A Visual and Statistical Benchmark for Graph Sampling Methods

Fangyan Zhang[1] Song Zhang[1] Pak Chung Wong[2] J. Edward Swan II[1] T.J. Jankun-Kelly[1]

[1]Mississippi State University [2]Pacific Northwest National Laboratory

Abstract

Effectively visualizing large graphs is challenging. Capturing the statistical properties of these large graphs is also difficult. Sampling algorithms, developed to more feasibly observe and analyze large graphs, are indispensable for this task. Many sampling approaches for graph simplification have been proposed. These methods can be grouped into three categories: node sampling, edge sampling, and traversal-based sampling. It is still an open question, however, which single sampling technique produces the best representative sample. The goal of this paper is to evaluate commonly used sampling methods through a combined visual and statistical comparison. Initial results indicate that the effectiveness of a sampling method is dependent on the type of graph, the size of the graph, and the desired statistical property. The benchmark can be used as a guideline in choosing the proper method for a particular graph sampling task. The resulting benchmark can be incorporated into graph visualization and analysis tools.

Keywords: graph sampling, graph properties, graph drawing, visualization.

1 Introduction

Graphs are widely used for information visualization, particularly for those datasets that can be easily represented as a network [5]. Graph analysis and visualization [9] has evolved into a very active area of investigation over the last several decades. However, as the size of a graph grows, effectively displaying all of the nodes and edges becomes an extremely difficult challenge. Performing calculations and analysis on large graphs is also not an easy task.

Several reasons lead to the need for sampling techniques. The first reason is visualization. Displaying even a relatively small graph of several thousand nodes on a screen is challenging because of the limit in the screen size. Even if we could display all of the nodes and edges, it is often difficult to discern the internal structure. Sampling provides us an abstract version of the original graph. Visualizing sampling results is easier than visualizing the original [11]. The second reason is that estimation or calculation on a large graph is costly. Proper sampling approach helps us estimate the graph properties on a smaller sample, thereby greatly reducing the computational cost. The third reason is the lack of the complete graph data. In some cases, obtaining all graph data is not permitted or is very time-consuming. We have to sample the whole graph in order to calculate the graph properties.

Due to the above reasons, sampling algorithms aim to reduce the drawing complexity and preserve properties of the original graph, allowing analysis of the small sample to yield the characteristics possessed by the original graph.

While numerous graph sampling techniques have been proposed [6], there is a lack of comparison between the available methods. Practical questions often arise on which sampling method one should use for a particular application. To answer those questions, proper graph properties and metrics for useful comparison must be identified. Analysis of each sampling technique and each metric must be explored to ascertain which sampling methods are most suitable for estimating the original graph properties.

In this paper, we design a benchmark for comparing a number of graph sampling methods. Our comparison considers two complementary aspects: how effectively the method preserves the visual properties and how well it preserves the statistical properties of the graph. We conduct our study on directed and undirected graphs separately and use a number of statistical properties for comparison. To properly compare graph sampling methods for visualization, we fix the graph layout in both the original and sampled graphs.

The main contributions of our work are as follows:

- We implement fourteen graph sampling techniques in the benchmark.
- We build a benchmark for evaluating graph sampling methods for both visual and statistical properties.
- We study a number of graph data sets with the benchmark and analyze the results.

2 Related Work

Existing graph sampling algorithms can be classified into three types: node sampling, edge sampling, and traversal-based sampling [8][6]. Node sampling constructs subgraphs based on sampling nodes. Nodes can be sampled uniformly or independently. In some cases, node sampling methods integrate traversal-based sampling in order to use graph topology information, such as random walk sampling. The Metropolis algorithm [7] is a modified version of node sampling. It replaces some sampled nodes with other nodes, which may lead to higher consistency in graph properties with the original. Edge sampling, likewise, builds a subgraph using sampled edges. Random edge sampling, for example, samples edges randomly. Traversal-based sampling creates subgraphs based on the topological information from the original graph. This method does not sample nodes or edges directly but instead selects nodes using traversal-based algorithms. Breadth-first [1][17], random walk [17][13], and snowball sampling [1] are commonly used traversal-based sampling algorithms that select nodes based on the topological information of the graph.

