 2005.

[14] A. S. Maiya and T. Berger-Wolf. Sampling community structure. In *WWW*, pages 701–710, 2010.

[15] A. S. Maiya and T. Y. Berger-Wolf. Online sampling of high centrality individuals in social networks. In *PAKDD*, pages 91–98, 2010.

[16] M. P. O'Brien and B. D. Sullivan. Locally estimating core numbers. In *ICDM*, pages 460–469, 2014.

[17] I. Ragoler, Y. Matias, and N. Aviram. Adaptive probing and communication in sensor networks. In *ADHOC-NOW*, volume 3158, pages 280–293, 2004.

[18] S. Shakkottai. Asymptotics of query strategies over a sensor network. In *INFOCOM*, pages 548–557, 2004.

[19] D. Thompson, J. Bennett, C. Seshadhri, and A. Pinar. A provably-robust sampling method for generating colormaps of large data. In *LDAV*, pages 77–84, 2013.

[20] J. Wu, X. Li, L. Jiao, X. Wang, and B. Sun. Minimum spanning trees for community detection. *Physica A*, 392(9):2265–2277, 2013.

8 Appendix

8.1 Estimations for Known Samples

Given additional information about the process by which $\hat{G}$ was generated, MAXOUTPROBE may sometimes be able to produces unbiased estimates for the node degrees and the average clustering coefficient without performing initial estimation probes. Although additional information is required, one is able to bypass the initial use of budget that would otherwise be required to estimate these parameters.

In this section, we discuss how MAXOUTPROBE can perform such estimations for the cases when one knows that $\hat{G}$ was generated by either the RandNode or RandEdge sampling methods discussed in Section 4, and one also knows the total number of nodes and edges in G.

8.1.1 Estimations for Random Node Sample

Estimating degree d_u: The Random Node sampling method randomly selects f_N fraction of nodes from G for exploration, and learns all neighbors of the selected nodes. Thus, if node u has observed degree d_u^{known} in $\hat{G}$, MAXOUTPROBE estimates its true degree as $\frac{1}{f_N} \times d_u^{known}$. E.g., if u is adjacent to 5 nodes in $\hat{G}$, and 10% of the nodes from G were selected during sampling, then MAXOUTPROBE estimates u's true degree d_u as 50. This estimation works well for high degree nodes, but is less accurate for low degree nodes. However, as observed in Section 4, note that low degree nodes are unlikely to be useful probes, so this method is accurate where it needs to be.

Claim: The estimator for d_u is unbiased.

Proof: Let $\hat{d_u}$ represent the estimator for d_u. We must show that for each u, $E(\hat{d_u}) = d_u$. Recall that f_N fraction of nodes from G were selected uniformly at random during sampling to produce $\hat{G}$; these nodes plus their neighbors constitute $\hat{G}$.

Consider a node u that has not been explored during the sampling process, but is present in the sample $\hat{G}$ (that is, u is adjacent to a node that was explored during sampling). Node u has d_u neighbors in G, and because f_N of the nodes from G were sampled uniformly at random to produce $\hat{G}$, $E(d_u^{known}) = f_N d_u$. Thus, $\frac{1}{f_N} E(d_u^{known}) = d_u$, so $E(\frac{1}{f_N} d_u^{known}) = d_u$. $\hat{d_u} = \frac{1}{f_N} d_u^{known}$, so $E(\hat{d_u}) = d_u$, so $\hat{d_u}$ is an unbiased estimator for d_u. □

Estimating clustering coefficient C: To find C, MAXOUTPROBE estimates the number of wedges (2-paths) and triangles in G.

Suppose that a triangle (x, y, z) exists in G. What is the probability that it will be preserved in $\hat{G}$? In order for us to observe all edges (x, y), (y, z), and (x, z) in $\hat{G}$, then at least one node from each of the three member edges must be fully explored during sampling. Thus, the triangle will be preserved if and only if at least two of the three nodes are selected. We can write the probability p_T that this occurs as:

$$p_T = 3f_N^2(1 - f_N) + f_N^3 \tag{3}$$

In other words, p_T is the probability that exactly two of the three nodes are selected, plus the probability that all three are selected. Thus, by multiplying the observed number of triangles in $\hat{G}$ by $\frac{1}{p_T}$, MAXOUTPROBE estimates the number of triangles T in G.

Now we calculate the probability that a length-2 path (x, y, z) from G is preserved in $\hat{G}$ (x may or may not be connected to z). To observe this path in $\hat{G}$, either x and z must *both* be probed, or y must be probed. The probability p_W of this occurring is as follows:

$$p_W = f_N^3 + 3(f_N^2(1 - f_N)) + f_N(1 - f_N)^2 \tag{4}$$

This is the probability that all three nodes are probed plus the probability that two nodes are probed plus the probability that just y is probed. MAXOUTPROBE then multiplies the observed number of length-2 paths from $\hat{G}$ by $\frac{1}{p_W}$ to estimate the number of wedges W in G.

G's clustering coefficient C is then estimated as $\frac{3T}{W}$.

Claim: The estimator for C is unbiased.

