Graph Characteristics Estimation via Random Walk Sampling

Phil H. Cui
Department of Electrical and Computer Engineering
Drexel University

Researchers have great interest on Online Social Network (OSN) since it contains plenty of interesting and valuable patterns which reflect human interaction and behaviors. However, many obstacles impede the intension to directly abstract the information from the graph of OSN for the following reasons: First, dataset suppliers may have limitations on accessing the complete graph, which makes it impossible to obtain the full graph. Second, it is hard to store locally the complete graph considering its size. Besides, even if the full graph is obtainable, it consumes a lot of time and requires expensive computers to crawl on it. Finally, crawling on the original graph may involve the privacy issue. Therefore, it is urgent to find a feasible, efficient and inexpensive solution to the above difficulties. And graph sampling points out a promising way by creating a representative subgraph in small scale while preserving the targeted properties of the original massive graph.

The report studies three papers which apply random walk for graph sampling. Random walk sampling has the advantage that it does not need the knowledge of the complete network topology. Instead, it acquires the samples by crawling on the graph only through the public interfaces. And the graph properties can be estimated based on the samples from the random walk. The first paper uses a single random walker[1] to estimate network size and clustering coefficient. The second paper introduces Frontier Sampling (FS)[2] which is a multi-dimensional version of random walk by coordinating multiple random walkers to sample the edges uniformly at random. FS can also be fully distributed with each walker sampling the graph independently and without any coordination cost. The third paper suggests a different type of random walk called Subgraph Random Walk (SRW)[3] which samples the targeted type of small connected subgraphs (motifs, graphlet or Connected Induced Subgraphs(CISs)) and estimates their concentrations.

I. Graph Characteristics

Graph Characteristics include but not limit to network size, Degree Distribution (DD)[1], Clustering Coefficient (CC)[1], Assortative Mixing Coefficient(AMC)[2], frequency of graphlets[4] and concentration of motifs [3]. Network size (N) is the number of nodes of the network. CC quantifies how well connected are the neighbors of a node in a graph. AMC represents a preference for a network's nodes to attach to others that are similar in some way. It is generally defined as the Pearson Correlation of similarity between neighboring nodes and node degree can be used as the similarity measure. Motifs consist of $k(k \ll N)$ nodes with at least k-1 edges. Motif sampling is applied for pattern recognition, gene expression profiling[5], protein-protein interaction prediction[6] and coarse-grained topology generation[7].

II. Graph Sampling Algorithms

The existed graph sampling algorithms for estimating the graph characteristics can be roughly classified into three categories: (1) Random Node (RN) sampling and Random Edge (RE) sampling; (2) Traversal based samplings and (3) Random Walk (RW) samplings. RN and RE sample the nodes and edges uniformly at random respectively. Graph traversal techniques visit each node exactly once while RW allows node to be revisited. The advantage of RW is that it can crawl on the graph without knowing the complete graph topology. So RW provides a solution for estimating graph properties when the complete graph is not available or when the graph is too large to be stored locally. In addition, the bias of RW can be quantified and corrected compared to that of the traversal based algorithms which is hard to be analyzed.

A. Random Node Sampling and Random Edge Sampling

Random Node (RN) sampling algorithm[8] selects a set of nodes uniformly at random with the attached edges unchanged. Random Edge (RE)[8] sampling algorithm selects edges uniformly at random together with the corresponding end nodes. Intuitively, RE samples a sparsely connected graph. The combined version of RN and RE is called Random Node-Edge (RNE) sampling[8], which first selects a node uniformly at random and then picks an edge incident to the node uniformly at random. Random PageRank Node (RPN) sampling[8], [9] samples a node with the probability proportional to its PageRank. And Random Degree Node (RDN) sampling[8] selects a node with probability proportional to its degree. RPN and RDN are non-uniform sampling algorithms.

B. Traversal Based Sampling

Breadth First Sampling (BFS) is a node sampling algorithm by selecting nodes closest to the initial node. It is biased toward high degree nodes[10], [11] and therefore has a higher local clustering coefficient than that of the original graph.

Moreover, the bias introduced by BFS has not been analyzed so far.

Forest Fire (FF)[12] starts from a seed node and burns the outgoing edges and the corresponding nodes. The node at the other end of the burned edge is able to burn the edges attached to it. And the burning process proceeds recursively until the conflagration dies out.

Snow Ball Sampling (SBS) is very similar to BFS. BFS exhaustively expand the neighborhood of current node while SBS only expands a fixed number of them.

C. Random Walk Sampling

1) Sampling by a Single Random Walker: Random Walk (RW) Sampling[13], [8], [1] starts from a randomly selected node and then walks on the graph by selecting the neighboring nodes uniformly at random. And the random walker flies back to the starting point with probability usually set as 0.15. As shown in [14], RW is biased toward high degree nodes. The bias of RW can be quantified by Markov Chain analysis and thus can be corrected by Re-Weighted Random Walk (RWRW)[15], [16], [17] to sample the nodes uniformly at random. RWRW re-weights the measured values using Hansen-Hurwitz estimator[18] after the walk.

Metropolis Hasting Random Walk (MHRW)[19], [20], [10] can also eliminate the bias of RW by modifying the transition probabilities. MHRW can sample nodes according to an arbitrarily specified node degree distribution after convergence. MHRW is usually applied to sample the uniform distribution with the nodes being visited uniformly at random. Specifically, MHRW uses a proposal function (function of degree of nodes) to rectify the bias toward high-degree nodes in the way that nodes with small degrees have more chances to be accepted although they are less likely to be selected compared to those nodes with high degrees.

MHRW obtains almost identical degree distribution to the original graph[10], [21], [11](advantage of uniform node sampling). However, since it was designed for connected graphs [11], MHRW performs not well on loosely connected ones. The technique studied in one of the three papers called Frontier Sampling (FS)[2] can improve on this problem.

2) Sampling by Multiple Random Walkers: Frontier Sampling (FS) [2] is a multidimensional version of random walk sampling by applying multiple random walkers to sample the edges uniformly at random. FS starts from a set of seed nodes which are randomly selected, and then selects a node with the probability proportional to its node degree. After that, FS uniformly sample an edge from the selected node's outgoing edges. The node at the other end of the edge will replace the old one as the seed node. The above process will repeat until the sampling budget is reached. FS can be made fully distributed with each walker sampling the graph independently and without any coordination cost.

3) Sampling Motifs by Random Walk: Prior to the algorithm Subgraph Random Walk (SRW) proposed by Wang et al, RE is applied by Kashtan et al. [22] to sample the motifs. It turns out that RE has a heavy bias and scales poorly with the motif size. Wernicke tried to enumerate subgraph trees[23] by RN, however, RN is not supported by most OSNs and is too resource intensive. Bhuiyan et al. proposed the algorithm called GUISE[4] which is based on MHRW to estimate the frequency of motifs with sizes three, four and five on a compounded motif relationship graph. GUISE rejects the targeted motifs due to the algorithm design of Metropolis-Hasting and therefore has a larger estimation error for estimating the concentration of the targeted motif compared to SRW. Furthermore, GUISE can only sample motifs with size three, four and five.

SRW[3] samples the targeted type of motifs without rejection. The motif visited by SRW on the CIS relationship graph will decide which node on the original graph to be queried. SRW does not need the complete network topology information and requires less samples than its competitive algorithms but still acquires the same accuracy. Specifically, motif sampling starts from initializing a motif by querying the corresponding nodes on the original graph. After that, both the starting motif and all its neighboring motifs on the CIS relationship graph will be available. Then the random walker picks a neighboring motif uniformly at random as the next stop. Since the selected motif include one (and the only one) node which has not been queried, it go back to the original graph and query that node. After the new node is queried, more motifs which are the neighbors of the existed motifs are available. A key observation is that the motif graph is not known in advance. The motifs and their inter-connections cannot be determined until the corresponding node on the original graph is accessed. However, this does not affect walking on the motif graph since the neighbors of the current motif is accessible. We can imagine the motif graph consists of the unlighted bulbs. After the node on the original graph is queried, the bulbs on the motif graph are illumined correspondingly. As we can see that, there is an interaction between the original graph and the motif graph, i.e., the future nodes to be queried on the original graph G are actually determined by the randomly selected motif on the motif graph $G^{(k)}$.

Pairwise Subgraph Random Walk (PSRW) is an upgraded version of SRW and has a less estimation error than that of SRW. To estimate the concentration of the targeted motif of size k, PSRW samples motifs of size $k-1$ by random walking on the $k-1$ node motif graph $G^{(k-1)}$ instead of the k node motif graph $G^{(k)}$. And the k-node motif is sampled by PSRW if and only if at least one of its associated edges on $G^{(k-1)}$ is sampled.

