Enriching Incomplete Networks

Sucheta Soundarajan
Syracuse University
susounda@syr.edu

Tina Eliassi-Rad
Rutgers University
eliassi@cs.rutgers.edu

Brian Gallagher
Lawrence Livermore Lab
bgallagher@llnl.gov

Ali Pinar
Sandia Lab
apinar@sandia.gov

Abstract. We observe that no matter how meticulously constructed, networks are often partially observed and incomplete. This incompleteness can lead to inaccurate findings (e.g., in terms of detecting community structure). We introduce the *Active Edge Probing* problem. Suppose that one is given a sample of a larger network and a budget to learn additional neighbors of nodes within the sample, with the goal of enriching the incomplete network. Which nodes should be further explored? We present *ε-WGX*, a graph-based multi-armed bandit algorithm for identifying which nodes in a sample should be probed. We compare *ε-WGX* to several baseline algorithms on four datasets using samples generated by four sampling methods, and find that although the best baseline strategy (such as probing high degree nodes) varies by network and application, *ε-WGX* consistently outperforms or nearly matches the performance of the best baseline strategy.

Introduction. Most network analyses are conducted on existing incomplete samples of much larger complete networks (e.g., networks collected over a short time period or extracted by a crawling or sampling algorithm). For example, a researcher studying a Twitter retweet network might initially obtain a sample from an online data repository. This data is incomplete. More complete data would lead to more accurate analyses, but data acquisition is costly. Given a query budget for identifying additional edges, which nodes should be further explored (i.e., probed) so that the resulting neighborhood information produces a better representation of the fully observed network? We call this problem the *Active Edge Probing* problem**.** Real applications of this problem abound. Consider a micro-loan company, which gets access to an applicant's Facebook account, and uses this information to make credit decisions about the applicant. The company knows who the applicant's friends are but does not know if they are friends with each other (which can be helpful for evaluating credit scores). In this case, the company wishes to learn about edges that produce triangles between individuals in the sample.

To solve the Active Edge Probing problem, we present *ε-WGX* (short for *epsilon Weighted Graph eXploration*), a graph-based multi-armed bandit strategy for identifying which nodes to probe. The key strength of *ε-WGX* is that it does not require any (1) background information about the structure of the sample, (2) knowledge of how the sample was created, and (3) details about the underlying network G. Compared to baseline probing strategies (such as probing high or low degree nodes), across separate experiments including 4 common sampling methods and 2 reward functions, averaged over 4 real network datasets, *ε-WGX* is the best or second best probing strategy. For example, for the task of maximizing the number of nodes in the sample, on samples generated by a BFS crawl, *ε-WGX* outperforms random probing by 20%. Moreover, *ε-WGX* requires no background information and so can be used if one is unsure which baseline is best. Figure 1 depicts the performance of *ε-WGX* vs. three baseline strategies on a random walk crawl of a student Facebook network, with the goal of observing as many new nodes as possible. *ε-WGX* performs roughly the same as the best baseline strategy without needing any of the aforementioned background information.

Problem Statement. We assume that a network G_{samp} is given, where G_{samp} is a (smaller) sample of a larger network G. We have no further knowledge about G, but can conduct probes on G_{samp} to obtain more information and enrich its structure. Although new nodes may be added to the sample as we conduct probes, we assume that only nodes that were present in the sample from the beginning can be selected for probing. When a node u is probed, we observe one neighbor of u selected uniformly at random from the set S_u, where S_u contains all neighbors of u from G that we have *not* already observed. We are also given a probing budget b, where each unit of budget corresponds to one probe (e.g., running one traceroute). We assume that there is a reward function that guides the selection of probes. For instance, a reward function can be to add as many new nodes as possible to the sample, or close as many triangles as possible in the sample. The problem then is to select which nodes in G_{samp} to probe.