Proof: Let $\hat{C}$ represent the estimator for C. We must show that $E(\hat{C}) = C$. f_N fraction of nodes from G were selected uniformly at random during sampling to produce $\hat{G}$.

Suppose that wedge (x, y, z) from G is present in $\hat{G}$. Suppose that (x, y, z) is a closed triangle in G (so edge (x, z) is present in G): what is the probability that edge (x, z) is present in $\hat{G}$, so (x, y, z) is a closed triangle in $\hat{G}$? Because we assume that (x, y, z) is present in $\hat{G}$, some of those three nodes must have been selected during the sampling process.

There are three possibilities: (1) all three of x, y, z were selected, (2) exactly two of x, y, z were selected, or (3) only y was selected (if only x or only z was selected, then either edge (y, z) or edge (x, z) would be absent from $\hat{G}$.

Possibility (1) occurs with probability f_N^3. Possibility (2) occurs with probability $f_N^2(1 - f_N)$. Possibility (3) occurs with probability $f_N(1 - f_N)^2$. In possibilities (1) and (2), edge (x, z) will certainly be present in $\hat{G}$. In possibility (3), edge (x, z) will not be present in $\hat{G}$. Thus, given that wedge (x, y, z) is present in $\hat{G}$, the probability P_{closed} that edge (x, z) is also present in $\hat{G}$ is as follows:

$$P_{closed} = \frac{f_N^3 + e f_N^2(1 - f_N)}{f_N^3 + 3f_N^2(1 - f_N) + f_N(1 - f_N)^2} \tag{5}$$

Thus, of the wedges that are closed wedges in G, we expect that P_{closed} fraction of the wedges present in $\hat{G}$ to be closed in $\hat{G}$. In other words, $E(P_{closed})C = C_{samp}$.

Note that the definition of $\hat{C}$ above is simply $\frac{1}{P_{closed}} C_{samp}$, so $E(\hat{C}) = C$, so $E(\hat{C})$ is an unbiased estimator of C. □

The results above show that the estimates are correct in expectation, but do not provide concentration bounds. We can prove such results that bound the errors in expectations by applying Hoeffding's inequality [10]. While these bounds may be weak for each individual entry, they can provide the theoretical basis for showing that we will be able to identify all large degrees vertices, and the degree of vertex that we predict to have a large degree, will not be too small. We are not including this analysis here due to space constraints, but the principles of Theorem 6.2 in [19], will apply to our application.

8.1.2 Estimations for Random Edge Sample

Estimating degree d_u: In this sampling method, f_E fraction of edges are sampled uniformly at random. Thus, to estimate a node's true degree d_u, MAXOUTPROBE simply multiplies its observed degree d_u^{known} by $\frac{1}{f_E}$. As with the Random Node sampling method, this estimation is inaccurate for low degree nodes, but these nodes are unlikely to be chosen as probes, and so inaccuracy here does not affect the final probe selections.

Claim: The estimator for d_u is unbiased.

Proof: As before, let $\hat{d_u}$ represent the estimator for d_u, and we must show that for each u, $E(\hat{d_u}) = d_u$. $\hat{G}$ consists of f_E of the edges from G selected uniformly at random.

Consider a node u in $\hat{G}$, with true degree d_u. Because f_E of the edges from G were sampled uniformly at random, $E(d_u^{known}) = f_E d_u$. Thus, $E(\frac{1}{f_E} d_u^{known}) = d_u$. This first term is simply $\hat{d_u}$, so $\hat{d_u}$ is an unbiased estimator for d_u. □

Estimating clustering coefficient C: MAXOUTPROBE first calculates the clustering coefficient C_{samp} of $\hat{G}$. Consider a length-2 path (x, y, z) in G that is present in $\hat{G}$. If that path is a closed triangle in G (i.e., x is connected to z), there is a f_E probability that the path will be a closed triangle in $\hat{G}$ (i.e., with f_E probability, edge (x, z) is present in $\hat{G}$). Thus, MAXOUTPROBE estimates $C = \frac{1}{f_E} C_{samp}$.

Claim: The estimator for C is unbiased.

Proof: As before, let $\hat{C}$ represent the estimator for C, and we must show that $E(\hat{C}) = C$. $\hat{G}$ consists of f_E of the edges from G selected uniformly at random.

C, the clustering coefficient of G, is the fraction of wedges (length-2 paths of nodes (x, y, z)) that are closed (x is connected to z). Suppose that edge (x, z) exists in G; then, independently of wedge (x, y, z) being present in $\hat{G}$, there is f_E probability that it is present in $\hat{G}$. Thus, of the wedges from G that are present in $\hat{G}$, we expect that $f_E C$ fraction of these wedges are closed in $\hat{G}$ (C is the probability that it was closed in G, and f_E is the probability that it remained closed in $\hat{G}$). If C_{samp} is the clustering coefficient of $\hat{G}$, then $E(C_{samp}) = f_E C$, so $E(\frac{1}{f_E} C_{samp}) = C$. The first term is $E(\hat{C})$, so the $\hat{C}$ is an unbiased estimator for C. □

The bounds discussed above also apply here.