A new algorithm for extracting a small representative subgraph from a very large graph

Harish Sethu and Xiaoyu Chu
Department of Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104-2875
Email: {sethu, xiaoyu.chu}@drexel.edu

Abstract

Many real-world networks are prohibitively large for data retrieval, storage and analysis of all of its nodes and links. Understanding the structure and dynamics of these networks entails creating a smaller representative sample of the full graph while preserving its relevant topological properties. In this report, we show that graph sampling algorithms currently proposed in the literature are not able to preserve network properties even with sample sizes containing as many as 20% of the nodes from the original graph. We present a new sampling algorithm, called Tiny Sample Extractor, with a new goal of a sample size smaller than 5% of the original graph while preserving two key properties of a network, the degree distribution and its clustering co-efficient. Our approach is based on a new empirical method of estimating measurement biases in crawling algorithms and compensating for them accordingly. We present a detailed comparison of best known graph sampling algorithms, focusing in particular on how the properties of the sample subgraphs converge to those of the original graph as they grow. These results show that our sampling algorithm extracts a smaller subgraph than other algorithms while also achieving a closer convergence to the degree distribution, measured by the degree exponent, of the original graph. The subgraph generated by the Tiny Sample Extractor, however, is not necessarily representative of the full graph with regard to other properties such as assortativity. This indicates that the problem of extracting a truly representative small subgraph from a large graph remains unsolved.

I. INTRODUCTION

Large networks, usually modeled as graphs, appear in a variety of contexts in computer science as well as in sociology, epidemiology, business and engineering [25]. Within computer science, tools that give us insight into the structure and dynamics of these networks are central to understanding the growth and evolution of the Internet [28], the nature of online social interactions [17], [23], data sharing patterns on peer-to-peer networks [29], and online epidemic behaviors (whether of ideas [7], [11], [14] or computer viruses and worms [26]). Some of these networks, however, are so large that technical limitations of storage, computing power, and bandwidth available to most researchers make it infeasible to crawl through the entire network (e.g., YouTube with over hundred million nodes or the network of web pages with billions of nodes). Collection of temporal data for understanding the evolution of these networks further increases the challenge because of the need for multiple snapshots of the networks. Even if the data is acquired, they can be prohibitively large for purposes of analysis, simulation or visualization on most computing systems. These challenges call for a fast algorithm that visits only a small fraction of a large graph to extract a sample subgraph which retains the most important topological properties of the original graph. The size of the sample subgraph needs to be significantly smaller than the original graph and free of measurement bias [18].

As we show later in this report, currently known sampling algorithms do not quite approach the properties of the original graph even with sample sizes as large as 20%. In this work, we pursue the above challenge with a target of shrinking a network to less than *five percent* of its original size while preserving a key property of the graph, its degree distribution. As has been argued in [12], finding the optimal subgraph S of a certain size that best matches a property of the original graph G is an NP-complete problem for most graph properties. Given that the size of G is often of the order of tens of millions, finding the optimal subgraph is obviously not feasible. A further constraint that adds to the challenge is the fact that all of the graph G is not usually visible to the crawler or is not even accessible. This is frequently the case while crawling an online social network, the network of web pages or peer-to-peer networks. Often, however, if access is secured to a node, it is possible to secure access to the neighbors of the node. The goal is to extract a representative subgraph based on crawling through the original graph beginning with a node in a known portion of the graph. A further goal is that the sampling algorithm be scalable, i.e., its computational costs should increase linearly with the desired size of the sample subgraph and not depend upon the size of the original graph.

In the following, we now formalize our problem statement. Consider a large graph G of n nodes. Our goal is to extract a subgraph S of G with the following properties:

1) S has h nodes where $h \leq 0.05n$.
2) S has a degree exponent as close as possible to the original graph G.

under the following constraints:

1) The number of nodes of G visited by the sampling algorithm is $O(h)$.
2) Properties of the original graph G are not inputs to the sampling algorithm.

Section II presents known solutions related to the problem statement described above and builds the rationale for the algorithm proposed in this report. Section III presents our *Tiny Sample Extractor*, a sampling algorithm that uses a biased random walk to discover new nodes and returns to the starting point upon discovery of each new node. The bias is used to compensate for the skewed distribution of nodes visited in random walks or a breadth-first search. Section

IV presents a comparison of our approach to other sampling strategies with respect to a number of graph properties including degree distribution, assortativity and clustering properties. The results show that our algorithm is able to extract a much smaller representative sample than other sampling algorithms while also achieving a closer convergence to the degree distribution of the original graph. Our results also show that a sample generated for preserving a particular property can fail to adequately preserve another property. Section V concludes the report.

