2011 31st International Conference on Distributed Computing Systems Workshops

Understanding Graph Sampling Algorithms for Social Network Analysis

Tianyi Wang[1], Yang Chen[2], Zengbin Zhang[3], Tianyin Xu[2]
Long Jin[1], Pan Hui[4], Beixing Deng[1], Xing Li[1]
[1] Department of Electronic Engineering, Tsinghua University, China
[2] Institute of Computer Science, University of Goettingen, Germany
[3] Department of Computer Science, U.C. Santa Barbara, USA
[4] Deutsche Telekom Laboratories/TU-Berlin, Germany

***Abstract*—Being able to keep the graph scale small while capturing the properties of the original social graph, graph sampling provides an efficient, yet inexpensive solution for social network analysis. The challenge is how to create a small, but representative sample out of the massive social graph with millions or even billions of nodes. Several sampling algorithms have been proposed in previous studies, but there lacks fair evaluation and comparison among them. In this paper, we analyze the state-of-art graph sampling algorithms and evaluate their performance on some widely recognized graph properties on directed graphs using large-scale social network datasets. We evaluate not only the commonly used node degree distribution, but also clustering coefficient, which quantifies how well connected are the neighbors of a node in a graph. Through the comparison we have found that none of the algorithms is able to obtain satisfied sampling results in both of these properties, and the performance of each algorithm differs much in different kinds of datasets.**

I. Introduction

The last few years have witnessed an explosive growth of online social networks (OSNs) that have attracted most attention from all over the world. Facebook, a social network service, has attracted over 600 million active users as of January, 2011 [1]. Twitter, a social microblogging service known as the "SMS of the Internet", has more than 190 million users who generate more than 65 million "tweets" every day [2]. The huge user base of these OSNs provides an open platform for social network analysis including user behavior measurements [11], social interaction characterization [4], and information propagation studies [10].

However, the huge size of social network graphs hinders researchers from a better understanding of these graphs. On one hand, it is hard to acquire the complete graph of a network. While network administrators are unwilling to provide their data to researchers [5], crawling the complete graph of these social networks is always impossible, especially considering the access rules set by the networks and the amount of time it would take. On the other hand, even with currently available datasets of these networks, processing them requires expensive and well-equipped computer clusters, as well as large time and computation overhead. Alternatively, graph sampling provides an efficient, yet inexpensive solution. By selecting a representative subset of the original graph, graph sampling can make the graph scale small while keeping the characteristics of the original social graph.

Several sampling algorithms have been proposed for graph sampling. Breadth-First Sampling (BFS) [4], [15], [17] and Random Walk (RW) [5], [7] are the most well-known sampling algorithms and have been used in many areas. However, previous studies [5], [15] show that BFS and RW make samples biased to high-degree nodes. Metropolis-Hasting Random Walk (MHRW) [5], [9] is employed to get unbiased samples in undirected social graphs, *i.e.*, keeping the node degree distribution of the original graph unchanged. USDSG [6] makes MHRW applicable in directed social graphs by considering all the unidirectional edges as bidirectional edges. Frontier Sampling (FS) [7] obtains sample graphs with the least mean square error compared with the original graphs. MHRW, USDSG, and FS all compare their algorithm with RW on node degree distribution using different datasets [5]–[7], while the face-to-face comparisons between these newly proposed algorithms is a vacant. Moreover, besides node degree distribution, other graph properties such as clustering coefficient have not been discussed compressively in the existing studies, which limits the scope of potential applications.

In this paper we try to explore how existing algorithms perform in maintaining different important properties of original social graphs. The datasets we choose are all real-world social graphs and have been widely recognized in many other researches. We evaluate these algorithms considering not only node degree distribution (NDD) [5], which has been widely studied, but also clustering coefficient (CC) [8], which is studied in several works [5], [7] considering the average value only. Through this evaluation, we give the first, to the best of our knowledge, comprehensive and relatively fair comparison among the existing sampling algorithms. According to our measurement study, we find that these algorithms perform diversely on maintaining different graph properties. Moreover, the performance is highly correlated with specific dataset. An algorithm can behave quite poorly in some datasets even though it performs quite well in another. We try to get some insights of these performance difference by studying the graph properties. For example, we find both MHRW and FS perform better in tightly connected graphs.

