Sampling Large Graphs

[Your Name]

Date

1 Introduction

A social network is modelled as a large graph, in which nodes denote entities or instances (e.g., user) and edges denote relationships between nodes (e.g., friendship, or co-authorship). In real world, social network appears everywhere, typical examples include friendship networks in Facebook, co-author and bibliography networks in DBLP. Such graphs are very large in size and contain millions of nodes and edges [2]. For example, some recent statistics show that Facebook, the largest online social network, has 2.072 billion active users as of January 2018, and there are one million links shared between friends every 20 minutes [1].

The massive amount of data created everyday on social networks have become a great source of information for different purposes such as sociology study and marketing analysis. Therefore, it is crucial for data miners to devise effective, efficient, and systematic approaches to handle cases when the size of the network is massive. One strategy to handle such large-scaled networks is Sampling i.e. select a subset of vertices or edges from the original graph such that properties of original graph can be approximated by the sampled graph. graph [3].

2 Motivation and Challenges

The growth in scale of the real-world social networks has imposed great challenges in information extraction, processing, and analysis for humans or even computers. Some of them are listed below:

- **Reduce computational cost :** For example, best time complexities for various tasks are $O(n * (n + m))$ for computing the all-pairs shortest-path length, eigen vector computation takes $O(n^3)$. For graphs with billions of nodes, none of these tasks can be solved in a reasonable amount of time! The runtime effort for all these tasks usually scales at least polynomially in the size of the graph, i.e. its number of nodes n. For large graphs, the run times of $O(n^c)$, where often $c \geq 3$, are not affordable [4]. Proper sampling approaches help us estimate the properties on a smaller sample, thereby greatly reducing the computational cost.
- **Incomplete graph data :** In some cases, network structures may be hidden or inaccessible due to privacy concerns. For example, some networks can only be crawled by accessing one-hop neighbors of currently visiting node. It is not possible to query the full structure of the network or is very time-consuming. Thus, we must obtain the properties of the graph by sampling.
- **Easy visualization :** Displaying even a relatively small graph of several thousand vertices on a screen is challenging because of the limit in screen size. Sampling provides an abstract version of the original graph. Thus, visualizing sampling results is easier than visualizing the original.

There are a number of traits which are found in every network, and can be useful in describing the general topology of a network. These include degree distribution, connected component size distribution, average path length, clustering coefficient, etc. A good sampling method must preserve these traits of the original network. An effective sampling algorithm aims to reduce the complexity of graph while allowing analysis of the small sample to yield the characteristics similar to those of the original graph.

3 Common Sampling Strategies

Sampling algorithms can be categorized into three groups: methods based on *randomly selecting nodes*, *randomly selecting edges* and the *exploration techniques* [5]. These categories are briefly described below:

1. **Random node selection :** There are three sampling algorithms based on random node selection. One way is to uniformly at random select a set of nodes, which is refered as *Random Node (RN) sampling*. In *Random Page Rank Node (RPN) sampling*, the probability of a node being selected into the sample is proportional to its Page Rank weight. In, *Random Degree Node (RDN) sampling*, the probability of a node being selected is proportional to its degree.

2. **Random edge selection :** Similarly to selecting nodes at random, one can also select edges uniformly at random. This algorithm is refered as *Random Edge (RE) sampling*. A slight variation of *RN* sampling is *Random Node-Edge (RNE) sampling*, where a node is picked uniformly at random followed by uniformly at random pick an edge incident to the node. In *Hybrid (HYB)* approach, a step of *RNE* sampling and a step of *RE* sampling is performed with probability $'p'$ and $'1-p'$ respectively.

3. **Exploration :** The idea behind sampling by exploration technique is that a node is uniformly selected at random and then for further selection, the nodes in its vicinity are explored. In *Random Node-Neighbor sampling*, a node is uniformly selected at random together with all of its out-going neighbors. In *Random Walk sampling*, the next-hop node is chosen uniformly among the neighbors of the current node. Same as random walk but with a probability $'p'$ we jump to any node in the network in *Random Jump sampling*. In *Snowball Sampling*, a small population of known nodes is taken in the beginning and the sample is expanded by asking those initial nodes to identify others to form a sample. *Forest Fire* is a recursive process, where a seed node is picked randomly and begin burning outgoing links and the corresponding nodes. If a link gets burned, the node at the other end point gets a chance to burn its own links, and so on recursively.

4 My Proposal

As we are interested in data analysis on very large graphs, we have to work around the problems such as:

- Computing the average shortest-path length of a large scale-free network requires high computation time. In a network with n nodes, computing values for average path length requires calculating $O(n^2)$ node distance [6]. If the network has millions of nodes or edges, the time needed in computing will be very large.
  Hence, we need to explore which sampling algorithm to use to estimate the average shortest path lengths in such networks.
- In some scenario, the storage of massive graphs is distributed over several systems. Hence, the need is to have a sampling algorithm that is massively parallel i.e. there is no central coordination, each vertex is processed independently, and only the direct neighbors of a vertex and a small subset of random vertices in the graph need to be known locally.

References

[1] https://www.ommicoreagency.com/facebook-statistics.

[2] Meng Fang, Jie Yin, and Xingquan Zhu. Active exploration for large graphs. *Data Mining and Knowledge Discovery Volume 30 Issue 3*, pages 511–549, May 2016.

[3] P. Hu and W. Lau. A survey and taxonomy of graph sampling. *arXiv.org*, pages 1–34, 2013.

[4] Christian Hübler, Hans-Peter Kriegel, Karsten Borgwardt, and Zoubin Ghahramani. Metropolis algorithms for representative subgraph sampling. *Proceedings of Eighth IEEE International Conference on Data Mining*, page 283–292, ICDM'08.

[5] Jure Leskovec and Christos Faloutsos. Sampling from large graphs. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '06*, page 631–636, 2006.

[6] Xiaohan Zhao, Alessandra Sala, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. Orion: shortest path estimation for large social graphs. *Proceedings of the 3rd Wonference on Online social networks*, pages 9–9, WOSN'10.