II. Related Work

A variety of strategies have been used to “shrink” a graph for purposes of analysis. Shrinking a graph by extracting an actual subgraph allows one to discover patterns in the subgraph that can be validated later in the original graph because of the one-to-one correspondence between the nodes in the two graphs. The generation of synthetic topologies which have a specific set of properties in them [2], [10], though useful in many contexts, is not considered in this report.

A sample subgraph induced by a randomly selected set of nodes has been discussed in several works on graph sampling [4], [5], [12], [27]. Selecting nodes randomly ensures that nodes of a given degree are chosen with probability proportional to the number of such nodes in the network. The selected set of nodes have a degree distribution very similar to that of the original graph, but these degrees are the degrees of the nodes in the original graph G and not the degrees in the induced subgraph S. There are at least two additional problems with such a sampling strategy: (i) when the desired sample size is as small as 5% of the original graph, the induced subgraph is highly likely to be a disconnected graph even if G is connected and thus, unrepresentative; and (ii) in real networks that have to be crawled, it is usually very hard or infeasible to generate a statistically valid set of uncorrelated random nodes from the full graph G given that the full graph is not known (even though a few random nodes can always be selected from within the known portion of the graph).

A related set of sampling strategies is based on selecting random edges instead of random nodes or a combination of node and edge sampling [5]. In general, however, edge sampling does not overcome the problems of node sampling mentioned above. Sampling strategies based on random deletion [15], [16] instead of selection also suffer the same problems and are not suitable as solutions to the problem statement expressed in Section I. Node or edge sampling is useful in contexts where the goal is to infer properties of nodes but not necessarily the topological properties of the graph. The choice of nodes guided by simulated annealing can target a specific set of topological properties [12], but this method also relies on randomly choosing nodes from the entire network.

As an improvement, one may resort to a random walk on the graph to ensure that a connected set of nodes is chosen for the sample. As discussed in [29], however, a random walk visits a node with probability proportional to its degree, leading to a biased sample[1]. This is corrected in the *Metropolized Random Walk (MRW)* [29], based on the Metropolis-Hastings method for Markov chains [9]. In MRW, a move from node x to node y is made with probability $P(x, y)$ given by:

$$P(x, y) = \frac{1}{\text{degree}(x)}$$

[1] The random walk technique is identical to one where we select nodes at random with a probability proportional to its PageRank [6].

and then, the move is accepted with probability:

$$\min\left(1, \frac{\text{degree}(x)}{\text{degree}(y)}\right).$$

If the move is not accepted, we return to node x and attempt a move again. The expected distribution of the degrees (in graph G) of the visited nodes in MRW is identical to the actual distribution of the degrees in G. However, the subgraph induced by the set of visited nodes is highly unlikely to have a similar distribution as G. This is because a random walk is more likely to yield a "string" of connected nodes rather than a scaled-down network. The MRW algorithm, however, is used in our approach, not to directly extract a sample subgraph but to first obtain an estimate of the degree distribution in the original graph G.

A further improvement in graph sampling is achieved with *Snowball sampling,* which chooses a random node from the known portion of the graph G and then proceeds with a breadth-first search until the desired size of the sample graph is achieved [24]. Snowball sampling and its derivatives have been used in social network analysis [19], [23]. As reported in [1], for small sample sizes, it is inconclusive if the clustering co-efficient of the sample network converges to that of the complete network (as will be verified in our work as well). In addition, Snowball sampling has been shown to over-sample "hubs" or large-degree nodes in a network because of its breadth-first strategy which hits a hub with a greater likelihood.

A related strategy is one called *Forest Fire,* first introduced in [21]. In this method, as in Snowball sampling, we choose a random node from the known part of G and use a breadth-first approach. With a "forward burning probability" p_f, the node burns links attached to it. The nodes at the other end of a burned link are added to the sample subgraph and they now continue spreading the "fire" by burning links attached to them. This continues until the desired size of the sample subgraph is achieved. It has been found in [20] that the Forest Fire sampling strategy works best with $p_f = 0.7$ and this is what we use in all our simulations in this report. In general, it has been found that methods based on BFS search are likely to overestimate node-degrees and underestimate symmetry [19]. As we will show later, the Forest Fire sampling strategy, being based on a "scaled-down" BFS, is not entirely able to reduce the likelihood of adding high-degree nodes to the sample subgraph.

