Measuring Bias in the Mixing Time of Social Graphs due to Graph Sampling

Abedelaziz Mohaisen* Pengkui Luo† Yanhua Li† Yongdae Kim‡ Zhi-Li Zhang†
*Verisign Labs, VA, USA †University of Minnesota, MN, USA ‡KAIST, Daejeon, Korea

***Abstract*—Sampling of large social graphs is used for addressing infeasibility of measurements in large social graphs, or for crawling graphs from online social network services where accessing an entire social graph at once is often impossible. Sampling algorithms aim at maintaining certain properties of the original graphs in the sampled (or crawled) ones. Several sampling algorithms, such as breadth-first search, standard random walk, and Metropolis-Hastings random walk, among others, are widely used in the literature for sampling graphs. Some of these sampling algorithms are known for their bias, mainly towards high degree nodes, while bias for other metrics is not well-studied. In this paper we consider the bias of sampling algorithms on the mixing time. We quantitatively show that some existing sampling algorithms, even those which are unbiased to the degree distribution, always produce biased estimation of the mixing time of social graphs. We argue that bias in sampling algorithms accepted in the literature is rather metric-dependent, and a given sampling algorithm, while may work nicely and unbiased to one property, may produce considerable amount of bias in other properties.**

***Index Terms*—Mixing Time, Social graphs, Biased estimation, Sampling**

I. Introduction

Measurement of online social networks is a quickly evolving area of research, where there has been recent interest in measuring and understanding simple and complex properties of social structures alike [1], [33], [20], [16], [15], [14], [13]. While measuring simple properties—such as degree distribution, for instance—is easy to perform on large social graphs that include tens of millions of nodes and billions of edges [11], performing other relatively complex, yet of interest, measurements for certain properties is quite challenging [27]. The challenge in performing such measurements is inherent in the complexity of best-known algorithms to compute these properties, requiring thousands of computation hours for such large graphs. An example of an interesting property is node betweenness, which is computed as the number of shortest paths between any pair of nodes which pass through a given node normalized by the total number of shortest paths in the graph [4]. Computing the betweenness for all nodes in a graph of n nodes requires computing all-pairs shortest paths, which can be accomplished using Floyd-Warshall algorithm in $O(n^3)$ time complexity [29]. This time complexity is pivotal to these algorithms' infeasibility as the size of social graphs grow moderately to millions of nodes.

Another interesting property of both theoretical and practical significance is the mixing time of social graphs. The mixing time, the length of the random walk starting from any node in the graph to achieve a certain statistical distance from the stationary distribution of the Markov chain on the graph, is bounded by a single parameter; the second largest eigenvalue of the transition matrix representing the social graph [31]. Computing the eigenvalues of a matrix can be accomplished using Arnoldi/Lanczos iterative algorithms [12], which run in $O(n^2 \log n)$, pronouncing the process infeasible for large graphs. Yet worse, more expensive computations that include vector-matrix multiplications are required to describe the rich pattern of mixing characteristics in social graphs [27], [23], [26], [25]. For example, to compute how close is the distribution of random a $\log n$ walk from the stationary distribution for all sources (nodes) in a graph, a straightforward implementation has the time complexity of $O(n^3 \log n)$.

From both examples, it is clear that running these algorithms on large social graphs, which typically consist of millions of nodes, to compute their properties is infeasible. To address this infeasibility issue, sampling algorithms are often used for extracting graphs that contain smaller sets of nodes and edges from the original graph [27], and properties are feasibly computed in the sampled graphs. Finally, conclusions are made on properties in the original graph by observing these properties in the sampled graphs.

Sampling algorithms are also used for crawling social graphs from online social network services, where most publicly available graphs are often samples of entire graphs of such services [8]. In both cases; sampling to address feasibility of measurements and sampling due to crawling, measurements are only real reflections to properties in the sampled graphs. Thus, such measurements cannot be directly used to make firm conclusions on properties in the original graphs without accounting for bias in sampling algorithms. Some of the popular and widely used sampling algorithms include the breadth-first search (BFS) [10], standard random walk based (RW) [7], and the Metropolis-Hastings random walk based (RWMH) sampling [7], among many others. In some of these sampling algorithms, theoretical results are driven on the amount of bias in the computed properties to that existing in the original graph. This bias is marked as an "error" in the measurement due to sampling, and can be used to correct the quantified property in the original graph, when possible.