D. Brief Summary of Graph Sampling Algorithms

Among the aforementioned graph sampling algorithms, RN, RPN and FF do not bias towards high degree nodes[8]. RDN, RW and BFS samplings[10], [8] do have bias toward densely connected parts of the graph (Note that RW results in uniform edge distribution on an undirected graph [24]). And RE tends to have a sparsely connected sample with the high degree nodes under-represented. MHRW rectifies the bias of RW and samples nodes uniformly at random. FS samples edges uniformly at random and can mitigate the estimation errors in the presence of disconnected subgraphs. GUISE is designed to estimate the frequency of different types of motif with sizes three, four and five. SRW and PSRW are good at estimating the concentration of the targeted motif on the CIS relationship graph whose motifs are with the same size. SRW and PSRW sample the targeted motif without rejection and therefore achieve a better estimation accuracy than its competitive algorithm Metropolis-Hastings Subgraph Random Walk (MHSRW) which is a tailored version of GUISE.

E. Papers Selected for the Report

The three papers studied in the report all apply random walk strategies to sample the graph. It ranges from sampling by a single random walker[1], sampling by multiple random walkers[2] and sampling the motifs by random walking on the CIS relationship graph[3].

The algorithm proposed by Hardiman et al. uses a single random walker[1] to estimate network size and clustering coefficient. It requires only external access through publicly available interface and a relatively small number of nodes locate at the frontier of previously explored nodes. Instead of exploring the Ego network which requires querying the immediate neighbors of nodes through random walk, the algorithms in the paper access only nodes acquired by random walk and achieve a better estimation accuracy than such Ego network algorithms as FS [2] and MHRW[11] on estimating network average clustering coefficient, global clustering coefficient and network size. The paper also suggests to use neighbor collision rather than node collision to estimate network size, the advantage of which is supported by both the theoretical and empirical analysis.

FS achieves a smaller Normalized Mean Squared Error (NMSE) than RW, BFS and MHRW[11], [10], [2], especially in the presence of disconnected or loosely connected graphs. FS acquires an accurate estimation of AMC, CC and DD. FS also performs better than RN for estimating the tail of degree distribution. In addition, FS can be fully distributed with each walker sampling the graph independently without any coordination cost.

SRW and PSRW[3] proposed by Wang et al. represent the network as a CIS relationship graph whose nodes are motifs with the same size. The algorithms run a random walk on the CIS relationship graph without the knowledge of the network topology. It turns out that PSRW uses less samples than other algorithms but converges faster with more accurate estimates.

III. Estimating Clustering Coefficients and Size of Social Networks via Random Walk

The size of the current networks is so large that directly computing such network measures as CC and network size is compromised. Therefore, the paper provides algorithms to estimate CC and network size. The paper provides estimators for CC which measures network connectivity in both global and network average ways. It also gives network size estimation using neighbor collision estimator which proves to have less variance than node collision estimator.

The algorithms in the paper require only external access through publicly available interface and access the graph by random walking through a relatively small number of nodes which locate at the frontier of previously explored nodes. Instead of exploring the ego network which requires querying immediate neighbors of nodes through random walk, the algorithms in the paper access only nodes acquired by random walk and achieve a better estimation accuracy.

A. Definitions

Local Clustering Coefficient (LCC):
LCC represents the connectivity centering around node v_i. From the standpoint of social network, it tells how popular is the phenomenon that someone's friends are also friends to each other.

Specifically, the local clustering coefficient of node v_i, denoted by c_i, is defined as the ratio of the existed number of triangles surrounding a node v_i to the maximum number of possible triangles surrounding v_i. Mathematically,

$$c_i = \frac{l_i}{C_{d_i}^2} = \frac{2l_i}{d_i(d_i-1)}$$

where d_i is the degree of v_i or the number of neighbors of v_i and $C_{d_i}^2 = \binom{d_i}{2} = \frac{d_i(d_i-1)}{2}$. Let $A_{i,k}, A_{k,i}$ and $A_{j,k}$ be the elements of the adjacency matrix $A_{n\times n}$ which describes the connectivity of graph G with a total number of nodes n, i.e., $A_{i,k} = A_{k,i} = 1$ if node v_i connects to node v_k and 0 otherwise (notice that $A_{i,i} = 0$). According to this definition, $l_i = \sum_{j<k} A_{i,j}A_{i,k}A_{j,k}$ is the number of edges between neighbors of v_i and it is also the number of triangles surrounding v_i.

Network Average Clustering Coefficient (NACC) :
The network average clustering coefficient $\mathfrak{c}_l$ is the sample mean of LCC:

$$\mathfrak{c}_l = \frac{1}{n}\sum_{i=1}^{n} c_i$$

Global Clustering Coefficient (GCC) :
The global clustering coefficient, denoted by $\mathfrak{c}_g$, is the ratio of the total number of existed triangles of a graph G to the number of all possible triangles of G. Mathematically,

$$\mathfrak{c}_g = \frac{\sum\limits_{i=1}^{n} l_i}{\sum\limits_{i=1}^{n} C_{d_i}^2} = \frac{2\sum\limits_{i=1}^{n} l_i}{\sum\limits_{i=1}^{n} d_i(d_i-1)}$$

Mixing Time:
The mixing time $\tau(\epsilon)$ is defined as the minimum number of random walk steps needed for reaching the stationary distribution, i.e. $\tau(\epsilon) = \min\ \{r | d(r) \leq \epsilon\}$, where $d(r) = \max\limits_{x_1=1}^{n} \max\limits_{i=1}^{n} |p_i - \mathbf{P}(x_r = i)|$, $p_i = \frac{d_i}{D}$, $D = \sum\limits_{i=1}^{n} d_i$, and ϵ is a small constant which acts as the threshold and r is the number of random walk steps. As pointed out in the paper, the number of steps for $d(r) \approx 0$ is $r = \log^2 n$ for Facebook, $r = 3\log^2 n$ for DBLP and Youtube networks and $r = 10\log^2 n$ for the Live Journal network.

B. Network Size Estimation

We rearrange the nodes sampled by random walk along a line in the order of the walking steps as Figure 1 shows. As denoted by the paper, x_k and x_l are the vertices sampled at step k and step l. Note that even if k and l are very far apart, the vertices x_k and x_l could be very close on the graph G.

To ensure that nodes x_k and x_l can be drawn from the stationary distribution, their distance should be large enough, i.e., $|k - l| \geq m$, where m need only be of the order $O(\log^2 n)$ because of the fast-mixing nature of social graphs as mentioned before. The following variable I is defined to count the number of node pairs whose index distance is greater than a threshold m:

$$I = \{(k,l)||k-l| \geq m, 1 \leq k,l \leq r\}$$

Actually, the larger the value of m, the smaller the bias of the estimation introduced by the correlation. However, in that case, fewer node pairs will be acquired and a larger estimator variance will be introduced by large value of m.

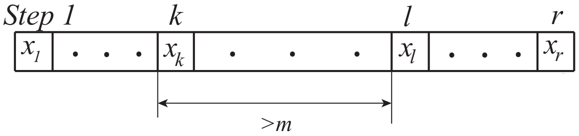

Fig. 1: Rearrangement of Vertices of the Original Graph G

1) Neighbor Collision Based Network Size Estimator: A neighbor collision is a pair of indices (k,l) such that vertices x_k and x_l share a common neighbor. To count the number of common neighbors between vertices x_k and x_l, denoted by $\phi_{k,l}$, define A_k as the set of nodes adjacent to x_k. Then $\phi_{k,l} = |A_k \cap A_l|$.

Notice that there are two ways to count the number of triplets of graph G. One is to count based on each single node v_k: N(triplets)=$\sum_{k=1}^{n} d_k^2$. The other way is to count based on a pair of nodes v_i and v_j: N(triplets)=$\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n}|A_i \cap A_j|$. Therefore,

$$\sum_{i=1}^{n}\sum_{j=1}^{n}|A_i \cap A_j| = \sum_{k=1}^{n} d_k^2 \tag{1}$$

The right side of equation 1 eliminates the summation over n which allows to estimate n by carefully designing a companion estimator Ψ_n as we will see it later. Based on equation 1, the mean of the weighted neighbor collisions $\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}} = |A_{x_k} \cap A_{x_l}|\frac{1}{d_{x_k}d_{x_l}}$ is then

$$\begin{aligned} E[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}] &= \sum_{x_k=1}^{n}\sum_{x_l=1}^{n}\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}\frac{d_{x_k}}{D}\frac{d_{x_l}}{D} \\ &= \sum_{x_k=1}^{n}\sum_{x_l=1}^{n}|A_{x_k} \cap A_{x_l}|\frac{1}{D^2} \\ &= \sum_{j=1}^{n}(\frac{d_j}{D})^2 \end{aligned} \tag{2}$$

Now, define Φ_n as the sample means of the weighted neighbor collision $\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}$ over all possible pairs $(k,l) \in I$:

$$\Phi_n = \frac{1}{|I|}\sum_{(k,l)\in I}\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}} \tag{3}$$

According to the strong law of large numbers (SLLN), the sample means Φ_n converges to $E[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}]$ almost surely. On the other hand, since

$$\begin{aligned} E[\Phi_n] &= \frac{1}{|I|}\sum_{(k,l)\in I}E[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}] \\ &= \frac{1}{|I|}|I|E[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}] \\ &= E[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}] \end{aligned} \tag{4}$$

we know that Φ_n converges to its mean $E[\Phi_n]$ almost surely.

