**Predictive Sampling Method for Spread Models in Networks**

Caijun Qin

[cqin@ufl.edu](mailto:cqin@ufl.edu)

*Instructor Info*

College of Liberal Arts & Sciences

University of Florida

Gainesville, FL 32611

**Abstract**

In this paper, we propose a novel network sampling algorithm suitable for creating spread models in (undirected) complex networks. The method induces a subgraph primarily built from chosen nodes while keeping any edges with both endpoints being sampled nodes. Two specific improvements over similar existing algorithms have been developed: increasing the visit frequency of high-degree nodes per n nodes sampled and preventing the distribution of shortest distances from sampled nodes to the start node to become too concentrated.

**Keywords**

spread model, network sampling, caterpillar tree

**1. Introduction**

Complex networks with abundant community structures have far-reaching applications in modeling real world systems such as social media circles, food webs, interstate trade, and the World Wide Web. As opposed to simpler and more regular graphs (lattices and tessellation-based graphs), networks used to model complex systems in our world can have high variations in edge density, node density, and community structure sizes across the entire network. Depending on the purpose, weights and attribute values may be attached to nodes and edges to represent application-specific semantic values. In many real networks, a higher degree for a node often indicates more importance in value as the degree of a node directly proports to how many connections to neighboring nodes it contains.

There are various metrics that can attest the quality of a network sampling algorithm in the context of building a spread model in a network. This paper will compare the performance of the proposed method with several related sampling algorithms, some well-established and some novel, by analyzing data with two metrics. One important metric is the frequency or hit rate of high-degree nodes. What exact degree value is considered high is subjective and dependent on the intended purpose of the sampling. Per *n* nodes collected overall by a sampling algorithm, the degree distribution of a sample (induced subgraph) visualizes this metric veraciously without being constrained by the exact boundary between high and low degrees.

To generate a spread model, another indicator of the usefulness, or more precisely robustness, of a sampling algorithm is the variance of distance of sampled nodes from the start node. The term *distance* between nodes *u* and *v* use here, also applicant to most graph theory contexts, is the number of edges forming any shortest path from *u* to *v*. Centers of the distance distribution of a sample such as median and mean may reveal the effectiveness of the sampling algorithm, but each has major drawbacks in a spread model. The mean in a network sample is highly influenced by outliers in the sampled node set, which are nodes with a distance to the start node larger than most distances. Furthermore, both the mean and mode do not accurately describe the shape of the distance distribution, which is directly determines the architecture of the spread model. The variance of distances can achieve this purpose.

**2. Related Work**

Several network sampling algorithms were compared through the two metrics stated in the introduction: center (mean and median) of the degree distribution and variance of the distance distribution. Previously developed sampling algorithms are quite recent, mostly from the last decade. These sampling algorithms are summarized in the rest of Section 2 and analyzed for advantages and drawbacks in creating a spread model inside a network.

**2.1 Random Walk Sampling**

Using a simple random walk (SRW) to traverse a network is one of the least computationally expensive and simplest traversal methods in a network. From the current visited node *v*, choose any adjacent node with uniform probability. The main concern with SRW derives from the lack of any optimization criteria or rules governing how the next node to visit is chosen. However, SRW naturally predisposes bias towards higher degree nodes.

**2.2 Common Neighbor Aware Random Walk Sampling**

A variant of random walk samplers, the common neighbor aware random walk (CNARW) focuses on reaching nodes less likely to be visited by the conventional SRW method. For each node *v* visited in the previous iteration, a random floating point number *q* () is created. Another value acts as a threshold, where is the number of common neighbors between nodes *u* and *v* [2]. A common neighbor *w* to nodes *u, v* is simply another node with an edge connected to *u* and *v* [2]. The developed strategy helps the walker to traverse farther away from the start node *s* at a faster rate than SRW. This makes the distance distribution of the sample less likely to be concentrated around a short distance, but the algorithm also shifts sampling bias towards lower degree nodes and away from higher degree nodes.

**2.3 Snowball Sampling**

This is an iterative sampling method that expands outward with a new layer of visited nodes per iteration. For some fixed positive integer *k*, each node *v* from the previous iteration chooses *k* adjacent, unvisited nodes [3]. By default, a random subset of size *k* is chosen from the neighborhood following a uniform probability distribution. Some serious drawbacks occur due to having a fixed parameter value. In densely connected networks, having a *k* too small will make searching for the highest degree nodes within unlikely. Likewise, having *k* to large will be computationally expensive and choke at low-degree nodes, meaning that all adjacent nodes are exhaustively visited when . Traversing a network with a high variance of degree introduces both variations of the problem [3].