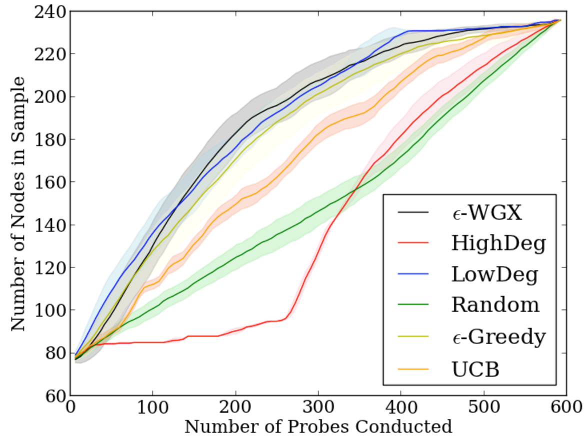

Figure 1: Performance of ε-WGX on a Facebook graduate student friendship network (described later). Twenty trials were performed on the same BFS sample. The x-axis represents the number of probes conducted. This value ranges from 0 (no probes conducted) to approximately 600 (every edge that could be learned has been seen). The y-axis represents the number of nodes present in the sample after conducting some number of probes. A good strategy will add many new nodes to the network after a small number of probes. The shading represents one standard deviation. ε-WGX performs as well as the best baseline strategy without assuming background knowledge such as the underlying structure of the fully observed network.

Background. The multi-armed bandit problem is motivated by a gambler attempting to choose which slot machine (a.k.a. 'bandit') to play from among a set of slot machines. Each machine gives a reward drawn from a distribution specific to that machine, and the goal of the gambler is to maximize this reward. A hallmark of solutions to the multi-armed bandit problem is the trade-off between *exploration* (i.e., selecting a random machine to play), and *exploitation* (i.e., playing a machine that has been known to perform well in the past).

Algorithms for the multi-armed bandit problem abound.[1] Two popular methods are the ε-greedy approach and the UCB (Upper Confidence Bound) algorithm. In the ε-greedy approach, one explores with probability ε, and exploits with probability $1-\varepsilon$ (i.e., selects the arm with the highest average reward so far).[2] The UCB algorithm begins by initially selecting each arm once, and then calculating an upper confidence bound on the expected reward for each arm.[3] In each step, the arm with the greatest upper confidence bound is chosen. If an arm has been selected a small number of times, one cannot calculate a tight bound, and so the upper confidence bound is higher than it would be if the arm had been selected many times. All else being equal, there is thus a preference for arms that have been chosen fewer times.

Proposed Method: We view the Active Edge Probing problem as a version of the general multi-armed bandit problem, where one may probe any of the nodes in the sample network, and depending on the selection, obtain information of varying value for the specified reward function. Because successful probes may exhibit significant structural variance depending on the application, the network, and so on, bandit algorithms are suitable here.

We present *ε-WGX*, a multi-armed bandit approach for the Active Edge Probing problem. When used for the active graph probing problem, *ε-WGX* has the following strengths. (1) It can be used *without background knowledge of the network structure or reward function*. In other words, one need not know what type of node (e.g., high degree) is useful. (2) It can *adapt* to the network and reward function. If probing some node is unsuccessful, it is unlikely to probe that node further. (3) It is *reliable*. We will see that the best baseline probing strategy varies substantially by sampling method, reward function, and network, but regardless of which baseline strategy happens to be the best for a given sample and reward function, *ε-WGX* typically outperforms or matches the performance of the best baseline strategy.

---

1 See https://sites.google.com/site/banditstutorial/ for a nice tutorial on bandits (given at ICML 2011).
2 C.J.C.H. Watkins. "Learning from Delayed Rewards." Ph.D. Dissertation, Kings' College, 1989.
3 P. Auer, N. Cesa-Bianchi, and P. Fischer. "Finite-time analysis of the multi-armed bandit problem." *Machine Learning*, 47(2-3):235-256, 2002.