A more rigorous but different approach to random subgraph sampling has only recently been attempted in [22] which evaluates a number of different strategies including Random Vertex Expansion [13]. However, while the subgraphs sampled in the methods proposed in [22] achieve a sampling of subgraphs uniformly at random, they do not actually extract a single subgraph that is most representative of the full graph with respect to any given property. As a result, random sampling of subgraphs do not readily help us discern properties of the full graph, especially since the sampling of subgraphs uniformly at random leads to an over-representation of properties from dense portions of the graph.

III. The Tiny Sample Extractor

Our algorithm relies on first finding an estimate of the degree distribution of nodes in the original graph G. We use the Metropolized Random Walk for this purpose and use the degree exponent, $\mathcal{D}$, as defined in [28], to capture the degree distribution. The complementary cumulative distribution function (CCDF) of a degree d is defined as the fraction of nodes that have degree

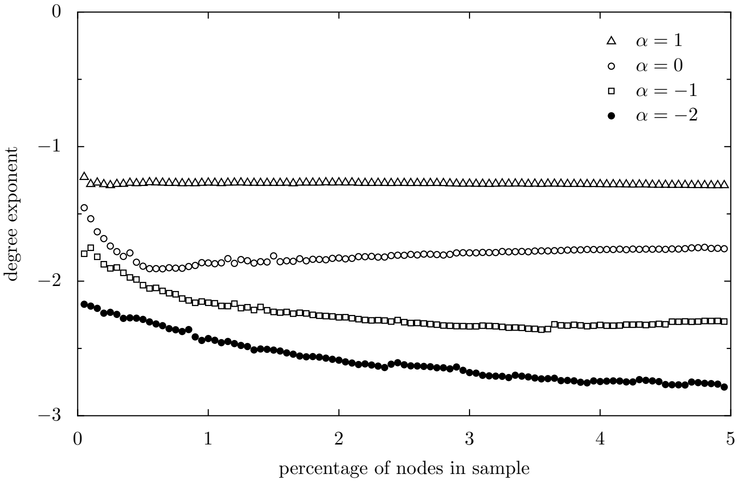


Fig. 1. The degree exponent of sample subgraphs extracted by the Biased Random Walk with Fly-Back (BRW-FB) for different values of α. The plot shows that there is a linear relationship between α and the degree exponent to which the sample subgraph converges.

greater than the degree d. As in [28], the degree exponent, $\mathcal{D}$, is defined as the slope of CCDF(d) against d on a log-log plot. An alternate definition of the degree exponent is one based on the frequency $f(d)$ with which a node of degree d appears in the network. Either definition would serve the purposes of this work but the degree exponent based on CCDF permits statistically superior curve-fitting to determine the slope of the log-log plot.

Given an estimate of the degree exponent of the full graph G, our sampling algorithm is based on compensating for the biases introduced by BFS-based methods such as Snowball and Forest Fire. Our algorithm begins with a random node in the known part of the graph G and starts a biased random walk until it finds a new unvisited node and then, flies back to the starting node to begin another biased random walk. The bias in the random walk is parametrized by α (we will shortly discuss how we determine α). Given α, if the algorithm is at node x, then it visits a neighbor y of x with probability $B(x, y)$ given by:

$$B(x,y) = \frac{[\text{degree}(y)]^{\alpha}}{\sum\limits_{n \in \Gamma(x)} [\text{degree}(n)]^{\alpha}} \tag{1}$$

In other words, the walk proceeds to a neighbor y with a probability proportional to the degree of y in G raised to the power of α. Visited nodes are added to the sample and this continues until the desired sample size is reached. The induced subgraph of these nodes becomes the sample subgraph S. Algorithm 1 presents the pseudo-code of BRW-FB.