Next, design the companion estimator Ψ_n to be the sample means of $\frac{d_{x_k}}{d_{x_l}}$ over all possible choices of $(k,l) \in I$:

$$\Psi_n = \frac{1}{|I|}\sum_{(k,l)\in I}\frac{d_{x_k}}{d_{x_l}} \tag{5}$$

Similar to equation 4, we have:

$$E[\Psi_n] = E[\frac{d_{x_k}}{d_{x_l}}] \tag{6}$$

Since Ψ_n converges to $E[\frac{d_{x_k}}{d_{x_l}}]$ almost surely according to SLLN, we can further conclude that Ψ_n converges to its mean $E[\Psi_n]$ with equation 6.
Now that

$$E[\Phi_n] = E[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}] = \sum_{j=1}^{n}(\frac{d_j}{D})^2 \tag{7}$$

and

$$\begin{aligned} E[\Psi_n] &= E[\frac{d_{x_k}}{d_{x_l}}] \\ &= \sum_{x_k=1}^{n}\sum_{x_l=1}^{n}\frac{d_{x_k}}{d_{x_l}}\frac{d_{x_k}}{D}\frac{d_{x_l}}{D} \\ &= n\sum_{j=1}^{n}\frac{(d_j)^2}{D^2} \end{aligned} \tag{8}$$

we have n:

$$n = \frac{E[\Psi_n]}{E[\Phi_n]} \tag{9}$$

As we have discussed, Ψ_n and Φ_n converge to their expectations almost surely, therefore $\frac{\Psi_n}{\Phi_n}$ converges to $\frac{E[\Psi_n]}{E[\Phi_n]}$ for sure. Based on this, the estimator for n is therefore defined as follows:

$$\hat{n} \triangleq \frac{\Psi_n}{\Phi_n}$$

.

2) Neighbor Collision Estimator VS Node Collision Estimator: The reason for applying neighbor collision estimator $\hat{n}$ rather than node collision estimator is that it has less variance according to Law of Total Variance (LTV):

Let C be the number of node collisions[25], [26], then C is a binomial RV, which is the sum of an indicator/Bernoulli RV. Specifically,

$$C = \frac{1}{|I|} \sum_{(k,l)\in I} \mathbf{1}_{x_k = x_l} \tag{10}$$

where $\mathbf{1}_{x_k=x_l}$ equals to 1 if $x_k = x_l$ and 0 otherwise. Based on this, we have:

$$\begin{aligned} E[\mathbf{1}_{x_{k+1}=x_{l+1}}|x_k, x_l] &= \mathbf{P}(\mathbf{1}_{x_{k+1}=x_{l+1}} = 1|x_k, x_l) \\ &= \sum_{i\in A_{x_k}\cap A_{x_l}} \mathbf{P}(X_i|x_k, x_l)\mathbf{P}(Y|X_i, x_k, x_l) \\ &= \sum_{i\in A_{x_k}\cap A_{x_l}} \frac{1}{d_{x_k}} \frac{1}{d_{x_l}} \\ &= |A_{x_k} \cap A_{x_l}| \frac{1}{d_{x_k} d_{x_l}} \\ &= \frac{\phi_{k,l}}{d_{x_k} d_{x_l}} \end{aligned} \tag{11}$$

where the second equality holds due to Total Probability Theorem (TPT) and event $X_i \triangleq \{$ a common node $i \in A_{x_k} \cap A_{x_l}$ is visited at step $k+1\}$ while event $Y \triangleq \{$ the same node i is visited at step $l+1\}$. X_i and Y are independent because m is large. The third equality holds because each neighbor is visited with equal probability.
Combining equations 3 and 11 we have:

$$\begin{aligned} \Phi_n &= \frac{1}{|I|} \sum_{(k,l)\in I} \phi_{k,l} \frac{1}{d_{x_k} d_{x_l}} \\ &= \frac{1}{|I|} \sum_{(k,l)\in I} E[\mathbf{1}_{x_{k+1}=x_{l+1}}|x_k, x_l] \end{aligned} \tag{12}$$

Next, comparing equations 10 and 12 we observe that Φ_n (denoted by $\hat{\Theta}$ for the following expression convenience) is an unbiased estimator[27], [28] of C (denoted by Θ for the following expression convenience) since $E[\tilde{\Theta}] = E[\hat{\Theta} - \Theta] = E[\hat{\Theta}] - E[\Theta] = E[E[\Theta|X]] - E[\Theta] = E[\Theta] - E[\Theta] = 0$ where Θ represents $\mathbf{1}_{x_k=x_l}$ and X is x_k, x_l in our case. The fourth equality holds because of Law of Iterative Expectation (LIE) ($E[E[\Theta|X]] = E[\Theta]$). Therefore, var($C$) $\geq$ var(Φ_n) according to the Law of Total Variance (LTV): var(Θ)=var(E[$\Theta|X$])+E[var($\Theta|X$)]=var($\hat{\Theta}$)+E[var($\Theta|X$)], which explains why the neighbor collision estimator has a smaller variance than the node collision one.

C. Clustering Coefficient Estimation

The idea of designing clustering coefficient estimators is similar to that of network size estimator. That is, combine two related estimators to find the targeted one.

Let us first consider the design of Φ for both network average and global clustering coefficient estimation which can be summarized as follows:

$$E[\phi_k f(x_k)] = \sum_{i=1}^{n} \mathbf{P}(x_k = i) E[\phi_k f(x_k)|x_k = i] \tag{13}$$

where $x_k = i$ means node i is sampled at step k. Notice that

$$\begin{aligned} E[\phi_k f(x_k)|x_k = i] &= E[A_{x_{k-1},x_{k+1}} f(v_i)|x_k = i] \\ &= f(v_i) E[A_{x_{k-1},x_{k+1}}|x_k = i] \\ &= f(v_i) \mathbf{P}(A_{x_{k-1},x_{k+1}} = 1|x_k = i) \\ &= f(v_i) \frac{2l_i}{d_i d_i} \end{aligned} \tag{14}$$

where the second equality holds due to the property of conditional expectation and the third one holds because $A_{x_{k-1},x_{k+1}}$ is a Bernoulli RV). And $\mathbf{P}(A_{x_{k-1},x_{k+1}} = 1|x_k = i) = \frac{2l_i}{d_i d_i}$ which is slightly different from the definition of local clustering coefficient $c_i = \frac{2l_i}{d_i(d_i-1)}$ because it is possible that the vertices sampled by random walk at step $k-1$ and $k+1$ are the same. And the number 2 on the numerator is because the order of the vertices visited at step $k-1$ and $k+1$ can be swapped. Insert equation 14 to equation 13 we have:

$$\begin{aligned} E[\phi_k f(x_k)] &= \sum_{i=1}^{n} \frac{d_i}{D} f(v_i) \frac{2l_i}{d_i d_i} \\ &= \sum_{i=1}^{n} \frac{1}{D} \frac{2l_i}{d_i} f(v_i) \end{aligned} \tag{15}$$

Based on equation 15, the two clustering coefficients are designed as follows:

Network Average Clustering Coefficient :
Let

$$\Phi_l = \frac{1}{r-2} \sum_{k=2}^{r-1} \phi_k \frac{1}{d_{x_k} - 1}$$

$$\Psi_l = \frac{1}{r} \sum_{k=1}^{r} \frac{1}{d_{x_k}}$$

Similar to the equation 4 and its following conclusion, we have:

$$\lim_{r\to\infty} \frac{\Phi_l}{\Psi_l} = \frac{E[\Phi_l]}{E[\Psi_l]} \tag{16}$$