Unfortunately, our understanding of bias in sampling algorithms is very limited to a few properties and sampling algorithms. The most widely explored property, and bias in this property due to sampling, is degree distribution. It has been shown that sampling algorithms like BFS and RW are biased towards high degree nodes, unlike RWMH which is unbiased sampler of degree distribution [7]. Even a formal quantification of bias in the degree distribution in social graphs due to the BFS sampling—one of the simplest sampling algorithms—was not possible until very recently [10].

Our limited understanding of bias in properties computed for sampled social graphs to derive insights and firm conclusions on properties in unsampled graphs motivates this work. The property we consider in this paper is the *mixing time*, the walk length on the social graph required to achieve a constant distance from the a fixed distribution. We choose this property for its practical significance, where the aforementioned problem of undermining effects of sampling on the property rises in many contexts. We consider several sampling algorithms and quantify the amount of bias in these algorithms when variating several parameters.

1) Contributions: The contribution of this paper is a comparative study on the bias of sampling algorithms in estimating the mixing time of social graphs. We consider several real-world social graphs with different structures, and several sampling algorithms and compare them according to their bias introduced on the measured mixing time. We conclude with several interesting remarks pointing out that, unlike degree distribution that is easy to understand, and even possible to formally model for the amount of bias introduced due to sample, the bias on the mixing time due to sampling has different patterns that are not captured in a single tendency.

2) Paper organization: The structure of the rest of this paper is as follows. In section II we introduce the related work on measurements and sampling. In section III we provide preliminaries by reviewing random walk theory on graphs, sampling algorithms, and the mixing time. In section IV we report our methodology and results. In section V we outline concluding remarks and future work.

II. Related Work

We will present various sampling techniques in § III, while here we discuss measurements-related work.

Social network measurement: is an evolving area which has been recently explored intensively. Mislove et al. measured several topological characteristics of large scale social networks [20]. Similar study on different graphs is conducted by Ahn et al. in [1]. Community structure of social networks is studied by Leskovec et al. [16]. By Leskovec et al. too, evolution patterns of social graphs are studied in [15], [14] and signed links inference is studied in [13]. Interaction graphs and their measurement are studied in [33] and [34].

Graph mixing time: is interesting an interesting property for different reasons. For example, it is used to quantify the connectivity of graphs [31] since it is bounded by the second largest eigenvalue which is used also for bounding the conductance of the graph, a measure of graph connectivity. Also, as a property the mixing time is used in a wide range of applications, such as Sybil [5] defenses [32], [17], [35], [36], [21], anonymous communication systems on social structures [28], [3], distributed computing [24], and routing [22]. Measuring the mixing time is explored in [27], though graphs for which the mixing time is computed are mostly obtained using the BFS-sampling method known for its bias. While some of these measurement are performed on entire graphs that include complete [1], or close to complete [20], social structures, the majority of these measurements are performed on samples obtained by running the BFS sampling algorithm, which is known for its bias towards high degree nodes.

To our knowledge, we are the first to study and compare a wide range of sampling techniques on estimating mixing time on social networks. We aim to better understand the performances and biases of various sampling techniques, which can guide in selecting the best suitable sampling method given a particular network.

III. Preliminaries

For a simple, undirected, and unweighted graph $G = (V, E)$, where $|V| = n$ and $|E| = m$, we define A as the adjacency matrix where each entry a_{ij} in A is 1 if node v_i is adjacent to node v_j in G and 0 otherwise. From A we define the transition matrix P, where each entry p_{ij} in P is defined as $p_{ij} = 1/\deg(v_i)$ if v_j is connected to v_i and 0 otherwise. A simple random walk on the graph G is a sequence of vertices that begins at a node and ends at another node, where transitions at each step follow the probability in P. This is, p_{ij} describes the probability of moving from node v_i to node v_j. At each time step, a node on the path of the random walk decides to forward the random walk to one of its neighbor uniformly at random.

A. The Mixing Time

Informally, a graph is said to be fast mixing if every random walk on the graph converges to a "stationary distribution" after t steps, where t is in the logarithmic order of n (i.e., $O(\log n)$). The stationary distribution for this graph is defined as an $(1 \times n)$-probability distribution π, where $\pi = [\deg(v_i)/2m]^{1\times n}$. Formally, for the same graph defined on n nodes, the mixing time T parameterized by a statistical distance ϵ (also known as total variation distance—which is in the reciprocal order of n for a fast mixing graph) is defined as $T(\epsilon) = \max_i \min\{t : |\pi - \pi^{(i)} P^t|_1\} < \epsilon$, where π is the stationary distribution above, $\pi^{(i)}$ is the distribution when starting from node v_i and the P is the transition probability defined as in above, while $|\cdot|_1$ is total variation distance, defined as $\frac{1}{2}\sum_j |\pi(j) - \pi^i(j)|$ between two distributions π and π^i. This definition here serves the purpose of our measurement, and more formal definition is provided in [27].

