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

Ricky Laishram
Katchaguy Areekijseree, Sucheta Soundarajan

Department of Electrical Engineering & Computer Science
Syracuse University

May 1, 2017

Introduction

- Sampling networks is important to obtain a smaller representative sample, or to collect data.
- **Sampling through crawling**: A small subgraph is initially known, and new nodes are discovered by querying for neighbors of observed nodes.
- Lots of works on sampling through crawling in undirected networks. Example: [Avrachenkov et al., 2014]
- Very few works on directed networks.

Challenges

- For each node, we need to decide if we should perform in-neighbors or out-neighbors query, or both.
- There is very little correlation between in-degree and out-degree of the high degree nodes in real world networks.
- In many real world cases, there are limits on the number of nodes returned for a query.

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

Table: Correlation between in-degree and out-degree

Problem Definition

Objective

Given a directed network $G = \langle V, E \rangle$ that can only be explored through crawling, obtain the sample $G_B^* = \langle V_B^*, E_B^* \rangle$ by querying B nodes such that the $|V_B^*|$ is maximized.

Problem Settings

Two type of queries on a node $u \in V_t^*$:

- In-query, $\gamma_x^i(u)$
- Out-query, $\gamma_x^o(u)$

A query on a node $u \in V_t^*$ return,

- all the neighbors. (Crawling without limits)
- at most m neighbors. (Crawling with limits)

Proposed algorithms

- Crawling without limits: Predicted Max Degree Sampling (PMD)
- Crawling with limits: Predicted Max Degree Sampling with Limits (PMDL)

Notation and Terminologies I

- $\Gamma^{\tau}(u)$: τ-neighbor of node u.
- $\gamma_x^{\tau}(u)$: Nodes returned on the x^{th} τ-neighbors query of node u.
  - In the case of crawling without limits, $\gamma_x^{\tau}(u) = \gamma_{x+1}^{\tau}(u)$.
- m: The maximum number of nodes returned on a single neighbor query.
  - For crawling with limits, $\max_{u \in V_x^*, x \in \mathbb{Z}} |\gamma_x^{\tau}(u)| \leq m$.
- $d^{\tau}(u)$: The τ-degree of a node u.

Notation and Terminologies II

Closed Nodes: Set of nodes on which at least one query has been made. (C_t)

If the query made is,

- in-neighbors: In-Closed Nodes (C_t^i)
- out-neighbors: Out-Closed Nodes (C_t^o)

Closed Nodes, $C_t = C_t^i \cup C_t^o$

Notation and Terminologies III

Open Nodes: Set of nodes on which has at least one type of query remaining. (O_t)
If the query remaining is,

- in-neighbors: In-Open Nodes (O_t^i)
- out-neighbors: Out-Open Nodes (O_t^o)

τ-Open Nodes, $O_t^\tau = V_t \setminus C_t^\tau$
Open Nodes, $O_t = O_t^i \cup O_t^o$

Predicted Max Degree Sampling

- For the case of crawling without limits.
- Select k nodes from O_t with the highest expected number of unobserved in/out degree.
- These nodes are selected by performing in and out queries on a random sample of size s from C_t.
- Open nodes that are observed frequently during this step are more likely to have higher in/out-neighbors.
- The algorithm consist of two components:
  - QueryNodes
  - BestNodes

Perform the appropriate queries on the nodes found by **BestNodes** and update the parameters.
The accuracy a is given by,

$$a = \frac{|\{(u, \tau) \in N\colon d^{\tau}(u) \geq d_{\phi}\}|}{|N|}$$

If $a \geq p$, the value of k is incremented. Otherwise decremented.
If a remains below p even after adjusting k, decrease ϕ.
The budget b_1 used in this step is $b_1 = k$.

QueryNodes II

Algorithm 1 QueryNodes Algorithm

procedure QueryNodes
    while $cost \leq B$ **do**
        $d_\phi \leftarrow \phi$ percentile degree from C
        $N \leftarrow BestNodes(C, O, p, d_\phi, k)$
        for $(u, \tau) \in N$ **do**
            Perfom τ query on u
            Update O and C
        end for
        Update p, k, ϕ and $cost$
    end while
end procedure

BestNodes I

Objective

Find set $N \subseteq O_t \times \{i, o\}$, such that

- $|N| = k$
- $|\{(u, \tau) : (u, \tau) \in N \land d^\tau(u) \geq d_\phi\}| \geq p \cdot |N|$
- Minimize b the amount of budget consumed.

