Predicted Max Degree Sampling : Sampling in Directed Networks to Maximize Node Coverage through Crawling

Ricky Laishram, Katchaguy Areekijseree, Sucheta Soundarajan
Syracuse University, NY 13244-4100
Email: {rlaishra, kareekij, susounda}@syr.edu

***Abstract*—Sampling through crawling is an important research topic in social network analysis. However there is very little existing work on sampling through crawling in directed networks. In this paper we present a new method of sampling a directed network, with the objective of maximizing the node coverage. Our proposed method, *Predicted Max Degree (PMD) Sampling*, works by predicting which k open nodes are most likely to have the highest number of unobserved neighbors in a particular iteration. These nodes are queried, and the whole process repeats until all the available budget has been used up. We compared PMD against three baseline algorithms with three networks, and saw large improvements vs. baseline sampling algorithms: With a budget of 2000, PMD found 15%, 87.4% and 170.2% more nodes than the closest baseline algorithm in the wiki-Votes, soc-Slashdot and web-Google networks respectively.**

1. Introduction

Sampling huge networks is a topic of great interest in the field of complex network analysis because of the computational and exploration cost involved. Depending on initial conditions, there are two types of sampling - representative subgraph sampling and unbiased sampling [1].

In this paper, the type of sampling we are considering is *sampling through crawling*. When sampling through crawling, we start with a small sub-graph, and the complete network is initially unknown. Information about the network is obtained only through querying the neighbors of the observed nodes. Sampling through crawling is a relatively new research area. It is increasingly becoming very important because crawling is the only way to fetch information in a lot of networks. For example, to sample the web or various online social networks, crawling is the only method.

In our algorithm, *Predicted Max Degree (PMD)* sampling, we use closed nodes (i.e., previously-queried nodes) to predict the open nodes (i.e., observed but unqueried nodes) with the highest unseen in/out neighbors. Then, the appropriate queries are performed on these nodes. The whole process continues until the all the budget has been used up. We describe this in more details in Section 4.

The main contributions of this paper are:

1) We present a novel method of sampling a directed network through crawling that maximizes node coverage with a given budget. Our method outperforms baseline methods by up to 170%.

TABLE 1. THE CORRELATION BETWEEN IN-DEGREE AND OUT-DEGREE FOR DIFFERENT NETWORKS.

Top %	Wiki-Votes	Twitter-Friends	Web-Stanford
10	-0.07	0.04	-0.01
20	0.08	0.19	0.01
50	0.24	0.36	0.03
100	0.31	0.43	0.04

2) We provide a discussion of why the existing methods developed for undirected networks does not work very well when applied to directed networks.

2. Related Works

There are many existing works are on representative subgraph sampling. In representative subgraph sampling, the complete graph is known and the goal is to obtain a subgraph that represents some property of the original graph. However, this type of sampling is not the focus of our paper.

There are also quite a few recent works on sampling a network through crawling. The most commonly used sampling algorithms are Breadth First Search and Random Walk, and variants of these.

Ye et al. [2] investigated the performance of some common sampling algorithms on online social networks. They evaluated the performance of the different algorithms based on the node coverage and edge coverage.

Avrachenkov et al. [3], proposed an algorithm called Maximum Expected Uncovered Degree (MEUD) for maximizing the node coverage within a given budget in an undirected network. In MEUD they assume that the degree distribution of the original graph G is known. If the degree distribution of G is not known, MEUD reduces to an algorithm they refer to as Maximum Observed Degree (MOD).

The works mentioned here on undirected networks. We could not find any work that addresses the problem of maximizing the node coverage in a directed network.

3. Problem Definition

Assume that there is a directed network $G = \langle V, E \rangle$, and $G_t^* = \langle V_t^*, E_t^* \rangle$ is the sub-graph after t queries. We assume that G can be explored only through crawling and we are limited to B queries. Our goal is to develop an algorithm that will maximize $|V_B^*|$.

The inputs to PMD are the initial sub-graph G_0^* (this can be constructed by a small BFS or random walk crawl, or might consist of only one node) and the maximum number of queries it can perform B, referred to as budget. After it has used up all the budget, the algorithm outputs the final sub-graph G_B^* as the sample.

For a node $v \in O_t$, there are two type of queries we can perform - the in-neighbors query $\Gamma_i(v)$ and the out-neighbors query $\Gamma_o(v)$. Each of these queries consumes one unit of the budget, and we can perform neither, one, or both queries on an observed node.

3.1. Challenges

In our problem the network is directed, and this presents challenges not found in undirected networks.

The first challenge is that we need to consider the in-neighbors and out-neighbors of a node separately. Since each of these queries consumes one unit of budget each, we need a way to decide if we want to query the in-neighbors, out-neighbors or both.