Notice that

$$\begin{aligned} E[\Phi_l] &= E[\phi_k \frac{1}{d_{x_k} - 1}] \\ &= \sum_{i=1}^{n} \frac{1}{D} \frac{2l_i}{d_i} \frac{1}{d_i - 1} \\ &= \frac{1}{D} \sum_{i=1}^{n} c_i \end{aligned} \tag{17}$$

where the second equality holds because of equation 15 and the last equality comes from the definition of local clustering

coefficient $c_i = \frac{l_i}{C_{d_i}^2}$.
Similarly we have:

$$\begin{aligned} E[\Psi_l] &= E[\frac{1}{d_{x_k}}] \\ &= \sum_{i=1}^{n} \frac{1}{d_i} \frac{d_i}{D} \\ &= \frac{n}{D} \end{aligned} \tag{18}$$

Combining equations 17 and 18 we have:

$$c_l = \frac{1}{n} \sum_{i=1}^{n} c_i = \frac{E[\Phi_l]}{E[\Psi_l]} = \lim_{r \to \infty} \frac{\Phi_l}{\Psi_l} \tag{19}$$

where the last equality holds because of equation 16. Based on equation 19, the estimator for network clustering coefficient c_l is designed as follows:

$$\hat{c_l} \triangleq \frac{\Phi_l}{\Psi_l}$$

Global Clustering Coefficient :
Let

$$\Phi_g = \frac{1}{r-2} \sum_{k=2}^{r-1} \phi_k d_{x_k}$$

$$\Psi_g = \frac{1}{r} \sum_{k=1}^{r} d_{x_k} - 1$$

From the SLLN we have:

$$\lim_{r \to \infty} \frac{\Phi_g}{\Psi_g} = \frac{E[\Phi_g]}{E[\Psi_g]} \tag{20}$$

Notice that

$$\begin{aligned} E[\Phi_g] &= E[\phi_k d_{x_k}] \\ &= \sum_{i=1}^{n} \frac{1}{D} \frac{2l_i}{d_i} d_i \\ &= \frac{1}{D} \sum_{i=1}^{n} 2l_i \end{aligned} \tag{21}$$

where the second equality holds because of equation 15. Similarly we have:

$$\begin{aligned} E[\Psi_g] &= E[d_{x_k} - 1] \\ &= \sum_{i=1}^{n} (d_i - 1) \frac{d_i}{D} \\ &= \frac{1}{D} \sum_{i=1}^{n} d_i(d_i - 1) \end{aligned} \tag{22}$$

Combining equations 21 and 22 we have:

$$c_g = \frac{\sum\limits_{i=1}^{n} l_i}{\sum\limits_{i=1}^{n} \frac{d_i(d_i-1)}{2}} = \frac{\sum\limits_{i=1}^{n} 2l_i}{\sum\limits_{i=1}^{n} d_i(d_i-1)} = \frac{E[\Phi_g]}{E[\Psi_g]} = \lim_{r \to \infty} \frac{\Phi_g}{\Psi_g} \tag{23}$$

where the last equality holds because of equation 19.

Based on equation 23, the estimator for global clustering coefficient c_g is designed as follows:

$$\hat{c_g} \triangleq \frac{\Phi_g}{\Psi_g}$$

D. Summary of Simulation Results

The paper uses only the network's largest connected component and removes the direction of the original network since the algorithms in the paper are designed for undirected graph.

The simulation compares the algorithm in the paper with Frontier Samping (FS) [2] and Metropolis-Hastings Sampling (MHS)[11]. FS and MHS are named as Ego network algorithms in the paper. The mined nodes by Ego network algorithms for estimating Network Average Clustering Coefficient (NACC) and Global Clustering Coefficient (GCC) include nodes visited by random walk and the neighbors of the visited nodes. So the number of mined nodes for estimating NACC and GCC of Ego network algorithms is larger than the random walk steps. On the contrary, the paper provides the way that the number of mined nodes is exactly the same as random walk steps when evaluating the performance of all network size estimators.

The result shows that the proposed random walk estimators outperforms Ego network estimators for estimating network average clustering coefficient, global clustering coefficient and network size. And the confidence intervals of the random walk estimators are tighter than that of Ego network estimators for all the tested networks include DBLP, Orkut, Flickr and LiveJ. To compare the performance of the estimators for GCC, the paper suggests a weighted sum of local clustering coefficients as the GCC for Ego network estimators since there is no prior algorithm for estimating GCC.

IV. Estimating and Sampling Graphs with Multidimensional Random Walks

The paper aims to estimate graph characteristics with Frontier Sampling (FS) which samples edges uniformly. The main difference between FS and the regular random walk sampling methods is that FS applies multiple random walkers and mitigates the problem that a single random walker is trapped by disconnected components which lead to large estimation errors when the characteristics of the network sampled by a single random walker are different from that of the overall graph. FS also has smaller Mean Squared Errors (MSE) than that of multiple independent random walkers. Furthermore, FS can sample the tail degree distribution with performance comparable to random edge sampling and better than random vertex sampling.

A. Frontier Sampling

The paper assumes that both the incoming and outgoing edges can be acquired by querying a vertex. The transition probability for the walker moving from a vertex u to one of its neighboring vertex v can be expressed as: $\forall(v_i, v_{i+1}) \in E$, **P**(the walker visits $v_{i+1} \in$ neighbors(v_i) at step $i+1$ — the walker locates at v_i at step i)=$\frac{1}{deg(v_i)}$, where $deg(v_i)$ is the outdegree of vertex v_i.

Figure 2 shows an example for the FS procedure. FS repeats the following three steps after initializing a state on the Cartesian graph G^m by randomly selecting m vertices on the original graph G:
Step 1: select a node v with the probability proportional to its

outdegree deg(v);
Step 2: uniformly select an outgoing edge associated with the selected node v.
Step 3: replace the starting node v by the end node of the outgoing edge.
The sampled edges with the labels of index are grouped in a set and the indices indicate the sampling order. Notice that FS can differentiate the order of sampled edges, however, this order information is not necessary for estimating the graph charateristics, which is important for the realization of distributed FS, as we will discuss later.

The definitions and notations for FS on graph G are summarized in Table I.

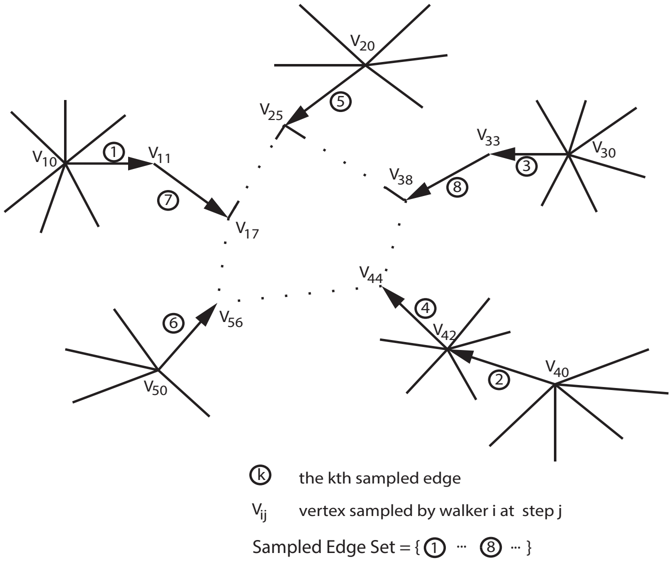
V20
V25
5
V10
1
V11
7
V17
V33
3
V30
V38
8
V44
4
V42
2
V40
6
V56
V50
k the kth sampled edge
Vij vertex sampled by walker i at step j
Sampled Edge Set = { 1 ··· 8 ··· }

Fig. 2: Frontier Sampling Scenario.

Figure 2 depicts the picture that m samplers walk in turn on the original graph, the process of which can be described in a succinct way by a Markov Chain based on the Cartesian power of G: $G^m = (V^m, E_m)$, where $V^m = \{(v_1, ..., v_m) | v_i \in V$ for all $i = 1, ..., m\}$ is the m-th Cartesian power of V. The two nodes (which are in the form of vectors) on the new formed graph G^m are connected only if the two vectors are different from each other by only one element, i.e., $\forall\ \mathbf{v}, \mathbf{u} \in V^m$, $(\mathbf{v}, \mathbf{u}) \in E_m$ if there exists an index i such that $(v_i,\ u_i) \in E$ and $u_j = v_j$ for $j \neq i$.

In the following paragraphs, we give detailed explanations on those critical conclusions in the paper.