Figure 1 presents an empirical demonstration of the linear relationship between α and the degree exponent of the sample generated by BRW-FB. In this figure, we use values of α in the range between -2 and 1 based on our ongoing work on crawling the network of YouTube users and the network of web pages. While this relationship is different on different networks, the linearity of it persists across all networks on which we have attempted our algorithm. If we know the degree exponent yielded by two chosen values of α, the linear equations are readily

Algorithm 1 Biased Random Walk with Fly Back (BRW-FB)

Input: G (input graph), h (desired sample size), α
Output: S (sample graph)

```
$N = \phi$ (set of nodes in sample graph)
Choose random node $p$ in $G$
Add $p$ to $N$
while $|N| < h$ do
  foundANewNode $\leftarrow$ False
  $x \leftarrow p$
  while not foundANewNode do
    Choose a neighbor $y$ of $x$ with probability $B(x, y)$ (see Equation (1))
    if $y \in N$ then
      $x \leftarrow y$
    else
      foundANewNode $\leftarrow$ True
      Add $y$ to $N$
    end if
  end while
end while
$S \leftarrow$ subgraph of $G$ induced by node set $N$
return $S$
```

solved to generate an estimate of the relationship between α and the degree exponent for the graph under consideration. Given this linear relationship, it is possible to target a specific degree exponent in the sample subgraph with the choice of an appropriate α.

We now present our *Tiny Sample Extractor*, which first executes the BRW-FB with $\alpha = 0$ and $\alpha = 1$ to estimate the sensitivity of the degree exponent to α and determine the underlying relationship. Extrapolating based on this linear relationship, the Tiny Sample Extractor computes the α corresponding to the degree exponent estimated by the Metropolized Random Walk. Algorithm 2 presents the pseudo-code.

IV. Performance Analysis

We use the Barabási-Albert scale-free network described in [3] for purposes of comparison. This being an extremely well-behaved network, it illustrates more acutely the fact that the existing sampling strategies do not approach properties of the actual network even with one-fifth of the nodes in the sample. The specific instance of the Barabási-Albert network we choose is one that begins with two unconnected nodes. When each new node is added to the network, two edges are created between it and two pre-existing nodes. The probability with which an edge connects to an existing node of degree d is $d/\sum d_i$ where d_i is the degree of node i. The resulting network is a connected network.

The degree distributions of the samples extracted from this network by four sampling strategies are shown in Figure 2. Figure 2(a) plots the degree exponents of the subgraphs induced by the

Algorithm 2 Tiny Sample Extractor

Input: G, h (desired sample size)

Output: S (sample graph)

```
  $D \leftarrow$ MRW$(G, h)$
  $S_0 \leftarrow$ BRW-FB$(G, h, 0)$
  $D_0 \leftarrow$ degree exponent of $S_0$
  $S_1 \leftarrow$ BRW-FB$(G, h, -1)$
  $D_1 \leftarrow$ degree exponent of $S_1$
  $\alpha \leftarrow -\left(\frac{D - D_0}{D_1 - D_0}\right)$
  $S \leftarrow$ BRW-FB$(G, h, \alpha)$
  return $S$
```

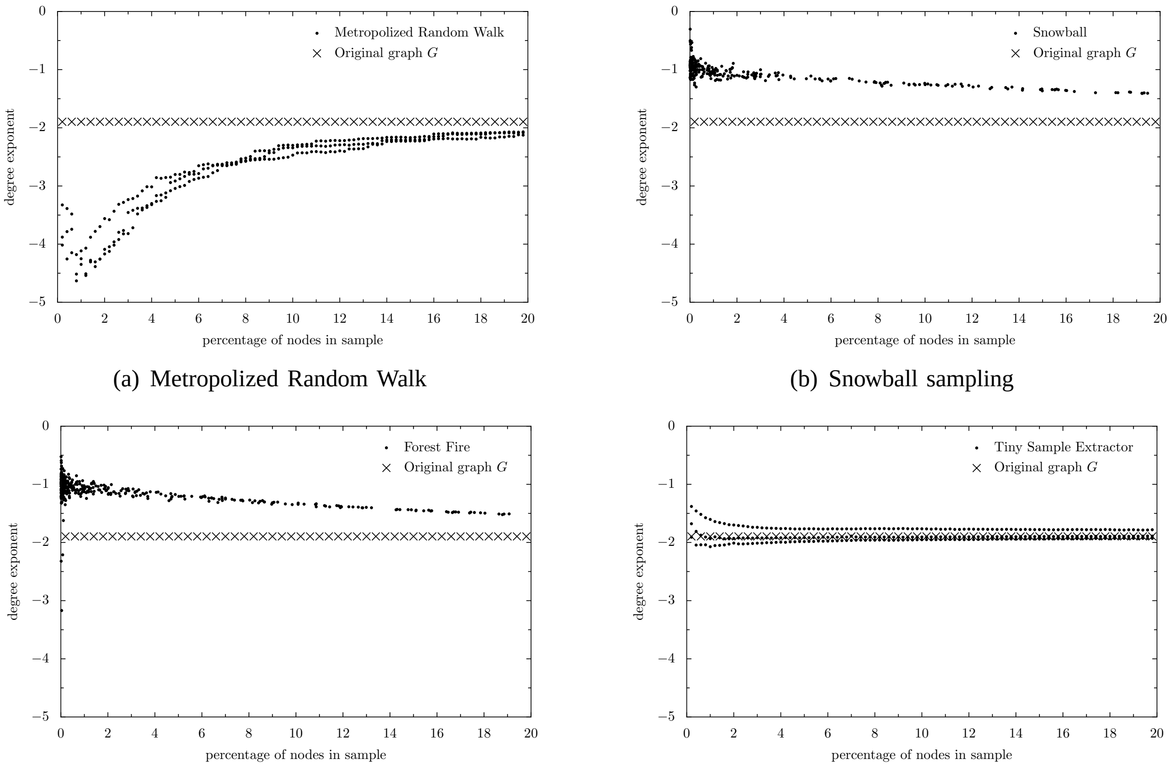
nodes visited by the Metropolized Random Walk (MRW) as the set of nodes in the sample grows. The figure plots three different samples, each starting from a different random node. The goal of MRW is *node sampling* and not *graph sampling*; therefore, it is not quite fair to the MRW algorithm to plot the degree distribution in the induced subgraph. However, we do so here largely to illustrate that node sampling does not directly yield a representative subgraph.