1545-0678/11 $26.00 © 2011 IEEE
DOI 10.1109/ICDCSW.2011.34

The rest of the paper is organized as follows. In Sec. II, we introduce the basic definitions and assumptions, as well as the graph properties we use in the paper. The three popular sampling algorithms are discussed in Sec. III. Finally, we evaluate the performance of algorithms altogether in Sec. IV and conclude this paper in Sec. V.

II. Background

A. Definitions and Assumptions

Social graphs with directional user interactions can be modeled as directed graphs $G_d = (V, E_d)$, where V is a set of nodes (users) and E_d is a set of directional edges (interactions between users). Let $(u, v), u, v \in V$ denote the social edge from node u to node v, k_v^i be the in degree of node v, which is the number of edges $(u, v), u \in V$ in E_d, and k_v^o be the out degree of node v, which is the number of edges $(v, w), w \in V$ in E_d. For some graphs in which the user interactions are undirected, they can be modeled as directed symmetric graphs. That is, $\forall(u, v) \in E_d$, $(v, u) \in E_d$.

While some algorithms require directional information of the edges, sometimes they are not necessary. We can generate symmetric graphs from the directed graphs G_d in these cases. We define $G = (V, E)$ be the symmetric graph of G_d, where

$$E = \bigcup\{(u, v), (v, u)\}, \forall(u, v) \in E_d.$$

We define k_v as the degree of node v in G, which is the number of edges connected with node v.

In the process of sampling, we make some assumptions as follows: (1) We can learn the incoming and outgoing edges at each node. This is true in most OSNs. For example, we can learn the number of followers and followees in the user's profile after we access one node in Twitter. (2) In order to make a fair comparison, we let each algorithm spend the same "cost" to get the sampled graph. We define a unit of cost as an operation to sample or visit a unique node, *e.g.*, downloading a new user's profile spends one unit of cost. This is reasonable because downloading a user's profile is much more time-consuming compared with the calculation to choose the next node. With this assumption we can compare the performance of different sampling algorithms at the same cost.

B. Graph Properties

In this paper, we consider the following two general graph properties:

- **Node Degree Distribution (NDD)**
  NDD is one of the most important properties of a graph. In a directed graph, a node has in degree (or out degree) which is the number of nodes connected to this node by an in (or out) edge. In a social graph node degree represents the number of users that one user interacts with. It is a very important metric for user behavior study. We use θ_k to represent the fraction of nodes with (in or out) degree less than or equal to k. We define the normalized mean square error (NMSE) of node degree k as:
  $$NMSE(k) = \frac{\sqrt{E[(\hat{\theta}_k - \theta_k)^2]}}{\theta_k} \quad (1)$$
  where $\hat{\theta}_k$ is the estimation of θ_k based on the sampled graph. We use $NMSE(k)$ to show the difference between the degree distribution of the sampled graphs and original ones.
- **Clustering Coefficient (CC)**
  CC is a measure of the degree to which nodes in a graph tend to cluster together. The local clustering coefficient for node u in undirected graphs is given by:
  $$C_u = \begin{cases} \frac{2|E_{v,w}|}{k_u(k_u-1)} & \text{if } k_u > 1 \\ 0 & \text{otherwise} \end{cases} \quad (2)$$
  where $E_{v,w}$ is the set of edges among node u's neighbors. The average clustering coefficient is the network average clustering coefficient (NACC) of all nodes in the graphs:
  $$\overline{C} = \frac{1}{n}\sum_{i=1}^{n} C_i \quad (3)$$
  where n is the total number of nodes in the graphs.
  We also define the cumulative clustering coefficient distribution as γ_c, which is the fraction of nodes with local clustering coefficient less or equal to c. The NMSE for cumulative local clustering coefficient distribution is defined as:
  $$NMSE(c) = \frac{\sqrt{E[(\hat{\gamma}_c - \gamma_c)^2]}}{\gamma_c} \quad (4)$$
  where $\hat{\gamma}_c$ is an estimation of γ_c based on the sampled graph.

To quantify the sampling performance of the algorithms on different graph properties, we define a metric called relative error (RE) as follows:

$$RE = \frac{|samples - original|}{original} \quad (5)$$

in which $samples$ represents a property metric of the sampled graph and $original$ is that of the original graph.