The second challenge is that in most real networks, there is very little correlation between the in-degree and out-degree of the high degree nodes. Because our objective is to maximize the node coverage, the nodes with high degree (either in or out) are most useful. From Table 1, it can be observed that the correlation between the in-degree and out-degree is non-existent for these nodes. So, that means that if we have a closed node on which we have queried for the in-neighbors, we cannot say if we should query for its out-neighbors as well or not based on only this information.

4. Methodology

In this paper, we refer to the set of nodes for which we know all the in-neighbors or out-neighbors as the In-Closed Nodes C_t^i and Out-Closed Nodes C_t^o respectively. We define the Closed Nodes C_t as, $C_t = C_t^i \cup C_t^o$.

The set of nodes that are in the sub-graph G_t^* but not in C_t are referred to as Open Nodes (O_t). That is, $O_t = V_t^* \setminus C_t$. These are nodes that have been observed as a neighbor of a queried node, but have themselves not been queried.

At a high level, PMD works by selecting the k best nodes from O_t that are most likely to have the highest number of unobserved in/out degree, and the appropriate query for each of them. Then after querying these nodes, O_{t+b}, C_{t+b}^i and C_{t+b}^o are updated (here b is the budget used so far). This process continues until all the budget B has been used up.

Our proposed algorithm has two main components - *BestNodes* (Algorithm 1) and *QueryNodes*(Algorithm 2). *BestNodes* is responsible for selecting the set of best open nodes to query on and the type of query to perform on each node. *QueryNodes* in the function that actually performs the queries and updates the parameters k and ϕ.

The sample size s in Algorithm 2:1 is calculated such that if a sample S^* of size s is selected from C_t, the probability that a node $n \in O_t$ with in/out degree $d_* \geq d_\phi$ has an in/out neighbor in S^* is at least p. This value of s can be calculated by solving the inequality 1. (Proof is not included due to page limits.)

Algorithm 1 BestNodes

```
Input:
    $C$ : The set of closed nodes
    $O$ : The set of open nodes
    $p$ : The threshold accuracy
    $d_\phi$ : The threshold degree
    $k$ : The number of nodes to return
Output:
    The set of candidate nodes and type of query
    Initialization :
    $N \leftarrow set()$
    $score \leftarrow HashTable()$
 1: $s \leftarrow SampleSize(C, d_\phi, p)$
 2: $S^* \leftarrow RandomSample(C, s)$
 3: for $v \in S^*$ do
 4:     for $u \in \Gamma_i(v) \cap O$ do
 5:         $score[(u,'o')] \leftarrow score[(u,'o')] + 1$
 6:     end for
 7:     for $u \in \Gamma_o(v) \cap O$ do
 8:         $score[(u,'i')] \leftarrow score[(u,'i')] + 1$
 9:     end for
10: end for
11: for $i \in \{1, 2, \ldots, k\}$ do
12:     $key, value \leftarrow max(score)$
13:     $score[key] \leftarrow -1$
14:     $N \leftarrow N \cup \{key\}$
15: end for
16: return $N$
```

$$\prod_{i=1}^{d_\phi}(|C_t| + 1 - s - i) \leq (1-p) \cdot \prod_{i=0}^{d_\phi}(|C_t| + 1 - i) \tag{1}$$
$$s \in \mathbb{Z}_+$$

The accuracy in Algorithm 2:10 is defined by Equation 2, and it is used to determine the value of k for the next iteration. If the accuracy is above the threshold p, the value of k is incremented, otherwise the value of k is decreased.

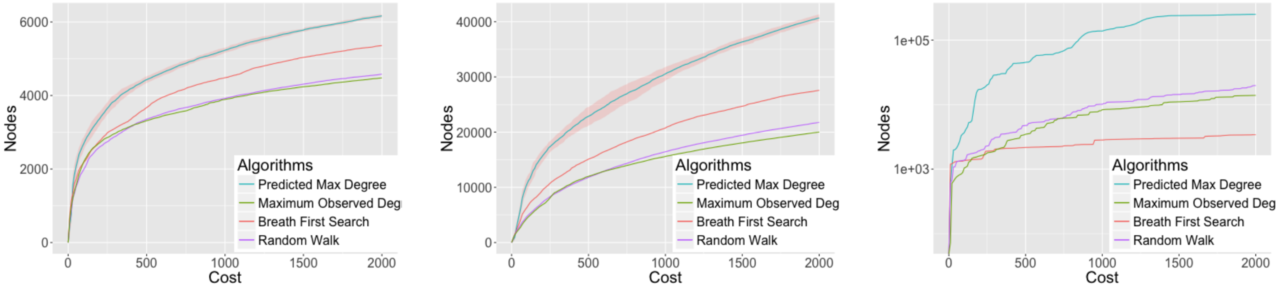
If there are not enough nodes with degree greater than d_ϕ left, the accuracy remains below p even after adjusting k. So if a fails to increase even after adjusting k, we ϕ decreased by 5% and d_ϕ is recalculated.

$$a = \frac{|\{(v,\tau) \in N \colon degree_\tau(v) \geq d_\phi\}|}{|N|} \tag{2}$$

5. Experimental Setup

In this section we describe the baseline algorithms that we compare to PMD. We also describe the dataset we use for our experiments.