BestNodes II

Algorithm 2 BestNodes Algorithm

procedure BestNodes
    $s \leftarrow$ Compute sample size
    $S^* \leftarrow$ Randomly select s nodes from C
    for $v \in S^*$ **do**
        Increment score of (u, i) for $u \in \gamma^o(v) \cap O$
        Increment score of (u, o) for $u \in \gamma^i(v) \cap O$
    end for
    $N \leftarrow$ Select k (u, τ) pairs with highest scores
end procedure

BestNodes III

The budget b_2 used in this step is,

$$b_2 = |S^* \setminus C_t^o| + |S^* \setminus C_t^i|$$

Since $\forall u \in S^*$, $u \in C_t^o$ or $u \in C_t^i$,

$$b_2 \leq s$$

The sample size s is given by,

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

Results: PMD

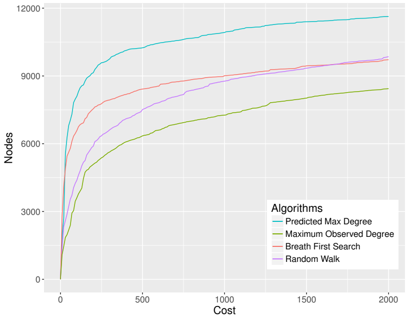

Figure: Node coverage on Twitter dataset

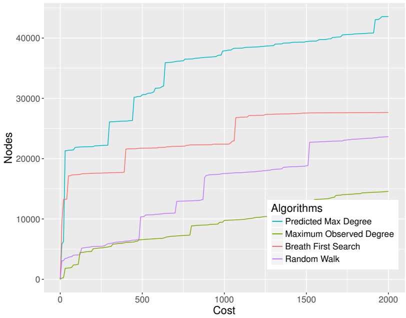

Figure: Node coverage on Web-Stanford dataset

Predicted Max Degree Sampling with Limits

Sampling algorithm for the case of crawling with limits.
Define a network model such that:

- Every node u is made up of an ordered list of sub-nodes, $[u'_1, u'_2, \ldots]$.
- All sub-nodes except the last one has m neighbors.
- The number of sub-nodes is not known without going through the entire list.

Modifications to BestNodes Algorithm

We need to make modification to the scoring function in **BestNodes**.

- $E^{\tau}(S, u)$ is the set of edges from S^* to a node $u \in O_t$
- Node u has been queried i times.

The set of already observed neighors of u is,

$$\bigcup_{x=1}^{i} \gamma_x^{\not\tau}(u)$$

The $\not\tau$-score of u is given by,

$$score(u, \not\tau) = |E^{\tau}(S, u) \setminus \bigcup_{x=1}^{i} \gamma_x^{\not\tau}(u)|$$

Observations

- If B is "small" compared to the d_{avg}, *PMDL* will offer no significant improvement over naive algorithms.
- The fraction of highest degree nodes to query on completely until κ fraction of of the queries become sub-optimal is,

$$f \geq \left(\frac{\kappa(\alpha - 1)d_{min}}{m(\alpha - 2)(1 - \kappa)} \right)^{\alpha - 1}$$

Results: PMDL

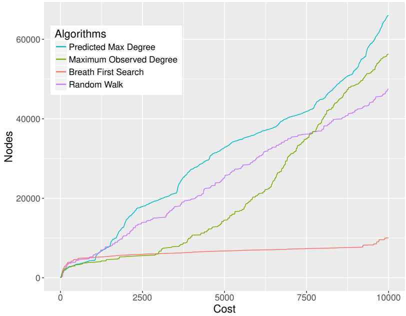

Figure: Node coverage on Web-Google dataset

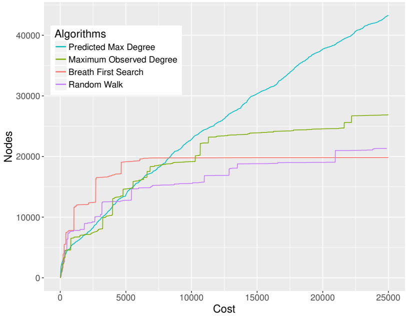

Figure: Node coverage on Web-Stanford dataset

Conclusion

- We examined the problem of sampling a directed network though crawling to maximize node coverage.
- We looked at two problem settings - *Crawling without limits* and *Crawling with limits*.
- We proposed two algorithms - *PMD* and *PMDL* for these two problem settings.
- We tested our algorithms against real world networks, and we achieved improvement of 15% to 170% over the closest baseline.

Thank You

Bibliography I

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