**2.4 Community Structure Expansion Sampling**

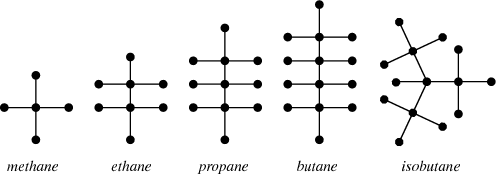
Although SRW is inherently biased towards sampling higher degree nodes, another sampling algorithm, namely community structure expansion sampling (CSES), explicitly targets higher degree nodes to expand the sample outwards from the current set of visited nodes. The central drive of the algorithm is the expansion factor , where is the current set of sampled nodes and *v* is the unvisited node adjacent to some visited node that can bring the maximum number of new unvisited nodes in its neighborhood [4]. The algorithm not only strives to expand the sample at a maximal rate but also target high degree nodes in each iteration, making CSES seem to excel in both metrics mentioned in Introduction.

**2.5 Frontier Sampling**

Frontier Sampling can be viewed as a group of *m* dependent SRW traversal operations [5]. Suppose that queue *L* contains *m* nodes in *G*, the original network to be sampled, symbolically . Select one of the queued nodes with probability , where *u* is any node adjacent to any node in . With chosen, replace with *u,* and repeat the process [5].

**3. Proposed Method**

**3.1 Definitions**

**caterpillar tree** - a tree such that any vertex is within 1 edge of a central path

**3.2 Overview**

Our proposed sampling algorithm, from a bird’s eye view, is a tree expander algorithm initially only including the start node *s*. Let the original network to be sampled be *G* with a node set and edge set . As an iterative sampling algorithm that renders a new layer of visited nodes per iteration, a sub-algorithm is operated on each node visited during the previous iteration. Similar in snowball sampling, only a subset of unvisited nodes adjacent to a visited node *v* (from the most recent iteration) become visited in the current iteration. We say a node *u* is visited once *u* is part of the sample (helping to grow the tree), symbolically . Let denote the neighborhood of *v*, which is the set of nodes adjacent to *v*. However, the number of nodes to be visited next may vary. Rather than have a fixed number of neighbors be visited, a fixed proportion () of nodes in each become the start of a new caterpillar tree that offshoots from *v*. The proportion *q1* captures all nodes in the neighborhood that fall at or above the percentile in the degree distribution of . Let such a set of nodes be . A second proportion () sets a lower threshold to entry, marking the percentile boundary on the degree distribution of . All adjacent nodes of *v* at or above the top but below the are visited only once, and no further execution of the sub-algorithm may traverse any deeper from these nodes. Such dead-ends are analogous to the endpoints one edge away from the central path of a caterpillar tree. Let such a set of nodes be .

**3.3 Caterpillar Quota Walk Sampling Algorithm**

|  |
| --- |
| Let *S* denote the sample of nodes so far.  Let *s* denote the start node.  Let *q1* and *q2* denote the two quotas.        FOR *v* IN L1:  Derive set *Q1* for *v.*  FOR *x* IN *Q1*:      Create set *Q2* for *v.*  FOR *y* IN *Q2*:    UNTIL STOP CRITERIA. |

**4. Experimental Methodology**

**4.1 Data**

Data will be initially generated as random graphs from the NetworkX library, preferably with at least 1000 nodes to reflect a small dataset.

*Notes*: Real-life datasets with less than 1 M nodes and 1 M edges may be attempted. An edgelist of 1.3 M nodes and 3.8 M edges was recently uploading, but available computing resources at the time (my computer) was ineffective in handling the amount of load.

**4.2 Measurements**

**4.2.1 Degree Distribution**

Across each of the 6 network sampling algorithms, including the proposed method, the degree distribution (DegD) will be compared after sampling the prepared network. Comparisons will be done pairwise, rendering a 6 x 6 facet grid with each plot being a quantile plot of the DegD between two sampling algorithms. To draw clear conclusions, the DegD that is more left skewed (distribution mean and median are larger) performs better at capturing important nodes. As an alternative point estimate of DegD, the sum of degrees of the entire sampled set is computed.

**4.2.2 Distance Distribution**

As another dimension of testing the ability of a network sampling algorithm, a histogram of the shortest distance from each visited node to the start node *s*, is plotted for each algorithm. The variance of the distance distribution (DistD) is computed to ensure that walkers in an algorithm are not being trapped inside any dense communities with relatively very few edges to escape into the rest of the network.

TBC…

5. Results

6. Conclusion

**References**

[1] (reserved to be used in introduction)

[2] Li, Y., Wu, Z., Lin, S., Xie, H., Lv, M., Xu, Y., & Lui, J. C. S. (2019). Walking with perception: Efficient random walk sampling via common neighbor awareness. *Proceedings - International Conference on Data Engineering*, *2019*-*April*, 962–973. https://doi.org/10.1109/ICDE.2019.00090

[3] Goodman, L. A. (1961). Snowball Sampling. *The Annals of Mathematical Statistics*, *32*(1), 148–170. https://doi.org/10.1214/aoms/1177705148

[4] Maiya, A. S., & Berger-Wolf, T. Y. (2010). Sampling community structure. *Proceedings of the 19th International Conference on World Wide Web, WWW ’10*, 701–710. https://doi.org/10.1145/1772690.1772762

[5] Ribeiro, B., & Towsley, D. (2010). Estimating and sampling graphs with multidimensional random walks. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*, 390–403. https://doi.org/10.1145/1879141.1879192