The baseline algorithms are MOD [3] (the current state-of-the-art for maximizing node coverage in undirected networks), BFS, and Random Walk. For each of these baseline algorithms, we consider three different versions - one that follows only in-edges, out-edges, and both.



(a) Wiki-Votes (b) Soc-Slashdot (c) Web-Google

Figure 1. The comparison between PMD and the baseline algorithms. The line is the mean of 10 experiments and the shaded area is the standard deviation. The standard deviations of the baselines are not shown to make the figures less cluttered.

Algorithm 2 QueryNodes

```
Input:
    $B$ : The budget available
    $p$ : The threshold accuracy
    $\phi$ : The threshold percentile
    $k$ : The number of nodes to return
    $G^*$ : The initial sub-graph
Output:
    The sample network $G^*$
while $cost \leq B$ do
    $d_\phi \leftarrow PercentileDegree(C, \phi)$
    $N \leftarrow BestNodes(C, O, p, d_\phi, k)$
    for $(v, \tau) \in N$ do
        $\gamma \leftarrow \Gamma_\tau(v) \setminus (O \cup C)$
        $O \leftarrow O \cup \gamma$
        $O \leftarrow O \setminus \{v\}$
        $C^\tau \leftarrow C^\tau \cup \{v\}$
    end for
    $a \leftarrow UpdateAccuracy(N, p)$
    $k \leftarrow UpdateK(a, k)$
    $\phi \leftarrow UpdateThershold(a)$
    $cost \leftarrow UpdateCost()$
end while
return $G^*$
```

We use the wiki-Vote, soc-Slashdot0922 and web-Google datasets from SNAP[1], representing different types and sizes of networks.

The initial values of the parameters for all experiments are $p = 0.9$ and $\phi = 90$. The inital sub-graph is generated through 20 steps of BFS.

6. Experimental Results

In this section we will present the results of PMD sampling against the baseline algorithms described in Section 5. Recall that the objective of PMD sampling is to maximize the node coverage, or fraction of nodes that have been observed. But since the total number of nodes in the complete graph G is a constant, it is sufficient to compare only the number of observed nodes, $|O_B \cup C_B|$.

For each of the datasets in Section 5, we select 10 nodes randomly as seed nodes, and the budget is set to 2000. Figure 1 shows comparison of the number of observed nodes between PMD and the baselines against the cost. The lines represent the mean of the 10 trials and the shading is the standard deviation. For the baseline algorithms, only the best variant of each method is shown. For the wiki-Votes and web-Google networks, the variant that follows both in and out neighbors is best. For soc-Slashdot, the variant that follows only the out-neighbor is best.

It can be observed from Figure 1 that PMD out-performs all the baseline algorithms for all the budgets. In the case of the wiki-Votes network after 2000 queries, PMD was able to obtain 15% more nodes than the nearest baseline. However, in the larger networks, this difference is much higher. In soc-Slashdot network, the node coverage of PMD after 2000 queries is 87.4% higher than the nearest baseline, and in web-Google network, the difference is 170.2%.

7. Conclusion

In this paper, we presented Predicted Max Degree (PMD) Sampling, a novel algorithm to maximize node coverage. By comparing PMD with different baseline algorithms for different datasets, we showed that the algorithm can obtain samples with significantly higher node coverage.

Although these results are promising, there is still much work to do in this area. For example, in many networks, there are limits to the number of nodes that can be obtained per query (eg. Twitter has the API rate limit). A research direction we are currently pursuing is to explore ways to handle this additional constrain.

References

[1] A. S. Maiya and T. Y. Berger-Wolf, “Benefits of bias: Towards better characterization of network sampling,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 105–113, ACM, 2011.

[2] S. Ye, J. Lang, and F. Wu, “Crawling online social graphs,” in *Web Conference (APWEB), 2010 12th International Asia-Pacific*, pp. 236–242, IEEE, 2010.

[3] K. Avrachenkov, P. Basu, G. Neglia, B. Ribeiro, and D. Towsley, “Pay few, influence most: Online myopic network covering,” in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*, pp. 813–818, IEEE, 2014.

[1] http://snap.stanford.edu/data