Figure 2(b) plots the degree exponents achieved by Snowball sampling. The subgraph generated by Snowball sampling grows very fast because of its BFS approach, and therefore, very soon includes a large percentage of the nodes. Step i of the sampling strategy includes all nodes reachable in i hops or less from the starting node. Since only a few (2–4) steps is needed to reach 20% or more of the nodes in the network, plotting degree exponents for only a small number of samples as they grow does not fully illustrate the rate of convergence of the subgraph to the degree distribution of the original graph. The figure, therefore, plots degree exponents from one hundred samples as each sample grows to 20% of the network. As mentioned in Section II, Snowball sampling over-samples high-degree nodes and, as a result, the induced subgraph does not quite approach the degree exponent of the original graph. In fact, even with 20% of the nodes in the sample, the degree exponent of the sample subgraph does not converge to that of the original graph. Figure 2(c) similarly plots the degree exponents corresponding to one hundred different subgraphs extracted by the Forest Fire sampling strategy. As is readily observed, its performance is very similar to Snowball sampling, though slightly better.

Finally, Figure 2(d) plots the degree exponents reached by the Tiny Sample Extractor. As in the case of the Metropolized Random Walk, we plot the degree exponents for three samples. The figure demonstrates that the Tiny Sample Extractor converges to the degree distribution of the original graph significantly faster than other sampling algorithms.

We now focus on two additional properties of graphs: the average clustering co-efficient and its assortativity [8]. Since the Forest Fire and Snowball sampling strategies are similar in performance with the Forest Fire faring slightly better, for purposes of clarity in the plots, we omit Snowball sampling in subsequent analysis.

Figure 3(a) plots the assortativity of the sample graphs generated by Forest Fire and the Tiny



(a) Metropolized Random Walk

(b) Snowball sampling

(c) Forest Fire

(d) Tiny Sample Extractor

Fig. 2. The degree exponent of sample subgraphs obtained using different sampling strategies. The original graph used is a million-node Barabási-Albert scale-free graph with two edges added per new node. For the Metropolized Random Walk and the Tiny Sample Extractor, we extract three sample subgraphs. For Snowball sampling and Forest Fire, we extract hundred sample graphs.