III. Analysis on Existing Sampling Algorithms

Generally, current sampling methods can be classified into two categories, *i.e.*, node sampling and edge sampling. As their names suggest, in node sampling the sampling operation is executed on the nodes, while the edges among the sampled nodes remain unchanged. BFS, MHRW, and UGDSG are examples of node sampling methods. On the other hand, edge sampling obtains the sampled graph by sampling the edges of the original graph, and the end nodes of sampled edges are selected. FS is an example of edge sampling.

Among these sampling algorithms, BFS has been widely used in previous studies while MHRW and FS are newly proposed. We do not compare random walk (RW) because

previous work has compared RW with MHRW [5] and FS [7], and RW is proved to perform worse than both.

The process of sampling a graph usually starts from one single seed or multiple seeds. After a node has been sampled, the knowledge of the node's in edges and out edges can be used to choose the next node. The policy of choosing the next node depends on the design of the sampling algorithms. The policies are introduced as following:

Breadth-First Sampling (BFS): BFS is a node sampling algorithm which has been widely studied [14], [15] and applied in user behavior study of OSNs [4], [10], measurement and topological characteristics analysis of OSNs [16], [17]. BFS can find the nodes closest to the initial node and is used to determine distance in graph analysis.

BFS works in the following way. It starts from a randomly selected seed. There are two queues in the sampling process: queue $Sampled$ stores sampled nodes, while queue $Processed$ stores nodes that have been processed. By "processed" we mean sampled or with one of their neighbors sampled. Initially, the seed is stored in queue $Processed$. At each loop, the first node v in queue $Processed$ is moved to queue $Sampled$, and all the neighbors of node v are inserted into queue $Processed$, unless the node has already been processed, *i.e.*, in queue $Processed$. The process loops until the fixed budget is reached. It is possible that the budget is never reached if the initial node locates in a very small isolated subgraph. In this case, another randomly selected seed is inserted to queue $Processed$. Since the nodes in OSNs are usually highly connected with each other, the chance of falling into this case is rare.

It's known that BFS is bias to high degree nodes, as is pointed out by [5]. In BFS, nodes with a higher degree will be visited more frequently. This phenomenon will be shown in Sec. IV and we also show BFS obtains higher local clustering coefficient than the original ones due to the bias.

Metropolis-Hasting Random Walk (MHRW): MHRW is a Markov-Chain Monte Carlo (MCMC) algorithm to obtain random node samples according to the degree probability distribution of the nodes [9]. This is normally difficult to achieve by directly sampling. In MHRW, a proposal function is designed based on the probability distribution. By randomly accepting or refusing the proposal, the proposal function changes the transition probabilities, making the samples converge to the probability distribution.

In this paper, we use MHRW to approximate the uniform distribution because we want the nodes be visited uniformly. Initially, a randomly selected node with non-zero degree is set as the seed. We define the proposal function as $Q(v) = k_v$, which is the degree of node v. From node v's neighbors, MHRW randomly chooses a node w, and then generates a random number p from uniform distribution $U(0, 1)$. If $p \leq Q(v)/Q(w)$, the proposal is accepted and the sampling process will transit to w; otherwise, it stays at node v. Note that if it stays at node v, it does not spend a cost, since the node's profile has been downloaded already. The proposal function changes the transition probabilities in this way: if the degree of w ($Q(w)$) is small, although w will have a small chance to be chosen as the candidate, there will be a high probability that the proposal will be accepted once it happens. Thus the proposal function rectified the bias towards high-degree nodes. MHRW stops when the budget is reached.

MHRW was originally designed for undirected graphs. In [6] a method called USDSG is developed based on MHRW to work in directed graphs. USDSG considers all the unidirectional edges as bidirectional edges. To apply USDSG, we need to change a directed graph G_d to a symmetric graph G. This methodology is also used in Frontier Sampling (FS). Since this is the only difference between MHRW and USDSG, to be simple, we will use term MHRW to represent both the original MHRW and USDSG from now on.

MHRW considers all the duplicated nodes as valid nodes. These duplicated nodes make the node distribution converge to uniform distribution. We do not need to consider the case when we walk to a node with zero degree except for the seed, since the fact that a node can be visited inherently demands that its degree is not 0 [6].

MHRW obtains almost identical degree distribution to the original graphs [5], [6]. However, NMSE of the degree distribution is a little worse than FS (Section IV). Besides, MHRW performs better in well connected graphs than in loosely connected graphs, as it was originally designed for connected graphs [9]. We will show the results in Sec. IV.