Unlike other multi-armed bandit algorithms, *ε-WGX* takes advantage of the graph structure inherent in the problem. We make the following key observation: When we probe a node and obtain an edge, we have gained information *not just about that node, but also about the observed neighbor*. In a typical multi-armed bandit problem setting, such as advertising, when one selects an arm, one obtains reward data about only that arm. Some methods have been developed for cases in which arms have linked rewards (e.g., if a user frequently clicks on an ad about boats, we might infer a higher reward for other ads about boats). However, we are not aware of algorithms that use this type of graph structure, in which a reward for one arm directly translates into a reward for another arm, but the rewards of those two arms are linked only once, when the corresponding edge is observed.

Reward Function: *ε-WGX* requires that the user specify a reward function, which is used to evaluate each probe. In the present work, we consider three reward functions. The first one maximizes the number of nodes in the sample. If a probe results in an edge leading outside of the sample, the reward value is 1. Otherwise, the reward value is 0. The second reward function aims to close triangles within the sample. If a probe results in an edge between nodes that are two steps away in G_{samp}, then the reward is the number of triangles created by the addition of that edge. Otherwise, the reward is 0. The third reward function aims to connect nodes in the sample. If a probe results in an edge between nodes that are both within G_{samp}, then the reward is 1. Otherwise, it is 0.

Observed Rewards: For each node u in G_{samp}, *ε-WGX* maintains a vector R_u, containing a list of rewards corresponding to the probes conducted on that node. Note that the mean of this vector is an unbiased estimate of the true mean reward for node u, because each probe returns a uniformly random edge adjacent to u. Thus, the mean of R_u is an unbiased estimator of the true mean reward of node u.

Additionally, we maintain a vector R'_u, corresponding to the rewards observed when we probed some other node v and obtained edge (v, u). This vector represents the key difference between *ε-WGX* and existing multi-armed bandit algorithms. When *ε-WGX* probes a node, it can update not only the set of rewards corresponding to that node, but also the set of rewards corresponding to the observed neighbor.

Unlike R_u, the mean of R'_u is *not* an unbiased estimator of the true mean reward of u. To see this, consider the first reward function described above (in which we receive a score of 1 if the observed edge connects to a node outside of the original sample, and a score of 0 otherwise). Because we only probe nodes that were present in G_{samp} from the beginning (and thus only maintain reward vectors for these nodes), the R'_u vector can only contain 0s!

Node Selection: *ε-WGX* requires an exploration parameter ε between 0 and 1. In each step, with probability $1-\varepsilon$, *ε-WGX* selects the highest scoring node, and with probability ε, it selects a random node using the reward vectors in accordance with the following rules:

1. If the reward function is such that R'_u can contain both 0s and 1s,[4] then we assign each node u a weight equal to the mean of the values in R_u and R'_u, plus a small constant to allow nodes with a mean of 0 to have positive weight (e.g., we use 0.1). For nodes u such that both R_u and R'_u are empty, we assign a weight equal to the average of the weights with non-empty reward vectors. In other words, if we have no information about a node, we treat it as equal to the average node for which we do have information. We then select a node with probability proportional to its weight. This node is then probed.

2. If the reward function is such that R'_u can have only 0s or only 1s,[5] then we follow the same procedure as above, except that we generate the weights by only looking at R_u, rather than both R_u and R'_u. If we select a node that has previously been probed, then we probe that node. Otherwise, we reconsider the set S of all nodes that have not yet been probed. From S, we select the node with the 'best' R'_u vector. That is, we select the node u with the R'_u vector with the most 1s or the fewest 0s (recall that in this case, R'_u contains only 0s or only 1s). In other words, the biased R'_u vector is used as a tiebreaker for nodes lacking unbiased information. The selected node is then probed.

[4] The triadic closure reward function satisfies this case.
[5] The reward functions incentivizing outgoing edges or internal edges satisfy this case.

After probing a node u, *ε-WGX* updates the reward vector R_u with the observed reward from that probe. If the observed edge connected u to another node v within the sample, then we update R'_v: if the observed reward was 0, then we add a 0 to R'_v. Otherwise, we add a 1 to R'_v. If u has no further neighbors to be learned, *ε-WGX* removes it from future consideration.