B. Sampling Algorithms

There is a wide range of sampling algorithms in the literature [6], [7], [30], [2], [9], which can be classified into *non-probability*, *equal probability*, and *non-equal probability* sampling algorithms [19]. The non-probability sampling includes breadth-first search (BFS) [10], snow ball sampling, etc, where the probability of each sample being obtained is unknown, thus this type of methods generally suffer from bias issue. On the other hand, the equal probability sampling, e.g., Metropolis-Hastings random walk (RWMH) sampling and random vertex sampling (RVS) has known and equal probability of each sample being taken, whereas for non-equal probability sampling, e.g., reweighting random walk sampling (RWRW) sampling and random edge sampling (RES), the probability of obtaining each sample is known but unequal. The known sampling probabilities of equal and non-equal probability sampling methods enable judicious designs of (asymptotically) unbiased estimators for some graph properties, such as degree distribution, label density, etc [30], [7]. However, it is still unclear for the graph properties, such as mixing time, how the (unbiased) estimator could be designed, and how much bias got introduced by various sampling techniques.

In this paper, we take three representative sampling methods from the above three categories, i.e., BFS, RWRW, RWMH, performing the first systematic study on analyzing and elucidating the biases of estimating mixing time, associated with various sampling techniques. We briefly review these algorithms below.

1) Breadth-First Search Sampling: BFS is one of the simplest algorithms to consider for sampling. In BFS, a starting node is marked as the root, and every neighboring node is explored (at the same level). From that level, every node at the next level is explored, and added to the sample in a breadth-first fashion until the number of nodes explored by this method is equal to the number of nodes required in the sample.

2) Random Walk Sampling: Using the (standard) random walk defined at the beginning of this section one can sample nodes in the graph as follows. Beginning by a starting node v_s, one create a long enough random walk that traverses nodes at random according to the transition probabilities defined earlier. If the current hop of the random walk is not traversed before, the node is added to the set of traversed nodes. The walk is continued until the number of nodes in the traversed set of nodes equals the needed sample size, after which the graph is constructed by establishing all links in the original graph that connect these nodes in the sample.

3) Metropolis-Hastings Random Walk (RWMH) Sampling: The RWMH (with a uniform stationary distribution over all nodes in the graph) is a special type of random walk: after the current node v_i randomly selects a neighbor v_j, it makes the move with probability $\min(1, \deg(v_i)/\deg(v_j))$. In other words, moving towards a lower-degree node is always accepted, while moving towards a higher-degree node may or may not be rejected. RWMH corrects degree bias, and the resulting transition probability from node v_i to node v_j becomes $p_{ij} = \min(\frac{1}{\deg(v_i)}, \frac{1}{\deg(v_j)})$ for $v_j \sim v_i$, $p_{ij} = 1 - \sum_{k \neq i} p_{ik}$ for $v_j = v_i$, and $p_{ij} = 0$ otherwise.

IV. Results and Discussions

In this section we demonstrate the impact of graph sampling, using the three sampling algorithms described earlier, on its mixing time. Before detailing our method used for conducting these measurements, we describe the datasets used in this study.

TABLE I
Datasets used in the measurements and their size.

Dataset	# of Nodes	# of Edges
Physics 1 [14]	$4,158$	$13,422$
Wiki-vote [13]	$7,066$	$100,736$
Physics 3 [14]	$8,638$	$24,806$
Physics 2 [14]	$11,204$	$117,619$
DBLP [18]	$614,981$	$1,155,086$
Youtube [20]	$1,134,890$	$2,987,624$
Flickr [20]	$1,624,913$	$15,476,464$
Facebook [34]	$3,097,165$	$23,667,394$