Lemma 5.1 in the paper gives an important statement that the Frontier sampling process is equivalent to the sampling process of a single random walker over G^m. We present the details of how to get the transition probability from state L_n to state $L_(n+1)$, where $L_n = (v_1, \cdots, v_m)$ is the state of FS and also the node on graph G^m at the n-th step. And $e(L_n)$ denotes the set of all outgoing edges associated with the vertices of L_n and is the edge frontier at the n-th step. The FS sampling is equivalent to randomly sampling an edge from $e(L_n)$ with probability $\frac{1}{|e(L_n)|}$. To understand this, consider $\mathbf{P}$(the Edge related to vertex u is Selected) = $\sum_{v \in L_n}$ $\mathbf{P}$(a Vertex v is Selected) $\mathbf{P}$(the Edge related to vertex u is Selected | the Vertex v is Selected) which can be expressed in a compact way:

$$\begin{aligned}\mathbf{P}(ES) &= \sum_{v \in L_n} \mathbf{P}(VS)\mathbf{P}(ES|VS) \\ &= \frac{deg(v_1)}{\sum\limits_{v \in L_n} deg(v)} \cdot 0 + \cdots + \frac{deg(u)}{\sum\limits_{v \in L_n} deg(v)} \cdot \frac{1}{deg(u)} + \\ &\quad \cdots + \frac{deg(v_m)}{\sum\limits_{v \in L_n} deg(v)} \cdot 0 \\ &= \frac{1}{\sum\limits_{v \in L_n} deg(v)} = \frac{1}{|e(L_n)|} \end{aligned} \tag{24}$$

where vertices $v_1, \cdots, v_m$ are elements of state L_n and vertex u is the one being sampled at step n.

Theorem 5.2 in the paper gives the steady state distribution of $L_\infty = (v_1, \cdots, v_m)$. Here we give the explanation that why $\mathbf{P}(L_\infty) = \frac{\sum_{i=1}^m deg(v_i)}{m|V|^{(m-1)} vol(V)}$. Each state (i.e. each node in G^m) $L = (v_1, \cdots, v_i, \cdots, v_m)$ which includes v_i as its element has $deg(v_i)$ edges connected to other states (nodes) in G^m because of v_i, since by fixing the walkers on other vertices than the one on v_i, the walker on v_i has $deg(v_i)$ neighbors as the options for the next step. According to the definition on how the states $L = (v_1, \cdots, v_i, \cdots, v_m)$ are connected on G^m, there are exactly $deg(v_i)$ edges concerning v_i (totally more than $deg(v_i)$ edges) attached to the nodes $L \in G^m$ which has vertex v_i as its element. Since the total number of nodes on G^m containing v_i is $m|V|^{m-1}$, the total number of edges attached to these nodes concerning v_i is $m|V|^{m-1} deg(v_i)$. Therefore, the total number of edges(outdegrees) on G^m is $m|V|^{m-1}(deg(v_1) + \cdots + deg(V_{|V|})) = m|V|^{m-1} vol(V) = m|V|^{m-1}|E_m|$, where V is the vertex set for the original graph G and E_m is the edge set for directed graph G^m. Notice that for undirected graph, $vol(V) = \sum_{i=1}^{|V|} deg(v_i) = 2|E|$. Figure 3 gives the relationship between graph G and G^m.

Lemma 5.3 gives the probability of the event that there are exactly k random walkers in V_A at steady state. Notice that random variable (RV) $K_{fs}(m)$ denotes the number of random walkers in V_A at steady steady state. Further, we define set $\{L_k\}$ as the set of states $L \in V^m$ satisfying k walkers(vertices) $\in V_A$ at steady state. Based on the above definition, we have the following derivations:

$$\begin{aligned}\mathbf{P}[K_{fs}(m) = k] &= \sum_{L \in \{L_k\}} \mathbf{P}[L = (v_1, \cdots, v_m)] \\ &= \sum_{L \in \{L_k\}} \frac{\sum_{i=1}^m deg(v_i)}{m|V|^{m-1} vol(V)} \ , where(v_i \in L) \\ &= \frac{\sum_{L \in \{L_k\}} \sum_{i=1}^m deg(v_i)}{m|V|^{m-1} vol(V)} \end{aligned} \tag{25}$$

Next we use an example to illustrate how to further expand the numerator of equation 25: Suppose the number of walkers $m = 4$ and the number of walkers $\in V_A$ is $k = 2$. Then there are $\binom{m}{k} = \binom{4}{2} = 6$ combinations of random walkers satisfying the condition that exactly k walkers $\in V_A$ as

Notation	G_d	V	E_d	L_v	L_e
Explanation	directed graph	vertex set	edge set of a directed graph	set of vertex labels	set of edge labels
Example	$G_d = (V, E_d)$	Each vertex has at least one incoming or outgoing edge.	$\{\cdots, (u,v), (v,u), \cdots\}$	$L_v(V) \subset L_v$, $\forall v \in V$, $e.g. L_v(V)$ =indegree(V)	$L_e(u,v) =$ $\{\cdots, k, l, m, \cdots\}$
Notation	G	V	E		
Explanation	symmetric directed graph		edge set of an undirected graph		
Example	$G = (V, E)$, deg(V) = in-degree(V) = out-degree(V)	vol(V) $= \sum\limits_{\forall v \in V} deg(v)$	E $= \cup\{(u,v),(v,u)\}$, $\forall(u,v) \in E_d$		
Notation	G^m	V^m	E_m		
Explanation	m-th Cartesian power of G	Set of nodes of G^m	Set of edges of G^m		
Example	$G^m = (V^m, E_m)$	$L \in V^m, L = (v_1, ..., v_m)$			

TABLE I: Notations for Frontier Sampling

shown in Table II, where the checkmark indicates that the corresponding walker is in set V_A and the crossmark indicates that the corresponding walker is in set V_B. By listing all vertex combinations for each row in Table II, we observe that the number of all possible appearances of each vertex $\in V_A$ is the same, and equals to $\frac{|V_A|^k |V_B|^{m-k} k}{|V_A|}$. Similarly, the number of all possible apearances of each vertex $\in V_B$ is also the same, and equals to $\frac{|V_A|^k |V_B|^{m-k}(m-k)}{|V_B|}$. Therefore, for each row in Table II, the number of out-degrees of all vertices $\in V_A$ is $\frac{|V_A|^k |V_B|^{m-k} k}{|V_A|} \sum_{v_i \in V_A} outdeg(v_i)$, and the number of out-degrees of all vertices $\in V_B$ is $\frac{|V_A|^k |V_B|^{m-k}(m-k)}{|V_B|} \sum_{v_j \in V_B} outdeg(v_j)$.

Now we are ready to express the numerator of equation 25 as follows:

$$\begin{aligned}\sum_{L \in \{L_k\}} \sum_{i=1}^{m} deg(v_i) &= \binom{m}{k} \Big(\frac{|V_A|^k |V_B|^{m-k} k}{|V_A|} \sum_{v_i \in V_A} outdeg(v_i) \\ &\quad + \frac{|V_A|^k |V_B|^{m-k}(m-k)}{|V_B|} \sum_{v_j \in V_B} outdeg(v_j) \Big) \\ &= \binom{m}{k} |V_A|^k |V_B|^{m-k} \Big(\frac{k}{|V_A|} \sum_{vi \in V_A} deg(v_i) \\ &\quad + \frac{m-k}{|V_B|} \sum_{v_j \in V_B} deg(v_j) \Big)\end{aligned} \tag{26}$$

Combining equations 25 and 26 we have: $\mathbf{P}[K_{fs}(m) = k] = \binom{m}{k} \frac{|V_A|^k |V_B|^{m-k}}{m|V|^{m-1} vol(V)} \Big(\frac{k}{|V_A|} \sum_{vi \in V_A} deg(v_i) + \frac{m-k}{|V_B|} \sum_{v_j \in V_B} deg(v_j) \Big)$, which is the same as equation (10) in the paper.