One purpose of sampling is to simplify the graph for better visualization. With millions or billions of nodes or edges, it is challenging to visualize all of them effectively. Several techniques have been proposed to enhance graph visualization, such as clustering [14], sampling [8][11], and special layout. These techniques aim to reduce the overlap between nodes and edges in visualization. Graph clustering techniques have two categories: nodes clustering [16][10] and edge clustering. Sampling approaches produce a small graph and make it less cluttered in visualization. Layout techniques mainly explore how to arrange nodes and edges when visualizing graphs. Many layout techniques have been proposed, such as Tree layout [12][15] and force-directed layout [3].

3 VISUAL AND STATISTICAL BENCHMARK

3.1 Graph Sampling Methods

Within the benchmark, we implement random node (RN), random node-edge (RNE), random node-neighbor (RNN), streaming nodes (SN), random edge (RE), induced edge (IE), streaming edge (SE),breadth-first (BF), depth-first (DF), random first (RF), snowball (SB), random walk (RW), random walk with escape (RWE), and forest fire (FF) sampling. We apply these sampling methods to four types of datasets: directed random graph datasets, undirected random graph datasets, VAST 2013 network flow dataset (directed), and American Airlines connections dataset (undirected). The number of nodes in random datasets ranges from 500, 750, 1000, to 1250. We also compare two random graphs with 500 nodes and two random graphs with 1000 nodes to study the consistency of the random graph results. Directed graphs and undirected graphs have different properties; therefore, we evaluate them separately.

3.2 Statistical Comparison

We use seven statistical properties for comparing undirected graphs and eight statistical properties for comparing directed graphs. We employ Kolmogorov-Smirnov (KS) D-statistic to evaluate the similarity between the original graph and a sampled graph based on those properties. For undirected graphs, we use the degree distribution (DD), betweenness centrality distribution (BB), clustering coefficient (CCD), average neighbor degree distribution (ANDD), degree centrality distribution, edge betweenness centrality distribution (EBCD), and hop distribution (HD). For directed graphs, we use in-degree distribution (InDD), out-degree distribution (OutDD), betweenness centrality distribution (BCD), average neighbor degree distribution (ANDD), in-degree centrality distribution (InDCD), out-degree centrality distribution (OutDCD), hops distribution (HD), and hops distribution in largest weakly connected component (HLCCD). To compare computing time for each sampling method, we keep a record of the execution time for each method.

3.3 Visual Comparison

We use Gephi [2] for visual comparison between sampling methods. We first draw the original graph and use this layout for all sampled graphs—i.e., the same node in all sampled graphs will occupy the same location as in the original graph. Also, the same node in all sampled graphs has the same color and label size as the original graph. We do not preserve the attributes of edges, such as edge color, edge weight, etc.

For the American Airlines connection data, we use the geospatial layout. For the random graph, we use the force-directed layout.

4 Results and Analysis

Because of limited space, we show only part of the results in this section.