A. Datasets

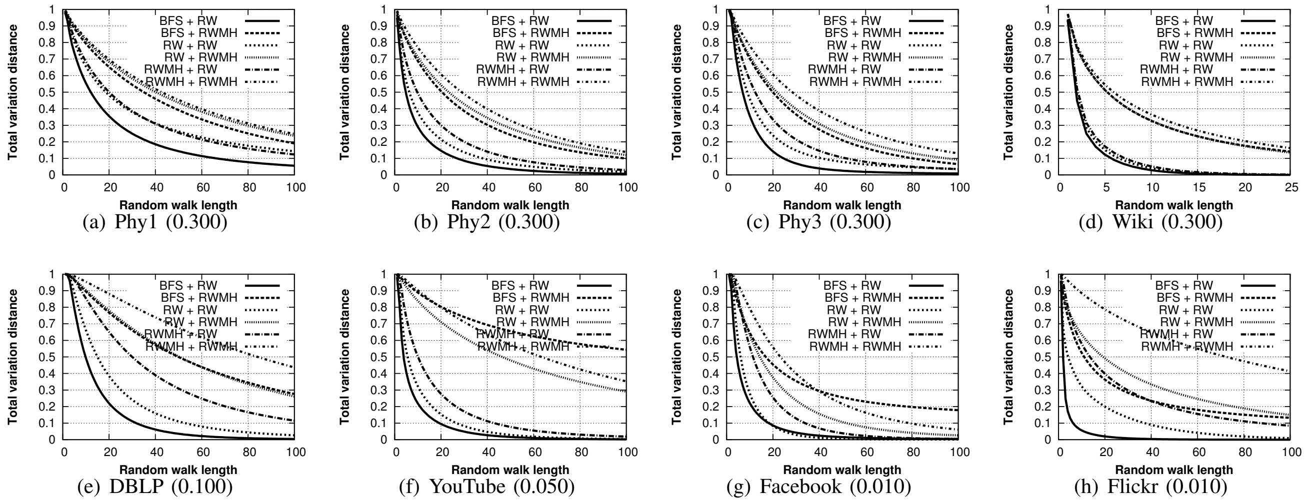
The datasets used in our measurement are previously used in [27] and are for social and information networks. The datasets are shown in Table I with their basic size properties (size). For further details on these datasets, and their characteristics, see [27], or the papers where these datasets were first introduced and used. Worth noting is that the DBLP dataset is as of 2004. Some of these graphs are originally directed, where direction is omitted to simplify measurements and agree with the previous measurements in [27] and how these graphs are used in the literature. Investigating how directionality of edges impact the mixing time is a separate issue worth separate investigation, which we studied in [26].

B. Methodology

A two-process methodology is summarized as follows

1) Sampling Process: Given a dataset, we randomly pick 100 nodes, starting from each of which we initiate each of the three sampling algorithms (i.e. BFS, RW and RWMH), and aggregate the mixing statistics over these starting nodes at the end. For each starting node and each sampler, we try various levels of sampling coverage, corresponding to several samples size: from 0.001 to 0.99 proportion of nodes in the original graph (for large graphs we only sample up to the 0.01 level for feasibility reasons). For each starting node, on each sampler and each level of coverage, we sample a subgraph.