Empirically, we found a value of $\varepsilon = 0.3$ to work well or the trade-off between exploration and exploitation. A formal (i.e., model selection) method for picking ε is part of our future work.

Data and Experimental Setup. We consider four networks: FB-Grad and FB-Ugrad, which are portions of the Facebook social network corresponding to graduate and undergraduate students at Rice University;[6] FB-SocCir, which consists of social circles in Facebook;[7] and Amazon, which is a product co-purchasing network.[8] For validation purposes, we assume these networks are fully observed and produce samples using BFS crawls, random edge sampling, and random walks with and without jump. Our samples contain 10% of the total number of edges from the original network.

We compare *ε-WGX* to three simple baseline strategies: probing high degree nodes in order (*HighDeg*), probing low degree nodes in order (*LowDeg*), and probing nodes in a random order (*Random*). We also compare *ε-WGX* against the *ε*-greedy and UCB multi-armed bandit strategies.

Experimental Results. First, to motivate the need for a multi-armed bandit strategy over a simpler baseline strategy, we demonstrate that the best baseline strategy varies substantially depending on the particular network, sample type, and the reward function being considered. Thus, one cannot simply select a simple strategy that was successful in the past. Table 1 lists the baseline strategies that were best for various networks and sampling combinations, when the reward was to bring in as many new nodes as possible to the sample. Similar results hold for other reward functions. There is little consistency within rows or columns. For example, *HighDeg* is successful in one case, but *LowDeg* is successful for a different case.

Sample Type	FB-Grad Network	FB-Ugrad Network	FB-SocCir Network	Amazon Network
BFS	*LowDeg*	*LowDeg*	Varies	*LowDeg*
RandEdge	*LowDeg*	*LowDeg*	*HighDeg*	*HighDeg*
RW	Varies	Varies	Varies	*HighDeg*
RWJ	Varies	Varies	*HighDeg*	*HighDeg*

Table 1: The most successful baseline strategy for various networks and sampling methods, with the reward of maximizing the number of nodes brought into the sample. "Varies" indicates that for samples of the same type, different baseline strategies were most successful. There is little consistency in terms of which strategy is the best.

Figure 2 depicts the performance of *ε-WGX*, the baseline strategies, and the other bandit-based strategies over all networks and sampling methods, for all 3 reward functions, presented as fraction improvement over random probing.

When the reward is to maximize the number of nodes in the sampled incomplete network, *ε-WGX* is the best strategy for random edge and random walk samples. On BFS samples, the *LowDeg* strategy is the best, but the three bandit strategies all perform substantially better than random. When the reward is to close triangles in the sample, *ε-WGX* has the best performance for random walks with jumps, and is a close second on the other three sampling methods. The other bandit-based strategies also do well. For the goal of adding edges within the sample, *ε-WGX* is the clear winner in all cases.

We reach two important conclusions: First, the bandit-based strategies (*ε-WGX*, *ε*-greedy, and UCB) perform much better than random probing. Second, in almost every case, *ε-WGX* exceeds or matches the performance of the best baseline strategy, regardless of what that strategy might be.

[6] Obtained from Alan Mislove.
[7] https://snap.stanford.edu/data/egonets-Facebook.html
[8] https://snap.stanford.edu/data/#amazon