4.1 Undirected Graph

For undirected graphs, we apply the sampling approaches to the American Airlines connection data and six random undirected graphs. The American Airlines connection data has 235 nodes and 1297 edges. The random graphs are created using NetworkX [4], which is a Python package dealing with graphs or networks. In this paper, we conduct an experiment with sampling rate from 10 to 50 percent with a 10 percent interval on both the American Airlines graph and the random undirected graphs. All of these sampling rates are based on the number of edges. The bar charts in the following figures show the KS distance between the sampling result and the original graph for each sampling method and each statistical property. The horizontal axis in the bar chart is the KS value between the sampling result and the original graph ranging from 0 to 1. A smaller value indicates more consistency between the sampling results and the original graph. The vertical axis lists the sampling methods grouped by the graph properties. From the bar charts, it is not difficult to determine which sampling method perform best for each graph property. Figure 1 and Figure 2 show the KS distance of two individual sampling rates (10% and 50%) for the sampling methods and statistical properties. Figure 3 shows the average KS distance of the five sampling rates of the American Airlines connections graph. Figure 4 shows the average KS distance of the five sampling rates of one random undirected graph with 1000 nodes and 4989 edges. We also show the execution time for each sampling method (Figure 5) for the random undirected graph. When comparing sampling methods between two graphs, we use ranks to analyze the performance of the sampling methods. If one sampling method appears in top 3 in both graphs, we consider it as consistently good. We present the rank consistency between the results of the two random undirected graphs (Table 1).

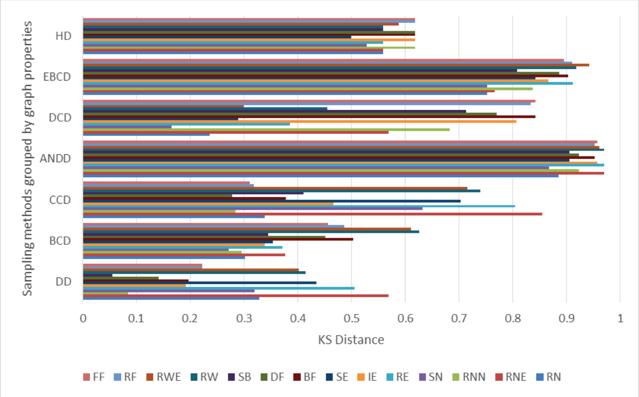

Figure 1: Statistical comparison among sampling methods for American Airlines connection data with 10 percent sampling rate (undirected graph).

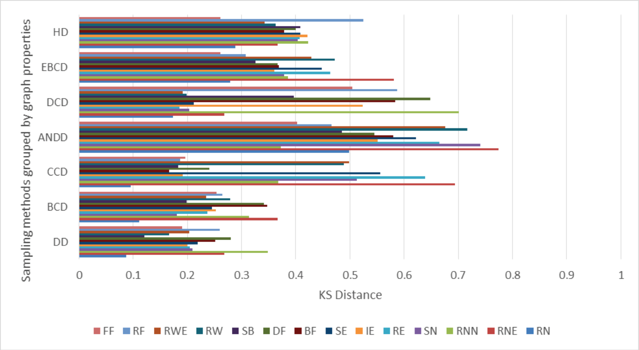

Figure 2: Statistical comparison between sampling methods for American Airlines connection data with 50 percent sampling rate (undirected graph).

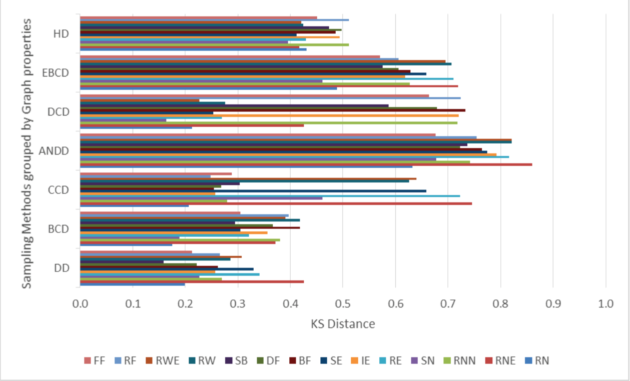

Figure 3: Average result of the statistical comparisons between sampling methods for American Airlines connection data (undirected graph) with 10 to 50 percent sampling rates.