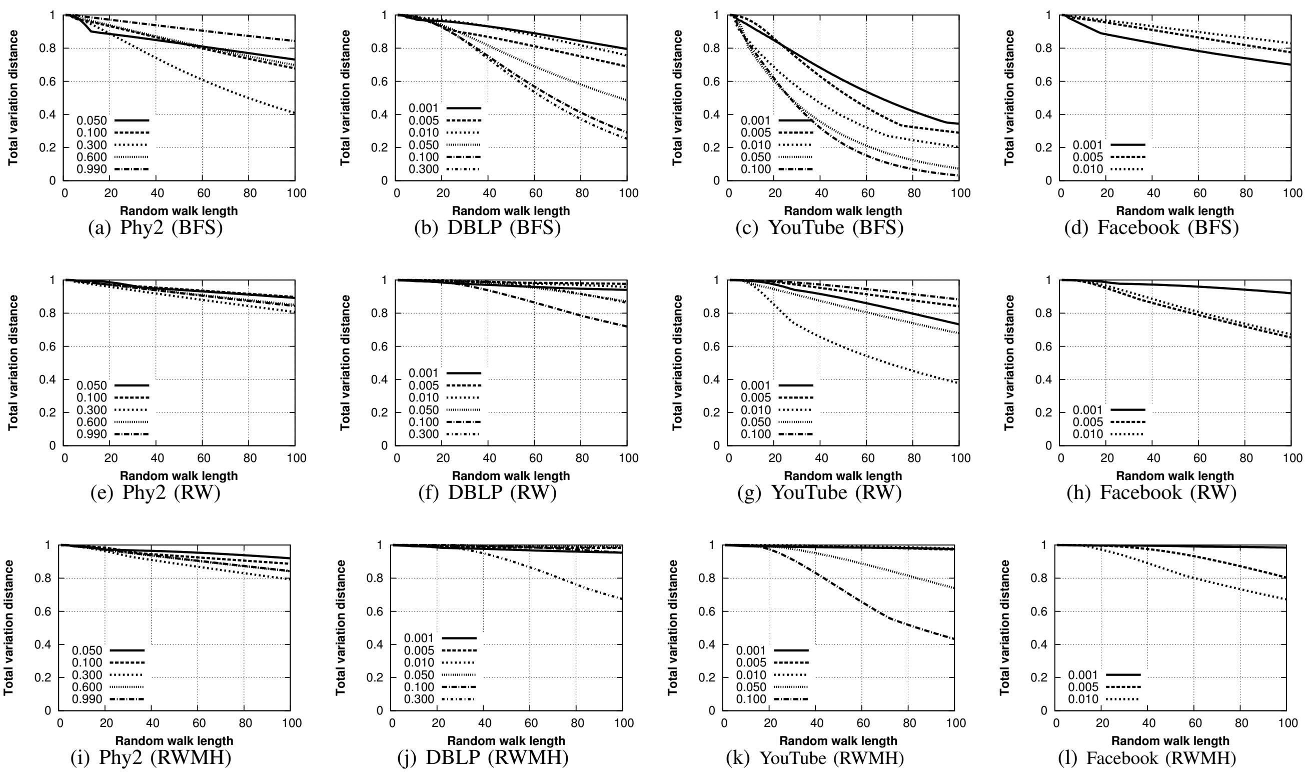
2) Mixing Process: Given a sampled subgraph, we compute its mixing time. By definition one should initiate a random walk from every node in the sampled subgraph in order to accurately calculate the mixing time. Again for feasibility reasons we pick 100 random nodes and initiate each of the two random walk mixing algorithms (i.e. RW and RWMH) from each of the nodes, by iteratively multiplying the corresponding transition probability matrix, up to a random walk length of 500 steps. Our calculation may underestimate the maximum of the mixing times, but should accurately reflect the mean value given that we have sufficient repeats. We calculate the maximum and the mean of the total variation distance for each random walk length, and aggregate them over all starting nodes in the sampling process in the end. For example, by referring to section III-A, we use the definition of the mixing time and using the standard random walk computation (for both the transition probability explained therein and the bounding stationary distribution), we perform the process above so as to compute ϵ for each random walk starting from each of the



(a) Phy1 (0.300) (b) Phy2 (0.300) (c) Phy3 (0.300) (d) Wiki (0.300)

(e) DBLP (0.100) (f) YouTube (0.050) (g) Facebook (0.010) (h) Flickr (0.010)

Fig. 1. *Average* Mixing Times for Different "Sampler+Mixer" Combination of Each Dataset. The number in each subfigure title is the sampling coverage; "BFS+RW", for instance, denotes the combination of BFS sampler and RW mixer.



(a) Phy2 (BFS) (b) DBLP (BFS) (c) YouTube (BFS) (d) Facebook (BFS)

(e) Phy2 (RW) (f) DBLP (RW) (g) YouTube (RW) (h) Facebook (RW)

(i) Phy2 (RWMH) (j) DBLP (RWMH) (k) YouTube (RWMH) (l) Facebook (RWMH)