Frontier Sampling (FS): FS is an edge sampling algorithm newly proposed in [7] based on RW. It requires a special estimator function to estimate the metric to remove the bias introduced by RW. The process of FS is as follows:

FS firstly randomly chooses a set of nodes, S, as seeds. Then FS will select a seed v from the set of seeds with the probability defined as follows:

$$P(v) = \frac{k_v}{\sum_{u \in S} k_u} \tag{6}$$

An edge (v, w) is selected uniformly from node v's outgoing edges, and v will be replaced with w in the set of seeds and edge (v, w) will be added to the sequence of sampled edges. FS repeats these steps until the budget is reached.

FS requires that at least one of the in degree and out degree of the nodes is not 0. Otherwise the node has neither incoming nor outgoing edges, which means, this node is isolated. In real OSNs the number of isolated nodes is small and in most researches isolated nodes are not considered [5], [7].

FS obtains very good degree distribution of the original graph [7] and the NMSE is the smallest among the three sampling algorithms compared in this paper, as shown in Sec. IV. Besides, it also obtains quite good clustering coefficient distribution according to our evaluation. However, FS does not perform well when the degree or clustering coefficient is small. Moreover, while studying any metric, we need to construct a particular estimator, rather than just studying the sampled nodes and edges directly.

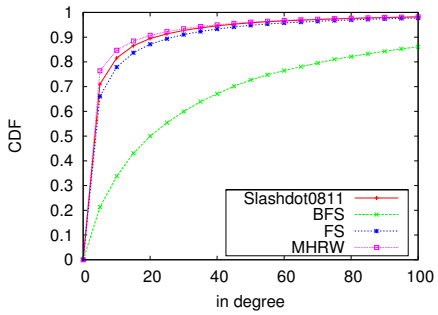

Fig. 1. Slashdot0811 In Degree CDF

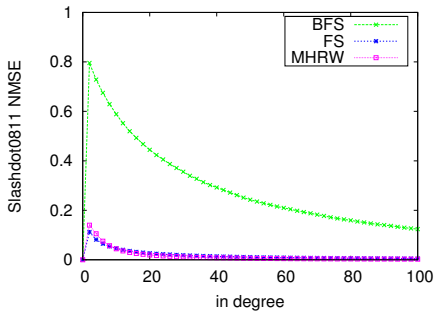

Fig. 3. Slashdot0811 In Degree NMSE

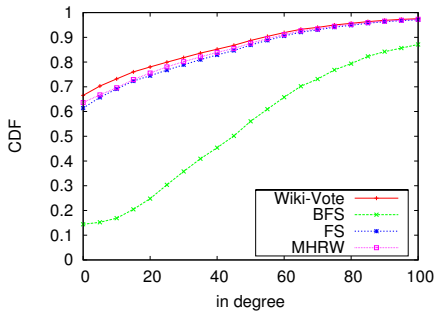

Fig. 2. Wiki-Vote In Degree CDF

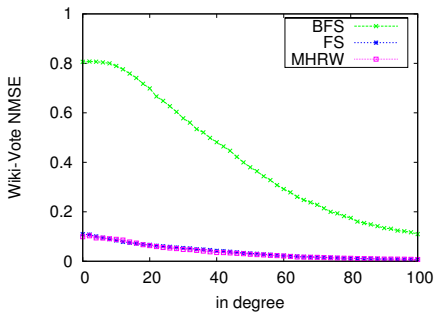

Fig. 4. Wiki-Vote In Degree NMSE

IV. Evaluation

In this section, we evaluate BFS, MHRW, and FS, in terms of NDD and CC. We introduce the datasets we use in this paper, and then analyze the performance of each algorithm on different graph properties. We compute the corresponding properties of the original graphs to use as the ground truth.

A. Dataset

We use four different datasets from Stanford large network dataset collection [13]. This is a collection of datasets of a wide variety of networks, including social networks. (1), (2) **Slashdot Datasets**: Slashdot is a technology-related news website known for its specific user community. The network contains friend/foe links between the users of Slashdot. The Slashdot0811 dataset was obtained in November, 2008 and the Slashdot0902 dataset was obtained in February, 2009; (3) **Wikipedia Dataset**: Wikipedia is a free encyclopedia written collaboratively by volunteers all over the world. In Wikipedia a user needs to be voted to become an administrator. The Wiki-Vote dataset contains all the users and discussion from the inception of Wikipedia till January, 2008. Nodes in the network represent Wikipedia users and a directed edge from node i to node j represents that user i voted on user j [13]; (4) **Epinions Dataset**: The soc-Epinions1 dataset is from a who-trust-whom network of a general consumer review site Epinions.com. A user can decide whether to "trust" another user in the website. The trust relationships form the web of trust and decide which reviews are shown to the user combined with review ratings.