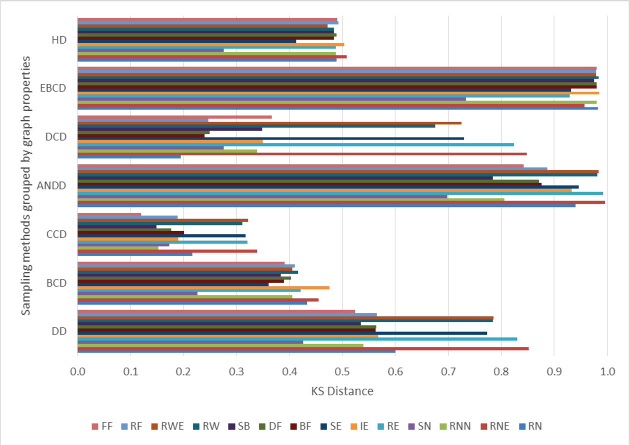

Figure 4: Average result of statistical comparisons between sampling methods for undirected random graph data with 10 to 50 percent sampling rates.

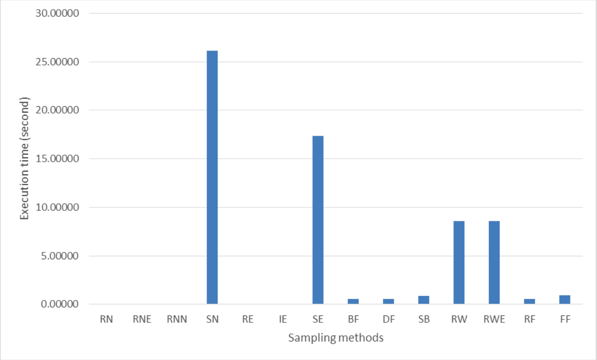

Figure 5: Average execution time between sampled methods for undirected random graph data. X axis represents KS distance. Y axis represents execution time in seconds for sampling methods.

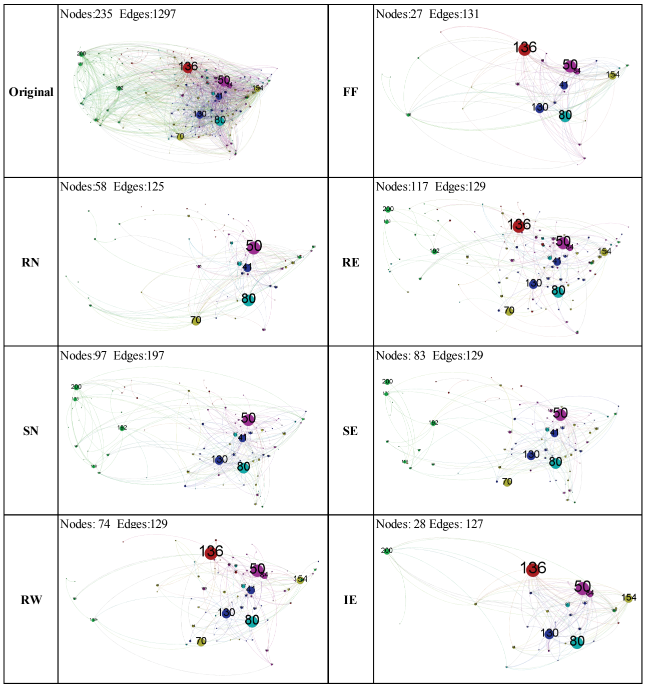

Figure 6: Visual comparison between sampling methods for American Airlines connection data (undirected graph) with sampling rate 10% on edges

Table 1: Sampling method consistency between two random undirected graphs with 1000 nodes.

DD	BCD	CCD	ANDD	DCD	EBCD	HD
SN	SN	FF	SN	RN	SN	SN
FF	SE	SB	SB	BF	RE	SE
SB	SB	RNN	RNN		SE	RWE

In addition to the statistical comparison above, we provide visual comparison of the sampling results (Figure 6). The visual comparison provides an intuitive method for identifying similarities and differences between the sampling results. We choose the same number of nodes (10%) for each sampling method in this result. Due to the page limit, we only provide several sampling results in the visual comparison.