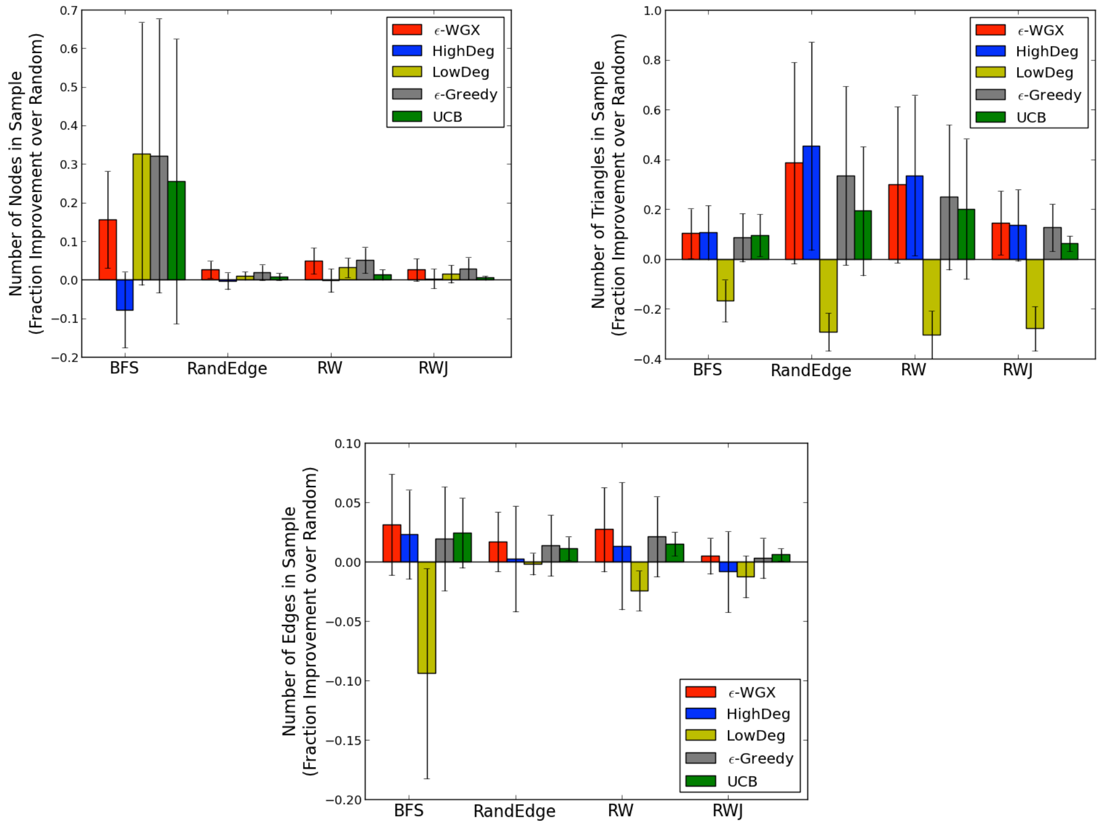

Figure 2: Performance of ε-WGX as compared to the HighDeg, LowDeg, ε-Greedy, and UCB baseline probing strategies as measured by the fraction improvement over random probing, for maximizing the number of nodes in the sample (top left), closing triangles in the sample (top right), and connecting nodes in the sample (bottom). Results are averaged over all budgets over all networks. Note that the three bandit strategies (ε-WGX, ε-Greedy, and UCB) tend to perform well. That is, they often match or exceed the performance of the best baseline strategy, regardless of what that strategy happens to be. ε-WGX in particular performs well.

Conclusions. We presented the *Active Edge Probing* problem, in which one is given an incomplete sample of a larger network, and is allowed to conduct a fixed number of probes to enrich the network's structure. This problem is important because the phenomena generating the networked data are often partially observed. We presented a multi-armed bandit solution, called *ε-WGX*, as a solution to this problem. "Good" probes can vary depending on the characteristics of the underlying system that generated the data, how the sample was collected, and the reward function. *ε-WGX* does not make *a priori* assumptions about these. As our experiments demonstrate, *ε-WGX* matches or exceeds the performance of the best baseline strategy across several networks, sampling methods, and reward functions.

Acknowledgements. Brian Gallagher and Sucheta Soundarajan performed this work under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. Ali Pinar is supported by the DARPA GRAPHS program. 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. Tina Eliassi-Rad was supported in part by NSF CNS-1314603, by DTRA HDTRA1-10-1-0120, and by DAPRA under SMISC Program Agreement No. W911NF-12-C-0028.