Next we explain equation (12) in the paper:
We repeat the following three equations in the paper:

$$\frac{mpd_A + (m - mp)d_B}{md} = 1 \tag{27}$$

$$\lim_{m \to \infty} \frac{k^-(m)d_A + (m - k^-(m))d_B}{md} = 1 \tag{28}$$

And

$$\lim_{m \to \infty} \frac{k^+(m)d_A + (m - k^+(m))d_B}{md} = 1 \tag{29}$$

Equations 27, 28 and 29 tell us how the value of k affects $\frac{kd_A + (m-k)d_B}{md}$: When k equals to the mean of RV $K_{un}(m)$, i.e., $k = mp$, $\frac{kd_A + (m-k)d_B}{md} = 1$, as shown by equation 27. Equations 28 and 29 discuss when k is around the mean of RV $K_{un}(m)$, i.e., $k = k^{\pm} = mp \pm z(m)\sqrt{mp(1-p)}$, what the limit of $\frac{kd_A + (m-k)d_B}{md}$ is. And the conclusion is that both the limits are equal to 1. Now we are ready to show why the limits of $\mathbf{P}[K_{fs}(m) = k]$ and $\mathbf{P}[K_{un}(m) = k]$ are equal. Since Lemma 5.3 gives that

$$\mathbf{P}[K_{fs}(m) = k] = \frac{kd_A + (m-k)d_B}{md} \mathbf{P}[K_{un}(m) = k],$$

Apply equations 27, 28, 29, together with

$$\lim_{m \to \infty} \mathbf{P}[K_{un}(m) < k^-(m)] = 0$$

$$\lim_{m \to \infty} \mathbf{P}[K_{un}(m) > k^+(m)] = 0$$

and

$$\lim_{m \to \infty} \frac{k(m)d_A + (m - k(m))d_B}{md} < \infty, (k(m) = o(m)),$$

we can conclude that:

$$\lim_{m \to \infty} \mathbf{P}[K_{fs}(m) = k] = \lim_{m \to \infty} \mathbf{P}[K_{un}(m) = k], \forall k \geq 0$$

B. Distributed FS

When the walkers of FS sample the graph G independently, FS becomes distributed FS and a continuous random process rather than a discrete Markov Chain is required to describe the random process. For FS, the time interval between two neighboring edges in the sampled sequence does not matter since the walkers sample the graph G in a coordinated and sequential manner. However, the inter-arrival time for distributed FS matters since its walkers do not communicate with each other and the probability of transition from state

Walker 1	Walker 2	Walker 3	Walker 4
✓	✓	✗	✗
✓	✗	✓	✗
✓	✗	✗	✓
✗	✓	✓	✗
✗	✓	✗	✓
✗	✗	✓	✓

TABLE II: Combinations of FS random walkers when $k = 2$ walkers in V_A and $m = 4$ walkers in total

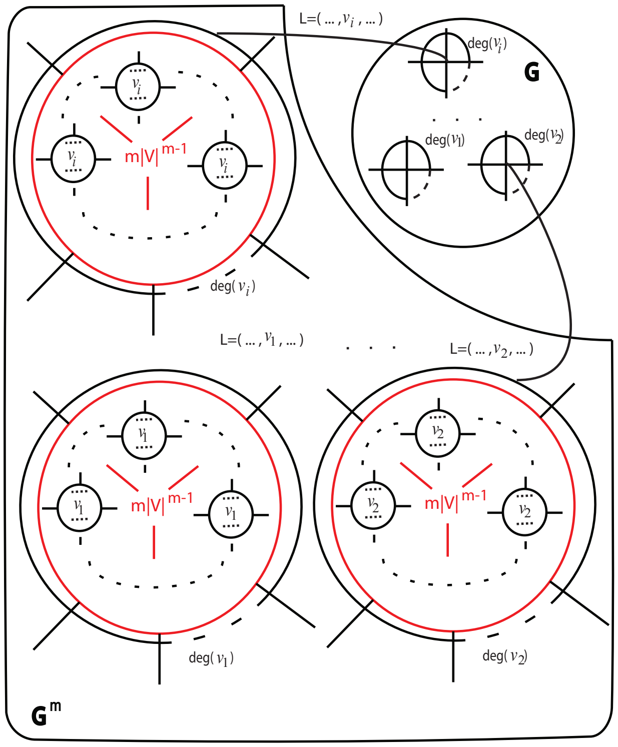

Fig. 3: Relationship between G and G^m.

L_n to L_{n+1} depending on the elapse of time: consider the Cartesian power G^m, since some walker j may sample a vertex after a walker i with a delay by time t, the transition from L_n to L_{n+1} is actually a continuous time process. Such a transition process can be modeled by a Poisson Process since the sampling procedure of the distributed FS is naturally a memoryless process (Recall that the interarrival times of poisson process are i.i.d. exponential random variables, the distribution of which has the memoryless property. Also notice that the Poisson distribution and exponential distribution share the same parameter – the arrival rate λ.)

Now the question comes down to how to connect the original discrete Markov Chain for FS with the continuous Poisson process for the distributed FS. Uniformization of Markov Chains [29] is a powerful tool to convert a continuous-time Markov model into a discrete-time Markov model without distorting the information that the former contains. So instead of analyzing the distributed FS process with a continuous Poisson process, we can work on a less complicated discrete Markov Chain model.

Specifically, the Uniformization of Markov Chain creates a fictitious event with a self-loop transition which has no effect on the behavior of the original chain: The state itself has not been affected, and, by the memoryless property, the time left until the next actual transition out of some state i also remains unaffected, that is, it is still exponentially distributed with parameter λ_i. As a result, the new uniformized chain should be stochastically indistinguishable from the original one. That is, given a common initial state x_0, the two models should have identical state probabilities $\pi_i(t)$ for all t and states i.

If we model the distributed FS as a Poisson process, then it can be decomposed into small and individual Poisson process due to the Poisson Decomposition property. Here we briefly introduce Poisson Process Decomposition[30]: let $N = \{N_t; t \geq 0\}$ be a Poisson process with rate λ and let $\{X_1, X_2, \cdots\}$ denote an *i.i.d.* sequence of Bernoulli random variables independent of the Poisson process such that $\mathbf{P}(X_n = 1) = p$. Then $M = \{M_t; t \geq 0\}$ is a new process formed as follows: for each positive n, consider the n^{th} arrival to the N process, it is also an arrival to the M process if X_n=1; otherwise, the n^{th} arrival to the N process is not part of the M process. The resulting M process is a Poisson process with rate λp.

Based on the above introduction, the Poisson process decomposition in the paper is formed as follows: Let the rate of the composite Poisson process be $\Lambda = \sum_{\forall v \in L} deg(v)$. According to Algorithm 1 in the paper, a vertex u is sampled with probability p=deg(u)/$\sum_{\forall v \in L}$ deg(v). Therefore, the sampling process for each walker is actually a Poisson process with the rate $\lambda = \Lambda p = deg(u)$ which depends only on the outdegree of the vertex u and not on the state L. This process is equivalent to a Multiple RW process with m independent random walkers and sampling budget B (total amount of time), and the cost of sampling a vertex u (inter-arrival time) is an exponentially distributed RV with rate λ=deg(u).

Since each random walker of distributed FS works individually, ordinary and less expensive computers can be applied to do the sampling instead of a super powerful and highly expensive computer. The edges sampled by these small PCs actually have the same distribution as those sampled by a single super computer since the order of the sampled edges does not affect the estimators.

C. Summary of Simulation Results

The datasets evaluated include Flickr, Livejournal, YouTube and Internet RLT. FS is compared with SingleRW, MultipleRW, random vertex and edge sampling. The simulation results show that FS gives more accurate estimations and reaches the steady state faster than SingleRW and MultipleRW. Specifically, FS has smaller bias and NMSE than both

MultipleRW and SingleRW for estimating AMC and degree distribution (DD) of connected graph. Plus, FS is able to accurately estimate AMC and DD for loosely connected graph while both SingleRW and MultipleRW give a wrong estimation. FS accurately estimates the global clustering coefficient and has smaller error than both SingleRW and MultipleRW. The simulation result also shows that FS and random edge sampling are more accurate than random vertex sampling at estimating tail of degree distribution.

V. Efficiently Estimating Motif Statistics of Large Networks

A. Brief Introduction to Algorithms of Sampling Motifs

This paper aims to accurately estimate properties of small connected subgraph patterns (network motifs) with as few queries as possible and without the complete graph topology.

Prior to the algorithms proposed in the paper, Kashtan et al. uses random edge sampling to sample motifs. Wernicke designed the algorithm FANMOD to enumerate subgraph trees through random node sampling. Both algorithms need to know the complete graph topology. Bhuiyan et al. used a Metropolis-Hastings-based algorithm with the name GUISE to jointly estimate the frequency of three-, four- and five- node CIS. However, GUISE is not perfect for estimating the targeted motif concentration since the Metropolis-Hastings algorithm rejects samples and therefore wastes the sampling budget without gathering the targeted motifs, which leads to large estimation errors as pointed out by Ribeiro and Towsley [2].

Based on the above observation, Wang et al. aim to collect all the targeted samples without rejection and to acquire an accurate estimation of the targeted motif concentration by creating a CIS relationship graph on which the size of the motifs is the same as the targeted one. The methods proposed in the paper represent the network as a CIS relationship graph whose nodes are connected and induced subgraphs (CIS) or motifs of the original network. The algorithms run a random walk on the CIS relationship graph but do not require the complete CIS relationship graph. And the estimation of the motif concentration is only based on the queried nodes.