Sample Extractor. Assortativity measures the tendency of nodes to attach to other nodes that are similar or different in any particular way. The most commonly used definition of “similarity” used in studying assortativity is one based on degrees. We define assortativity as:

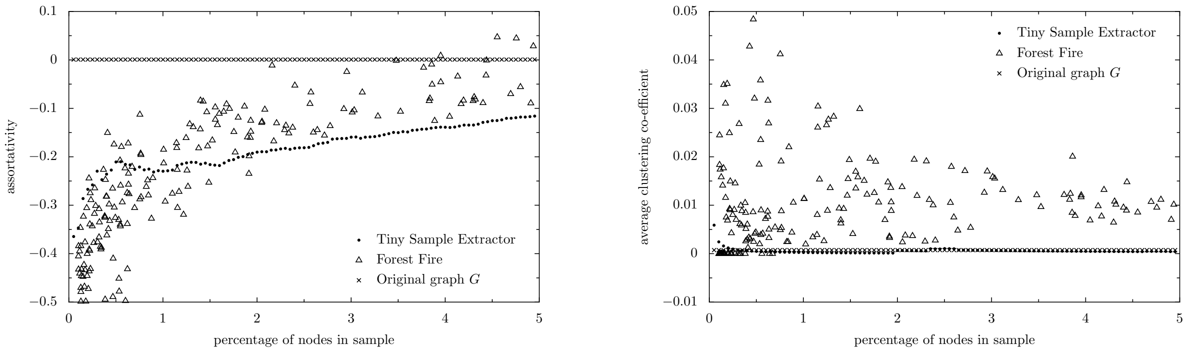
$$\frac{\langle d_i d_j \rangle - \langle d_i \rangle \langle d_j \rangle}{\sqrt{(\langle d_i^2 \rangle - \langle d_i \rangle^2)(\langle d_j^2 \rangle - \langle d_j \rangle^2)}}$$

where d_i and d_j are degrees of nodes at either end of an edge and the $\langle . \rangle$ notation represents an average over all edges in the network. The performance of Forest Fire is not as well-behaved as that of the Tiny Sample Extractor but slightly closer to the assortativity of the original graph.

Figure 3(b) plots the average clustering co-efficient of the sample graphs and the original graph. We define the clustering co-efficient of a node v as:

$$\frac{2T(v)}{d_v(d_v - 1)}$$

where $T(v)$ is the number of triangles that exist through node v. Note that the clustering co-efficient measures the fraction of triangles that exist out of all potential triangles through a



(a) Assortativity

(b) Average clustering co-efficient

Fig. 3. A comparison of assortativity and clustering properties of the sample subgraphs in relation to those of the original graph.

node. The average clustering co-efficient is the average over all nodes. We find that the Tiny Sample Extractor is highly accurate in generating a sample that matches the average clustering co-efficient of the original graph. In fact, as far as the clustering co-efficient is concerned, the algorithm converges to that of the original graph with as little as one percent of the nodes in the sample.

V. Concluding Remarks

In this report, we have presented a new crawling algorithm, called the Tiny Sample Extractor, which extracts a small sample subgraph from a large graph while retaining its essential properties, in particular its degree exponent and the clustering co-efficient. A key feature of our algorithm is that it achieves a convergence to these properties of the original graph faster with a smaller sample than other algorithms. This allows a crawler to take multiple snapshots of the crawled network in order to study the temporal evolution of large dynamic networks.

However, this work also illustrates that the problem of extracting a subgraph that is representative of the full graph with respect to all its properties remains unsolved. As shown in Section IV, the Tiny Sample Extractor does not do well with respect to the assortativity of the graphs, even though it does better than other algorithms on degree distribution and the clustering co-efficient.

References

[1] Ahn, Y.-Y., Han, S., Kwak, H., Moon, S., and Jeong, H. Analysis of topological characteristics of huge online social networking services. In *WWW '07: Proceedings of the 16th international conference on World Wide Web* (New York, NY, USA, 2007), ACM, pp. 835–844.

[2] Airoldi, E. M., and Carley, K. M. Sampling algorithms for pure network topologies: a study on the stability and the separability of metric embeddings. *SIGKDD Explor. Newsl. 7*, 2 (2005), 13–22.

[3] Barabási, A. L., and Albert, R. Emergence of Scaling in Random Networks. *Science 286* (October 1999), 509–512.

[4] Becchetti, L., Castillo, C., Donato, D., and Fazzone, A. A comparison of sampling techniques for web characterization. In *Workshop on Link Analysis (LinkKDD)* (August 2006).

[5] Bennett, C. More efficient classification of web content using graph sampling. In *CIDM 2007: IEEE Symposium on Computational Intelligence and Data Mining* (2007), IEEE, pp. 485–490.

[6] Brin, S., and Page, L. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems* (1998), Elsevier Science Publishers B. V., pp. 107–117.

[7] Carnes, T., Nagarajan, C., Wild, S. M., and van Zuylen, A. Maximizing influence in a competitive social network: a follower's perspective. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce* (New York, NY, USA, 2007), ACM, pp. 351–360.