4.2 Directed Graph

We apply the sampling approaches to the VAST data and six random directed graphs. VAST 2013 week 3 net flow data is a directed dynamic graph. We ignore the time sequence and treat it as a static graph in our experiment. The VAST graph has 1214 nodes and 15,653 edges. We also conduct an experiment with sampling rates from 10 percent to 50 percent with 10 percent intervals. All of these sampling rates are based on the number of edges. We only provide the average (Figure 7) of the five sampling rates on the VAST data. The following bar chart has the same layout with the previous bar charts.

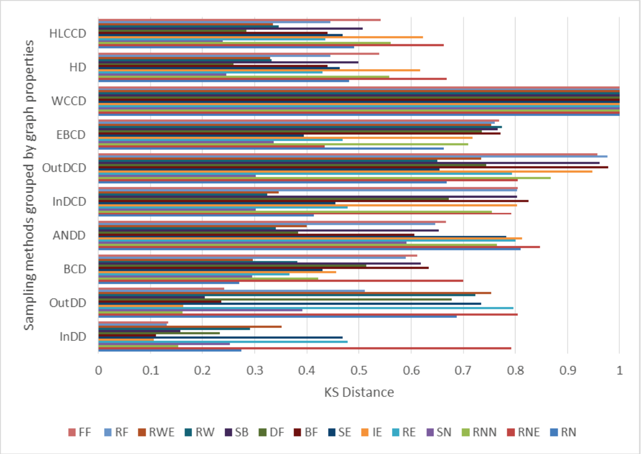

Figure 7: Average result of the statistical comparisons between sampling methods with 10 to 50 percent sampling rates for VAST data (directed graph).

4.3 Analysis

The benchmark allows us to compare different sampling methods quantitatively and qualitatively. We observe several findings after analyzing the results.

We analyze the results from five different sampling rates on American Airlines flight data, random graph data, and VAST data respectively. For each graph property, we observe the KS distance between the sampling results and the original graph. We also determine whether the sampling methods have stable performance on different sampling rates and for different graph types. In this paper, we define a sampling method as a "good" method for a property if the KS value between the sample and the original is ranked in the top three of all methods. When comparing sampling methods between two graphs, we regard a method as stable if it ranks in the top 3 for both graphs.

From Figures 1 and 2, we observe that, not surprisingly, by increasing the sampling rate of nodes from 10 percent to 50 percent, the sampling results show closer statistical properties to the original graph.

For random graph with the same number of nodes, we find that there are many consistently good sampling methods in both directed graph and undirected graphs. From two graphs of 500 nodes and two graphs of 1000 nodes (Table 1), we observe that the good sampling methods have better consistency with the increasing number of nodes.

However, the sampling methods' performances vary across different types of graphs. There are relatively few consistently good methods between the undirected random graphs and the American Airlines flight graph or between the directed random graphs and the VAST graph. We also find that the streaming node sampling has relatively good performance for many statistical properties in the VAST data. We believe that this result is because there are several high-degree server nodes in the VAST data. Such nodes have more probabilities entering the streaming pipe than exiting the streaming pipe. Thus, high-degree nodes have high probabilities being sampled, and these nodes preserve several graph properties better than other nodes.

When observing the random graphs from 500 to 1250 nodes, we find that sampling methods have inconsistent performance on the graphs with different numbers of nodes. The more differences there are in the number of nodes, the less consistency there is between them.

We apply some sampling methods to both directed and undirected graphs. We find that the same sampling method on both graphs have different performances.

Using visual comparison, we can clearly see the difference and similarity between each graph sampling method. Edge-related sampling methods, e.g., random edge sampling, induced edge sampling, streaming edge sampling, are biased towards high-degree nodes. These methods are more suitable if high-degree nodes are important in an application. For example, in Figure 4, the edge-related sampling results for the American Airlines connection data contain almost all nodes with high degree. On the other hand, random node sampling does not preserve high-degree nodes.