The algorithms presented in the paper include Subgraph Random Walk (SRW), Pairwise Subgraph Random Walk (PSRW) and Mixed Subgraph Sampling (MSS). SRW is a regular RW over the undirected graph $G^{(k)}$ to estimate the motifs of size k. The probabilities of sampling edges by SRW are equal when the SRW reaches the steady state. PSRW is an improved version of SRW with a more accurate estimation. To estimate the concentration of motifs of size k, PSRW walks on the CIS graph with motifs of size $k-1$ instead of directly sampling the motifs of size k by walking on the CIS graph whose motifs are of size k. Since PSRW does not reject samples, its estimation errors are significantly lower than those of GUISE. MSS is a generalized version of GUISE which can jointly estimate the concentration of motifs with sizes $k-1$, k, and $k+1$ for any $k \geq 4$ compared to GUISE which can only estimate the frequency of CIS with sizes three, four, and five. More precisely, the tailored version of GUISE which is called Metropolis Hasting Subgraph Random Walk (MHSRW) by the authors is a special case of MSS for when $k = 4$. MSS also achieves lower estimation errors of motif concentration than that of GUISE.

B. Random Walk on CIS Graph

We use an example of sampling the motifs of size three to illustrate how the algorithm SRW works. The notations for the example are summarized in Table III. The most important part is to find $X(S)$, which is the set of the neighboring motifs of the current motif S.

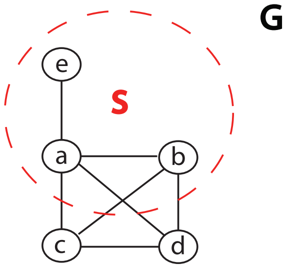

Fig. 4: Original Graph G

First, identify the elementary nodes of a motif and initialize it as the starting node on the CIS graph, e.g. $C_1^{(3)} = (e, a, b)$; Second, query one of neighboring nodes (node c or d) on graph G of the nodes consisting the current motif. Notice that we need identify all the neighboring motifs of S to be able to walk one step ahead on the CIS graph. And to acquire these motifs, we need both their vertices and the edges.

Since by querying each node of the initial motif S, we know what nodes on the original graph G are the neighbors of the nodes consisting of motif S. In addition, the edge connections for nodes inside and outsie of S are also available. Therefore, both $E(S)$ and $E^{(N)}(S)$ can be determined.

Next, we list all the possible vertex combinations for $X(S)$: $(e, a, c), (e, a, d), (e, b, c), (e, b, d), (a, b, c)$ and (a, b, d). And then count the number of existing edges for each of the above vertex combination. For those vertex combination with the number of edges $n(E) < N-1$, where $N = 3$ is the motif size for our example, they cannot form a motif since at least $N-1$ edges are needed to connect the vertices and form a motif with size $N = 3$. Based on this, we can determine both the vertex set and the edge set of $X(S)$. After we have $X(S)$, the walker on the CIS graph can walk one step forward. The above querying and walking process will repeat until the sampling budget is reached.

A key observation is that the motif graph is not known in advance. The motifs and their inter-connections cannot be determined until the corresponding node on the original graph is accessed. However, this does not affect walking on the motif graph since the neighbors of the current motif is accessible. We can imagine the future nodes of the motif graph as the bulbs which will be illumined when the corresponding node on the original graph is queried. Through the above analysis, we can see there is an interaction between the motif graph and the original graph, that is, walking on the original graph depends on walking on the CIS graph: which motif on the CIS

Notation	Explanation	Example
S	Original motif	$C_1^{(3)}$
S'	Neighboring motifs of S on the CIS graph	$C_2^{(3)}$, $C_3^{(3)}$, $C_4^{(3)}$, $C_7^{(3)}$
$V(S)$	Vertex set of S	$\{e, a, b\}$
$E(S)$	Edge set of S	$\{\overline{ea}, \overline{ab}\}$
$N(S)$	Vertex set consists of nodes that are neighbors of vertices of S	$\{c, d\}$
$E^{(N)}S$	Edge set consists of $\overline{v_1 v_2}$, where$v_1 \in V(S)$ and $v_2 \in N(S)$	$\{\overline{ac}, \overline{ad}, \overline{bc}, \overline{bd}\}$
$V(S')$	Vertex set of S'	$\{(e, a, c), (e, a, d), (a, b, c), (a, b, d)\}$
$E(S')$	Edge set of motif S'	$\{(\overline{ea}, \overline{ac}), (\overline{ea}, \overline{ad}), (\overline{ab}, \overline{ac}, \overline{bc}), (\overline{ab}, \overline{bd}, \overline{ad}\}$

TABLE III: Notations for Motif Sampling SRW

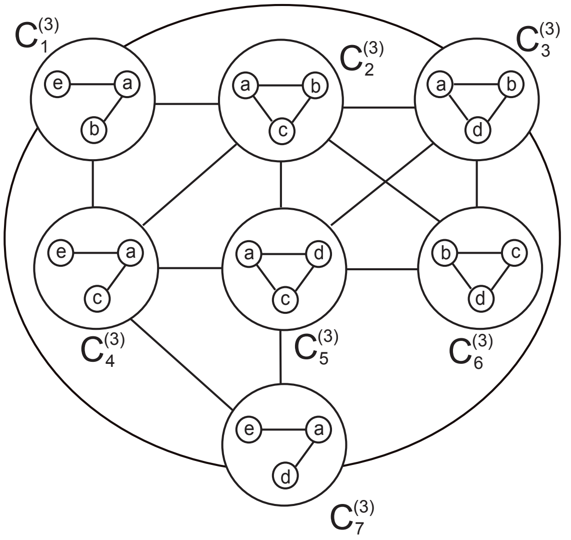

Fig. 5: CIS Relationship Graph of Motifs with Size Three

graph to be selected as the next stop determines which node on the original graph will be queried.

We summarize the procedures of sampling motifs as follows:
(1) Specify the targeted motif class $C_i^{(k)}$, i.e. the type of node connection(e.g. two edges for three nodes or three edges for three nodes).
(2) Initialize a node in the CIS relationship graph as the starting motif.
(3) Query the nodes on graph G which compose the current motif and determine the neighboring motifs on the CIS relationship graph.
(4) Select one of the neighboring motif as the next stop, if it is the targeted motif, add the counter by one for estimating its concentration.
(5) Repeat (3) and (4) until reaching the sampling budget.

C. Estimators for Motif Concentration

The concentration of a motif $C_i^{(k)}$ of type i and with size k is

$$\omega_i^{(k)} = \frac{|C_i^{(k)}|}{|C^{(k)}|} \tag{30}$$

where $1 \leq i \leq T_k$. Note that the NAUTY algorithm [McKay 1981, 2009] can help determine the subgraph class of a CIS S which is denoted as $C(S)$.

1) SRW: The asymptotically unbiased estimator of the concentration of subgraph class $C_i^{(k)}$ is repeated as follows:

$$\hat{\omega}_i^{(k)} = \frac{1}{L} \sum_{j=1}^{B} \frac{1(C(s_j) = C_i^{(k)})}{d^{(k)}(s_j)} \tag{31}$$

where $L = \sum_{j=1}^{B} \frac{1}{d^{(k)}(s_j)}$.
And the stationary distribution of the random walk is:

$$\boldsymbol{\pi}^{(k)} = (\pi^{(k)}(s) : s \in C^{(k)}) \tag{32}$$

where $\pi^{(k)}(s) = \frac{d^{(k)}(s)}{\sum_{t \in C^{(k)}} d^{(k)}(t)}$

2) PSRW: PSRW is an improved version of SRW with a more accurate estimation. To estimate the concentration of motifs of size k, PSRW walks on the k-1 node CIS graph instead of directly sampling the motifs of size k by walking on the k-node CIS graph. Note that two adjacent sampled (k-1)-node CIS (i.e., an edge in $G^{(k-1)}$) contains exactly k distinct nodes. Since SRW samples each edge in $G^{(k-1)}$ with equal probability at steady state. The k-node CIS x is sampled with the following probability:

$$\pi_E^{(k)}(x) = \frac{I^{(k-1)}(x)(I^{(k-1)}(x) - 1)}{\sum_{y \in C^{(k)}} I^{(k-1)}(y)(I^{(k-1)}(y) - 1)} \tag{33}$$

where $I^{(k-1)}(x)$ denotes the number of (k-1) node motifs contained by k node motif x.

PSRW gives an asymptotically unbiased estimator $\omega_i^{(k)}$ as follows,

$$\tilde{\omega}_i^{(k)} = \frac{1}{H} \sum_{j=1}^{B-1} \frac{\mathbf{1}(C(s_j^*) = C_i^{(k)})}{I^{(k-1)}(s_j^*)(I^{(k-1)}(s_j^*) - 1)} \tag{34}$$

where

$$H = \sum_{j=1}^{B-1} [I^{(k-1)}(s_j^*)(I^{(k-1)}(s_j^*) - 1)]^{-1}$$

PSRW produces more accurate estimates than SRW, but it remains an open theoretical problem why PSRW significantly outperforms SRW.