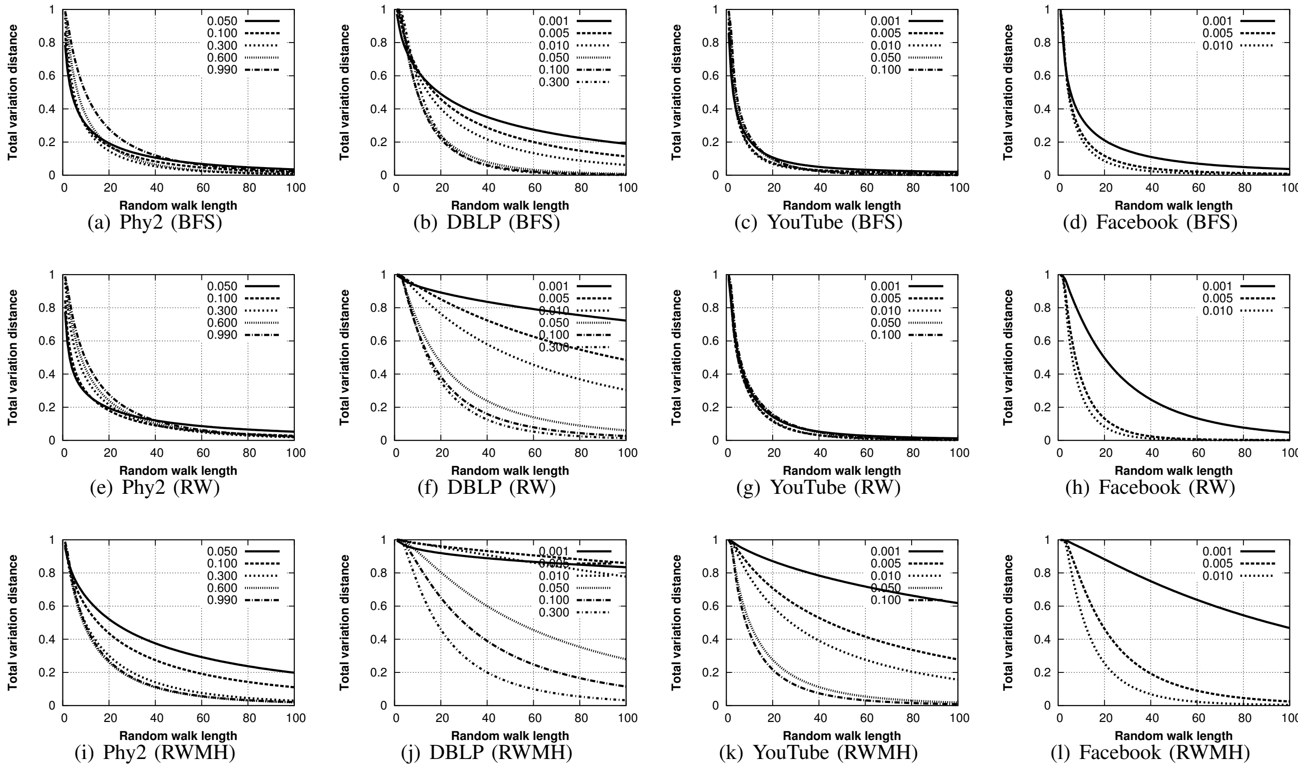
Fig. 2. *Maximum* Mixing Time Estimations for Different Samplers on Various Coverage Levels of Various Datasets. The dataset and the sampler are denoted in each subfigure title. The mixing algorithm is fixed to RW for all cases.

different sources. We repeat this up to the final length of the random walk. Similarly, when using the RWMH algorithm, we only change the stationary distribution (which is uniform over all nodes) and the transition probability as defined earlier.

C. Sampling-Mixing Algorithms Combination

Figure 1 shows the average mixing time for a certain level of sampling coverage (annotated in each subfigure title) on each of the eight datasets shown in Table I, for all combinations of samplers and mixing algorithms. The chosen coverage levels for the four large datasets (i.e. DBLP, YouTube, Facebook and Flickr) are the maximum levels that we actually sampled in our experiments. This figure has the following implications.

First, for the same mixing algorithm (either RW or RWMH), the BFS sampler yields the fastest mixing subgraph (faster than the original graph); and RWMH yields the slowest mixing sample. This qualitative relationship is intuitive: BFS expands slowest and obtains a subgraph biased towards high-degree nodes on which a random walker mixes fastest, while random walk samplers have higher chances of visiting far-away nodes and form a relatively sparse subgraph; compared with the



(a) Phy2 (BFS) (b) DBLP (BFS) (c) YouTube (BFS) (d) Facebook (BFS)
(e) Phy2 (RW) (f) DBLP (RW) (g) YouTube (RW) (h) Facebook (RW)
(i) Phy2 (RWMH) (j) DBLP (RWMH) (k) YouTube (RWMH) (l) Facebook (RWMH)

Fig. 3. *Average* Mixing Time for Different Samplers on Various Coverage Levels of Four Datasets. The dataset and the sampler are denoted in each subfigure title. The mixing algorithm is fixed to RW.

standard RW, RWMH has corrected the bias towards high-degree nodes and therefore yields slower mixing rates. It is worth noting that the RWMH sampler often results in subgraphs mixing slower than the original graph.

Second, for the same sampler (i.e. BFS, RW or RWMH), the standard RW mixing algorithm yields higher mixing rates than the RWMH mixing algorithm.