[8] Chakrabarti, D., and Faloutsos, C. Graph mining: Laws, generators, and algorithms. *ACM Comput. Surv. 38*, 1 (2006), 2.

[9] Chib, S., and Greenberg, E. Understanding the Metropolis-Hastings algorithm. *The American Statistician 49*, 4 (1995), 327–335.

[10] Eubank, S., Kumar, V. S. A., Marathe, M. V., Srinivasan, A., and Wang, N. Structural and algorithmic aspects of massive social networks. In *SODA '04: Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms* (Philadelphia, PA, USA, 2004), Society for Industrial and Applied Mathematics, pp. 718–727.

[11] Goecks, J., and Mynatt, E. D. Leveraging social networks for information sharing. In *CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work* (New York, NY, USA, 2004), ACM, pp. 328–331.

[12] Hubler, C., Kriegel, H.-P., Borgwardt, K., and Ghahramani, Z. Metropolis algorithms for representative subgraph sampling. In *ICDM '08: Eighth IEEE International Conference on Data Mining* (2008), IEEE, pp. 283–292.

[13] Kashtan, N., Itzkovitz, S., Milo, R., and Alon, U. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics 20*, 11 (2004), 1746–1758.

[14] Kempe, D., Kleinberg, J., and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2003), ACM, pp. 137–146.

[15] Krishnamurthy, V., Faloutsos, M., Chrobak, M., Cui, J.-H., Lao, L., and Percus, A. G. Sampling large Internet topologies for simulation purposes. *Comput. Netw. 51*, 15 (2007), 4284–4302.

[16] Krishnamurthy, V., Faloutsos, M., Chrobak, M., Lao, L., Cui, J.-H., and Percus, A. G. Reducing large Internet topologies for faster simulations. In *IN IFIP NETWORKING* (2005).

[17] Kumar, R., Novak, J., and Tomkins, A. Structure and evolution of online social networks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2006), ACM, pp. 611–617.

[18] Lakhina, A., Byers, J. W., Crovella, M., and Xie, P. Sampling biases in IP topology measurements. In *IEEE INFOCOM* (2003), IEEE, pp. 332–341.

[19] Lee, S. H., Kim, P.-J., and Jeong, H. Statistical properties of sampled networks. *Physical Review E 73* (2006), 016102.

[20] Leskovec, J., and Faloutsos, C. Sampling from large graphs. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining* (New York, NY, USA, 2006), ACM, pp. 631–636.

[21] Leskovec, J., Kleinberg, J., and Faloutsos, C. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining* (New York, NY, USA, 2005), ACM, pp. 177–187.

[22] Lu, X., and Bressan, S. Sampling connected induced subgraphs uniformly at random. *Lecture Notes in Computer Science 7338* (2012), 195–212.

[23] Mislove, A., Marcon, M., Gummadi, K. P., Druschel, P., and Bhattacharjee, B. Measurement and analysis of online social networks. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (New York, NY, USA, 2007), ACM, pp. 29–42.

[24] Newman, M. E. J. Ego-centered networks and the ripple effect. *Social Networks 25* (2003), 83–95.

[25] Newman, M. E. J. The structure and function of complex networks. *SIAM Review 45*, 2 (2003), 167–256.

[26] O'Donnell, A. J., and Sethu, H. On achieving software diversity for improved network security using distributed coloring algorithms. In *Proceedings of the ACM Conference on Computer and Communications Security* (2004), ACM Press, pp. 121–131.

[27] Shi, X., Bonner, M., Adamic, L. A., and Gilbert, A. C. The very small world of the well-connected. In *HT '08: Proceedings of the nineteenth ACM conference on Hypertext and hypermedia* (New York, NY, USA, 2008), ACM, pp. 61–70.

[28] Siganos, G., Faloutsos, M., Faloutsos, P., and Faloutsos, C. Power laws and the AS-level Internet topology. *IEEE/ACM Trans. Netw. 11*, 4 (2003), 514–524.

[29] Stutzbach, D., Rejaie, R., Duffield, N., Sen, S., and Willinger, W. On unbiased sampling for unstructured peer-to-peer networks. In *IMC '06: Proceedings of the 6th ACM SIGCOMM conference on Internet measurement* (New York, NY, USA, 2006), ACM, pp. 27–40.