Table I shows some basic information of datasets we use in this paper. We show the number of nodes and edges in the original graphs and the network average clustering coefficient (NACC). We also use Strongly Connected Components (SCC) to show the connectivity of the original graphs, which is the fraction of number of nodes in the largest strongly connected component. SCC shows the connectivity of a graph: if the value of SCC is larger, the graph is more tightly connected. We will show that connectivity greatly affects the performances of sampling algorithms. Among these datasets, Slashdot0811 and Slashdot0902 are more tightly connected while the other two are more loosely connected. This will help us to evaluate how connectivity affects the performance of sampling algorithms.

TABLE I
DATASET INFORMATION

	Nodes	Edges	NACC	SCC
Slashdot0811	77360	905468	0.055	0.909
Slashdot0902	82168	948464	0.060	0.868
Wiki-Vote	7115	103689	0.141	0.183
soc-Epinions1	75879	508837	0.138	0.425

B. Property Analysis

1. Node Degree Distribution: We use cumulative distribution function (CDF) and normalized mean square error (NMSE) to show the performances of the sampling algorithms in keeping node degree. The degree distribution of the original graphs is obtained in advance. We count the number of nodes to

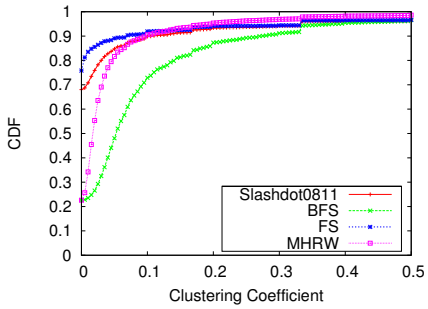

Fig. 5. Slashdot0811 Clustering Coefficient CDF

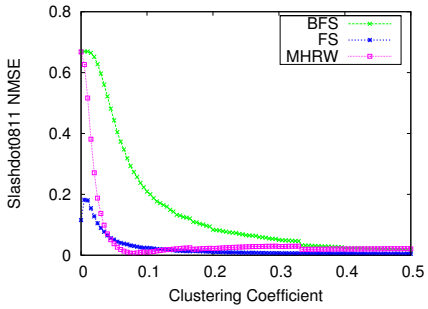

Fig. 7. Slashdot0811 Clustering Coefficient NMSE

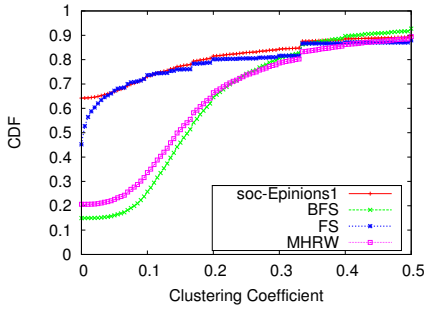

Fig. 6. soc-Epinions1 Clustering Coefficient CDF

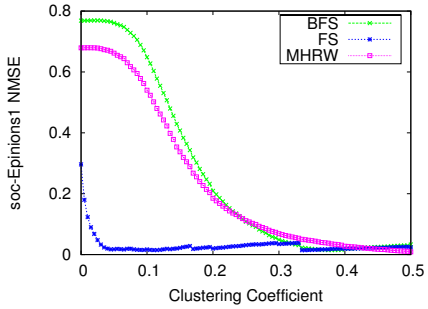

Fig. 8. soc-Epinions1 Clustering Coefficient NMSE

a certain degree k and calculate the fractions θ_k. For algorithms BFS and MHRW, we first need to recover a subgraph $G_s = (V_s, E_s)$ from the sampled nodes, where V_s is the set of sampled nodes, E_s is the set of edges among the sampled nodes, *e.g.*,

$$E_s = \bigcup\{(u,v) \in E_d\}, u \in V, w \in V$$

From the subgraph G_s we can get the degree distribution of the sampling algorithms. For FS, The process is much more complicated. An estimator function is used to estimate the degree distribution [7]. According to the strong law of large numbers, the estimator will converge to the real value, if the budget is large enough. The sampled edges are the input of the function and the estimated degree distribution is the output. Let the sampled edges obtained through FS be set $E_s = (u_i, v_i), i = 1, ..., B$. Then we get the following estimator for θ_k:

$$\hat{\theta}_{k^i} = \frac{1}{SB} \sum \frac{1(k^i_{v_i} \leq k^i)}{k_{v_i}}, i = 1, ..., B \tag{7}$$

where

$$S = \frac{1}{B} \sum \frac{1}{k_{v_i}}, i = 1, ..., B$$

$\hat{\theta}_{k_i}$ is an estimator of θ_{k_i}, which the fraction of nodes with in degree less or equal to k^i.