D. Effects of Sampling Coverage

Figure 3 shows the joint effects of the samplers and the sampling coverage on the average mixing time. We run all three samplers on all feasible levels of sampling coverage on each dataset, fix the mixing algorithm as the standard RW, and report the following findings.

First, from the “Phy2 (RWMH)” subgraph we observe that the RWMH sampler, which is known to be unbiased to the degree distribution, always produces biased estimation of the mixing time, meaning that the mixing time is not associated with the size of the graph for which it is measured.

Second, when the standard random walk is used as the mixing algorithm, BFS operates as better sampler than the RW and RWMH, where BFS would produce the least amount of bias among the three. This can be seen via comparing the three Phy2 subfigures in Figure 3. Also, if we scan across all twelve subfigures, we will see that the combination of the BFS sampler and the RW mixing algorithm tends to yield the least amount of deviations of mixing time on various levels of sampling coverage. This phenomenon is confirmed in the other four datasets, which we omit for the lack of space.

Third, for large-size graphs such as DBLP, YouTube and Facebook (as well as Flickr not shown here), sampled subgraph tends to mix faster as the sampling coverage goes up. However, for small-size graphs such as Phy2 (as well as Phy1/3 and Wiki that are not shown here), a clear monotonic relationship between the mixing rate and the sampling coverage seems not to exist. More precisely, the BFS and the RW tend to sample subgraphs that mix slower when the coverage goes up, contrast to the findings for large-size graphs; while the RWMH sampler behaves in the reverse way. In each case, the ultimate distribution reached as the sample covers the entire graph is the mixing time of the entire graph, though smaller, and originally slower mixing (as reported in [27]) graphs tend to have inconsistent correlation with sample size. One possible explanation for this behavior is the difference in the underlying structure in these graphs, and the size of samples: where very small size of samples from original sparse graphs is obtained in the smaller graphs, relatively larger samples are obtained from original denser graphs in the second case. The larger sample allows for capturing the inherent property of the original graph. Quantitatively verifying this possibility is left to future work.

E. Average versus Maximum Mixing Time

Figure 2 matches Figure 3, but instead it shows the estimations of the *maximum* mixing time, which by definition is the “mixing time” of the entire graph. By comparing the two figures, we can make the following findings. First, the “unbiased” RWMH sampler still produces biased estimation of the maximum mixing time. Second, the relative orders of mixing rates in the corresponding subfigures may not persist. In other words, the mixing times in different subfigures have different patterns that are not captured in a single tendency. Third, since we only pick 100 nodes as the starting points of the random walk for mixing a sampled subgraph, it may underestimate the actual mixing time. Fourth, some curves in some subfigures do not look "smooth". It is because we take the maximum of the mixing times over all 100 different sampled subgraphs, and therefore smoothness is not guaranteed.

V. Conclusion and Future Work

In this paper we comparatively study the bias of three sampling algorithms—the breadth first search, standard random walk, and Metropolis-Hastings random walk sampling—in estimating the mixing time of real-world social graphs. We quantitatively show that sampling algorithms which are unbiased to some graph properties (e.g. degree distribution) may still produce biased estimation of the mixing time. Accordingly, we argue that the bias in the sampling algorithms accepted in the literature is rather metric-dependent. In other words, a given sampling algorithm, while may work nicely and unbiased to one property, may produce large amount of bias in other properties. Unlike degree distribution that is easy to understand and model formally for the amount of bias introduced due to sampling, the bias due sampling on the mixing time has different patterns that are not captured in a single tendency. One consequence of these findings is that earlier work on measuring the mixing time of social graphs by sampling them [27] might fall short in not representing the exact (or good approximate of) mixing pattern in the original unsampled graph. Furthermore, findings of certain applications that use the mixing time as their basic operation property, while sampling their operation graphs, are now questionable and worth further investigation.

This study is entirely measurement-based. In the future, we will look at theorizing the bias in these sampling algorithms for the estimated mixing time over sample in relations with the original graph. While this would be a nontrivial task where potentials for finding firm results are slim, recent work like that in [10]—which considered the bias of BFS on the degree distribution—would be helpful as a starting point.

Acknowledgement— This work was supported by a research grant from KAIST, the NSF grants CNS-0905037, CNS 0917154, CNS-1017647 and CNS-1017092 and the DTRA grant HDTRA1-09-1-0050. The work of A. Mohaisen and Y. Kim was partly done while at the University of Minnesota.

References

[1] Y.-Y. Ahn, S. Han, H. Kwak, S. B. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW*, 2007.