A combined analysis of both results will allow users to make more informed choices on sampling methods and obtain a good estimation of the original graph properties.

5 Conclusion and Discussion

Our visual and statistical benchmark evaluates a number of graph sampling methods based on their effectiveness in preserving both the quantitative statistical properties and qualitative visual properties of the original graph.

Our initial analysis indicates that the ranking of these graph sampling methods is dependent on a list of factors including the graph type, the graph size, and the desired statistical property. This finding points to the necessity of a performance benchmark for the graph sampling methods.

Furthermore, the visual comparison of the sampling methods gives users intuitive understanding of the differences among them. The consistent graph layout in the benchmark facilitates the visual comparison and identification of features for each sampling method.

Finally, the results provide insight into the effectiveness of each sampling method in preserving statistical properties and visualization, helping users with their choices of these methods in applications.

Acknowledgments

This work is supported by the Pacific Northwest National Laboratory under the U.S. Department of Energy Contract DE-AC05-76RL01830.

References

[1] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of Topological Characteristics of Huge Online Social Networking Services," in *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 835–844.

[2] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An Open Source Software for Exploring and Manipulating Networks," *Third Int. AAAI Conf. Weblogs Soc. Media*, pp. 361–362, 2009.

[3] T. M. J. Fruchterman and E. M. Reingold, "Graph Drawing by Force-directed Placement," *Software-Practice Exp.*, vol. 21, no. 11, pp. 1129–1164, 1991.

[4] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, 2008, vol. 836, pp. 11–15.

[5] I. Herman, G. Melançon, and M. S. Marshall, "Graph visualization and navigation in information visualization: a survey," *IEEE Trans. Vis. Comput. Graph,* vol. 6, no. 1, pp. 24–43, 2000.

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

[7] C. Hübler, H. P. Kriegel, K. Borgwardt, and Z. Ghahramani, "Metropolis algorithms for representative subgraph sampling," *Proc. IEEE Int. Conf. Data Mining, ICDM*, no. 1, pp. 283–292, 2008.

[8] J. Leskovec and C. Faloutsos, "Sampling from large graphs," *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 631–636, 2006.

[9] E. A. Lopez-Rojas, "Social Network Analysis in the dataset US Air 97 with Pajek," *LiU*, no. 1, pp. 1–11, 2011.

[10] C. Muelder and M. Kwan-Liu, "Rapid graph layout using space filling curves," *IEEE Trans. Vis. Comput. Graph,* vol. 14, no. 6, pp. 1301–1308, 2008.

[11] D. Rafiei and S. Curial, "Effectively visualizing large networks through sampling," in *Proceedings of the IEEE Visualization Conference*, 2005, p. 48.

[12] E. M. Reingold and J. S. Tilford, "Tidier Drawings of Trees," *IEEE Trans. Softw. Eng.*, vol. SE-7, no. 2, pp. 223–228, 1981.

[13] B. Ribeiro and D. Towsley, "Estimating and Sampling Graphs with Multidimensional Random Walks," pp. 390–403, 2010.

[14] A. Sallaberry, C. Muelder, and K.-L. Ma, "Clustering, Visualizing, and Navigating for Large Dynamic Graphs," in *Proceedings of the 20th International Conference on Graph Drawing*, 2013, pp. 487–498.

[15] J. Q. Walker II, "A node-positioning algorithm for general trees," *Softw. -- Pract. Exp.*, vol. 20, pp. 685–705, 1990.

[16] A. Y. Wu, M. Garland, and J. Han, "Mining scale-free networks using geodesic clustering," *Proc. 2004 ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 719 – 724, 2004.

[17] S. Yoon, S. Lee, S. H. Yook, and Y. Kim, "Statistical properties of sampled networks by random walks," *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, vol. 75, no. 4, 2007.