The calculation of NMSE is given by equation (1) in Sec. II. In order to get the expectation in the equation, we run the algorithms for 100 times and first get the average of $(\hat{\theta}_k - \theta_k)^2$ to represent its expectation.

Fig. 1 and Fig. 2 plot the CDF of in degree of datasets Slashdot0811 and Wiki-Vote. Fig. 3 and Fig. 4 plot the NMSE of in degree distribution of these two datasets. The results of out degree are similar and are not shown due to space limit.

From these figures, we can see BFS is biased to high degree nodes significantly. NMSE of BFS is very large compared with MHRW and FS. This is the same results with previous study. From CDF plot we can see both MHRW and FS are almost identical to the original ones. In this sense both obtain good degree distribution of the original graphs. From Fig. 3 and Fig. 4, we also see both MHRW and FS quickly converge to almost 0. All the figures show the good performance of MHRW and FS in keeping node degree distribution.

Connectivity affects both MHRW and FS greatly considering node degree distribution. NMSE of both algorithms are smaller and converge faster in Slashdot0811 than in Wiki-Vote. Notice that Slashdot0811 is more tightly connected than Wiki-Vote. We can conclude that MHRW and FS perform worse in more loosely connected graphs. We confirm this by evaluating the other two datasets.

2. Clustering Coefficient: To evaluate clustering coefficient, we first compare NACC obtained through these algorithms. NACC of the original graphs is calculated as described in Sec. II. For both BFS and MHRW, we get the sampled graph G_s through the sampled nodes and then calculate the average clustering coefficient of G_s. For FS an estimator function is needed. the estimator function [7] is given by:

$$\hat{C} = \frac{1}{SB} \sum \frac{2f(u_i, v_i)}{k_{v_i}(k_{v_i} - 1)} \frac{1}{k_{v_i}}, i = 1, ..., B \tag{8}$$

where

$$S = \frac{1}{B}\sum \frac{1}{k_{v_i}}, i = 1, ..., B$$

$\hat{C}$ is an estimator for average clustering coefficient and $f(u, v)$ gives the number of common neighbors between node u and v. We show NACC and relative error in Table II.

TABLE II
CLUSTERING COEFFICIENT

	Slashdot0811 NACC (RE)	Slashdot0902 NACC (RE)	Wiki-Vote NACC (RE)	soc-Epinions1 NACC (RE)
original	0.0555	0.0603	0.141	0.138
BFS	0.106 (91.0%)	0.112 (85.7%)	0.350 (148.2%)	0.211 (52.9%)
MHRW	0.0504 (9.19%)	0.0523 (13.3%)	0.218 (54.6%)	0.196 (42.0%)
FS	0.0479 (13.7%)	0.0553 (8.29%)	0.0788 (44.1%)	0.158 (14.5%)

We can see that for all four datasets NACC obtained through BFS is significantly larger than the original one. Clustering coefficient is considered to strongly depend on node degree k_v [5]. Since BFS is biased to high degree nodes, it obtains larger average clustering coefficient. The performance of MHRW and FS greatly depends on the connectivity of the datasets. In tightly connected datasets (Slashdot0811 and Slashdot0902) the relative error is small compared with loosely connected datasets (Wiki-Vote and soc-Epinions1). The comparison between MHRW and FS is not very clearly through average clustering coefficient. MHRW performs better in datasets Slashdot0811 while FS performs better in the other three datasets. However, the difference is not very big between these two algorithms.