[2] K. Avrachenkov, B. F. Ribeiro, and D. F. Towsley. Improving random walk estimation accuracy with uniform restarts. In *WAW*, 2010.

[3] G. Danezis, C. Díaz, C. Troncoso, and B. Laurie. Drac: An architecture for anonymous low-volume communications. In *PETS*, 2010.

[4] S. Dolev, Y. Elovici, and R. Puzis. Routing betweenness centrality. *JACM*, 57:1–25, 2010.

[5] J. R. Douceur. The sybil attack. In *IPTPS*, 2002.

[6] M. Gjoka, C. T. Butts, M. Kurant, and A. Markopoulou. Multigraph sampling of online social networks. *JSAC*, 2011.

[7] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *INFOCOM*, pages 2498–2506, 2010.

[8] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou. Practical recommendations on crawling online social networks. *IEEE JSAC*, 2011.

[9] M. Kurant, M. Gjoka, C. T. Butts, and A. Markopoulou. Walking on a Graph with a Magnifying Glass. In *ACM SIGMETRICS '11*, 2011.

[10] M. Kurant, A. Markopoulou, and P. Thiran. On the Bias of Breadth First Search (BFS) and of Other Graph Sampling Techniques. In *ITC*, Amsterdam, Sept 2010.

[11] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, 2010.

[12] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Nat. Bur. Standards*, 45(4):255–282, 1950.

[13] J. Leskovec, D. P. Huttenlocher, and J. M. Kleinberg. Predicting positive and negative links in online social networks. In *WWW*. ACM, 2010.

[14] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *KDD*, pages 177–187. ACM, 2005.

[15] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM TKDD*, 2007.

[16] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *CoRR*, abs/0810.1355, 2008.

[17] C. Lesniewski-Lass and M. F. Kaashoek. Whānau: A sybil-proof distributed hash table. In *NSDI*, 2010.

[18] M. Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *SPIR*, 2009.

[19] S. Lohr. *Sampling: design and analysis*. Thomson, 2009.

[20] A. Mislove, M. Marcon, P. K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *IMC*, pages 29–42, 2007.

[21] P. Mittal, M. Caesar, and N. Borisov. X-vine: Secure and pseudonymous routing using social networks. In *NDSS*, 2012.

[22] A. Mohaisen, T. AbuHmed, T. Zhu, and M. Mohaisen. Collaboration in social network-based information dissemination. In *The 28th IEEE International Conference on Communications (IEEE ICC)*, 2012.

[23] A. Mohaisen, N. Hopper, and Y. Kim. Keep your friends close: Incorporating trust into social network-based sybil defenses. In *INFOCOM'11*, pages 336–340. IEEE Press, 2011.

[24] A. Mohaisen, H. Tran, A. Chandra, and Y. Kim. Socialcloud: Using social networks for building distributed computing services. Technical report, University of Minnesota, 2011.

[25] A. Mohaisen, H. Tran, N. Hopper, and Y. Kim. Understanding social networks properties for trustworthy computing. In *ICDCS Workshops*, pages 154–159, 2011.

[26] A. Mohaisen, H. Tran, N. Hopper, and Y. Kim. On the mixing time of directed social graphs and implications. In *ACM ASIACCS*, 2012.

[27] A. Mohaisen, A. Yun, and Y. Kim. Measuring the mixing time of social graphs. In *IMC*, 2010.

[28] S. Nagaraja. Anonymity in the wild: Mixes on unstructured networks. In *PETS*, 2007.

[29] S. Pallottino. Shortest-path methods: Complexity, interrelations and new propositions. *Networks*, 14(2):257–267, 1984.

[30] B. F. Ribeiro and D. F. Towsley. Estimating and sampling graphs with multidimensional random walks. In *IMC*, 2010.

[31] A. Sinclair. Improved bounds for mixing rates of mc and multicommodity flow. *Comb., Prob. & Comp.*, 1, 1992.

[32] N. Tran, J. Li, L. Subramanian, and S. S. Chow. Optimal sybil-resilient node admission control. In *INFOCOM*, 2011.

[33] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On evolution of user interaction in facebook. In *WOSN*, 2009.

[34] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *EuroSys*, pages 205–218, New York, NY, USA, 2009. ACM.

[35] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A near-optimal social network defense against sybil attacks. In *IEEE Symposium on Security and Privacy*, 2008.

[36] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *SIGCOMM*, pages 267–278, 2006.