To further study the performance of these sampling algorithms, we try to plot the CDF and NMSE of local clustering coefficient. The definitions are given in Sec. II. The estimator of FS for $\hat{\gamma}_c$ is given by:

$$\hat{\gamma}_c = \frac{1}{SB}\sum \frac{1(\hat{c} \leq c)}{k_{v_i}}, i = 1, ..., B \quad (9)$$

where

$$\hat{c} = \frac{2f(u_i, v_i)}{k_{v_i}(k_{v_i} - 1)}$$

and

$$S = \frac{1}{B}\sum \frac{1}{k_{v_i}}, i = 1, ..., B$$

Fig. 5 and Fig. 6 plot the cumulative clustering coefficient distribution and Fig. 7 and 8 plot the NMSE. BFS tends to have larger clustering coefficient due to the bias to high degree nodes. However, it is very close to the original one when the clustering coefficient is large enough (about 0.35 in the datasets). In Fig. 5 MHRW is very close to the original one and NMSE is small, as shown in Fig. 7. However, the difference between MHRW and the original graph is much larger in Fig. 6 because soc-Epinions1 is loosely connected. The clustering coefficient distribution obtained through FS is quite good except when clustering coefficient is very small (about less than 0.05). This explains why the average clustering coefficient got through FS is not good. The fraction of nodes with small clustering coefficient is large, thus generating more errors during the sampling.

V. CONCLUSIONS AND FUTURE WORK

In this paper we conducted a comprehensive study on several sampling methods in social graphs. We analyzed how these sampling methods perform in maintaining the properties of the original graphs. For existing sampling algorithms, BFS is biased to high degree nodes and obtains larger average clustering coefficient. Both MHRW and FS keep the degree distribution well. In terms of clustering coefficient, the performances of MHRW and FS highly depend on the datasets: both work better in tightly connected graphs while FS converge faster and is more accurate than MHRW.

ACKNOWLEDGMENTS

This work is supported by the National Basic Research Program of China (No. 2007CB310806) and the National Science Foundation of China (Nos. 60850003,60473087). We thank Mr. Cong Ding from University of Goettingen for his comments and suggestions.

REFERENCES

[1] MSNBC, "Goldman to clients: Facebook has 600 million users," http://www.msnbc.msn.com/id/40929239/ns/technology-and-science-tech-and-gadgets/.
[2] Quantcast.com, "Twitter.com - Quantcast Audience Profile," http://www.quantcast.com/twitter.com.
[3] J. Leskovec and C. Faloutsos, "Sampling from Large Graphs," In Proc. of ACM SIGKDD, 2006.
[4] C. Wilson, B. Boe, A. Sala, K. P. N. Puttaswamy, and B. Y. Zhao, "User Interactions in Social Networks and their Implications," In Proc. of ACM EuroSys, 2009.
[5] M. Gjoka, M. Kurant, C. T Butts, and A. Markopoulou, "Walking in Facebook: A Case Study of Unbiased Sampling of OSNs," In Proc. of IEEE INFOCOM, 2010.
[6] T. Wang, Y. Chen, Z. Zhang, P. Sun, B. Deng, and X. Li, "Unbiased Sampling in Directed Social Graph," In ACM SIGCOMM Computer Communication Review, 40(4):401-402, 2010.
[7] B. Ribeiro and D. Towsley, "Estimating and Sampling Graphs with Multidimensional Random Walks," In Proc. of ACM IMC, 2010.
[8] D. J. Watts and S. Strogatz, "Collective Dynamics of 'Small-World' Networks," Nature 393(6684): 440-442, 1998.
[9] Minas Gjoka, "Measurement of Online Social Networks," UC Irvine PhD Thesis, 2010.
[10] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a Social Network or a News Media?" In Proc. of WWW, 2010.
[11] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing User Behavior in Online Social Networks," In Proc. of ACM IMC, 2009.
[12] A. Sala, L. Cao, C. Wilson, R. Zablit, H. Zheng, and B. Y. Zhao, "Measurement-calibrated Graph Models for Social Network Experiments," In Proc. of WWW, 2010.
[13] Stanford Large Network Dataset Collection. http://snap.stanford.edu/data/index.html.
[14] M. Kurant, A. Markopoulou, P. Thiran, "On the Bias of Breadth First Search (BFS) and of Other Graph Sampling Techniques," International Teletraffic Congress, 2010.
[15] S. H. Lee, P. -J. Kim, and H. Jeong, "Statistical Properties of Sampled Networks," Physical Review E, 2006.
[16] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," In Proc. of ACM IMC, 2007.
[17] Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of Topological Characteristics of Huge Online Social Networking Services," In Proc. of WWW, 2007.