3) MSS: MSS can jointly estimate the concentrations of subgraph classes of sizes $k-1$, k, and $k+1$ ($k \geq 4$). MSS samples k-node CIS by random walking on $G^{(k)}$. It estimates the concentrations of k-node subgraph classes by SRW, estimates the concentrations of (k+1)-node subgraph classes by PSRW, and estimates the concentrations of (k-1)-node subgraph classes by the following equation:

$$\tilde{\omega}_i^{(k-1)} = \frac{1}{Q}\sum_{j=1}^{B}\frac{1}{d^{(k)}(s_j)} \cdot \sum_{s'\in C^{(k-1)}(s_j)} \frac{\mathbf{1}(C^{(k-1)}(s') = C_i^{(k-1)})}{|O^{(k)}(s')|} \tag{35}$$

where

$$Q = \sum_{j=1}^{B}\frac{1}{d^{(k)}(s_j)} \sum_{s'\in C^{(k-1)}(s_j)} \frac{1}{|O^{(k)}(s')|}$$

D. Summary of the Simulation Results

The experiments are performed on a variety of publicly available datasets taken from the Stanford Network Analysis Platform (SNAP). The simulation results show that PSRW uses less samples than other algorihtms but converges faster with more accurate estimations of motif concentration. Specifically, PSRW is significantly more accurate than the other methods for estimating the concentrations of the three-node directed CIS classes, three-node signed and undirected CIS classes. It also performs the best for estimating the concentrations of four-node, five-node, six-node undirected CIS classes.

As Table IV shows, PSRW has the smallest normalized root mean squared error of the concentration estimate $\omega_2^{(3)}$ for all the social network listed in the table. Table V compares the estimation results of $\omega_2^{(3)}$ for PSRW, MHSRW, FANMOD. The paper also gives a detailed simulation result for estimating the concentration of the three-node undirected CIS class 2 in Figure 2(a) in the paper. The estimation error of PSRW is almost an order of magnitude less than errors of MHSRW and FANMOD for Flickr and Pokec graphs. And PSRW reduces more than 10-fold the number of queries required to achieve the same estimation accuracy. SRW is more accurate than MHSRW and FANMOD but less accurate than PSRW. PSRW converges much quicker than the other methods and gives the most accurate estimation at steady state.

VI. CONTRIBUTION/ADVANTAGE /DISADVANTAGE OF THREE PAPER

A. Sampling by a Single Random Walker

The paper suggests to estimate CC and network size only by the nodes sampled by the random walker instead of querying the immediate neighbors of nodes through random walk which is used by Ego network algorithms. The algorithm in the paper achieves a better estimation accuracy than the counterpart Ego network algorithms like FS and MHRW and the confidence intervals of the random walk estimators are tighter than that of Ego network estimators. The paper also suggests to use neighbor collision rather than node collision to estimate network size, the advantage of which is supported by both the theoretical and empirical analysis.

B. Frontier Sampling

The advantages of FS over other random sampling algorithms lie in its multiple random walkers which allow to sample the graph from different locations. FS can mitigate the situation that a single random walker gets trapped in the loosely connected graph.

FS gives more accurate estimations and reaches the steady state faster than that those estimated by single random walker (SingleRW) and multiple independent random walkers (MultipleRW). In addition, FS has smaller bias and NMSE than both MultipleRW and SingleRW for estimating AMC and degree distribution (DD) of connected graph. Plus, FS is able to accurately estimate AMC and DD for loosely connected graph while both SingleRW and MultipleRW give a wrong estimation. FS accurately estimates the global clustering coefficient and has smaller error than both SingleRW and MultipleRW. FS is more accurate than random vertex sampling at estimating the tail of degree distribution.

FS can be made fully distributed by allowing the walkers to sample the graph independently. In that way, less expensive ordinary computers can be applied to do the sampling instead of a highly cost super machine. The edges sampled by these normal computers actually have the same distribution as those sampled by a single machine since the order of the sampled edges does not affect the estimators.

C. Sampling Motifs by Random Walk

The observation is that GUISE and SRW are designed for different purposes. GUISE aims to estimate the motif distribution by applying the Metropolis-Hasting sampler to sample the node (motif) uniformly at random on the composite CIS relationship graph which contains motifs with different sizes. SRW aims to estimate the concentration of a targeted motif by constructing a CIS relationship graph whose nodes (motifs) have exactly the same size as the targeted one.

The disadvantage of applying GUISE to estimate the concentration of the targeted motifs is that it rejects the targeted motifs, especially those with high degrees since the MHRW design determines that motifs with high degree have more chance to be selected but are more likely to be rejected(on the other hand, motifs with low degree have less chance to be selected but are less likely to be rejected). In addition, no matter the motif is selected or rejected, GUISE consumes one unit of sampling cost. Therefore, it wastes the budget to sample the motifs which are not k node ones, which leads to a large estimation error for estimating the targeted motif concentration with a smaller value than the true one. Therefore, the paper adapts GUISE to MHSRW which focuses on k node sampling by randomly selecting a k node CIS y from the set of neighbors of the current k node CIS x on CIS relationship graph $G^{(k)}$. The paper does not show the comparison of estimating $\omega_2^{(3)}$ between SRW and MHSRW but between PSRW and MHSRW. The result shows that PSRW gives an accurate estimation while MHSRW does not. Besides, PSRW has a smaller estimation error.

To sum up, GUISE and PSRW are designed for different purposes. GUISE is able to estimate the frequency of motifs

	PSRW	SRW	MHSRW	FANMOD
Flickr	0.15	0.21	0.32	0.35
Pokec	0.14	0.21	0.30	0.29
LiveJournal	0.20	0.27	0.41	0.42
YouTube	0.14	0.27	0.32	0.34

TABLE IV: NRMSE of Concentration Estimates of Three Node Undirected CIS Class ($\omega_2^{(3)}$)

	PSRW	MHSRW	FANMOD
Flickr ($\omega_2^{(3)}$=0.0404)	0.04	0.07	0.02
Pokec ($\omega_2^{(3)}$=0.0161)	0.016	0.025	0.018
LiveJournal($\omega_2^{(3)}$=0.0451)	0.045	0.005	0.052
YouTube($\omega_2^{(3)}$=0.0021)	0.0021	0.001	0.005

TABLE V: Concentration Estimates of Three Node Undirected CIS Class $\omega_2^{(3)}$

on the composite motif graph whose motifs are with sizes three, four and five. PSRW performs better on estimating the concentration of a targeted motif on the motif graph whose motifs are with the same size.

References

[1] S. J. Hardiman and L. Katzir, "Estimating clustering coefficients and size of social networks via random walk," in *Proceedings of the 22nd international conference on World Wide Web*, pp. 539–550, International World Wide Web Conferences Steering Committee, 2013.

[2] B. Ribeiro and D. Towsley, "Estimating and sampling graphs with multidimensional random walks," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 390–403, ACM, 2010.

[3] P. Wang, J. Lui, B. Ribeiro, D. Towsley, J. Zhao, and X. Guan, "Efficiently estimating motif statistics of large networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 9, no. 2, p. 8, 2014.

[4] M. A. Bhuiyan, M. Rahman, M. Rahman, and M. Al Hasan, "Guise: Uniform sampling of graphlets for large graph analysis," *Power*, vol. 2, no. 1, p. 0, 2012.

[5] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature genetics*, vol. 31, no. 1, pp. 64–68, 2002.

[6] I. Albert and R. Albert, "Conserved network motifs allow protein–protein interaction prediction," *Bioinformatics*, vol. 20, no. 18, pp. 3346–3352, 2004.

[7] S. Itzkovitz, R. Levitt, N. Kashtan, R. Milo, M. Itzkovitz, and U. Alon, "Coarse-graining and self-dissimilarity of complex networks," *Physical Review E*, vol. 71, no. 1, p. 016127, 2005.

[8] J. Leskovec and C. Faloutsos, "Sampling from large graphs," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 631–636, ACM, 2006.

[9] A. L. Garcia, "Probability, statistics, and random processes for electrical engineering," *Prentice Hall, Upper Saddle River, NJ*, 2008.

[10] T. Wang, Y. Chen, Z. Zhang, T. Xu, L. Jin, P. Hui, B. Deng, and X. Li, "Understanding graph sampling algorithms for social network analysis," in *Distributed Computing Systems Workshops (ICDCSW), 2011 31st International Conference on*, pp. 123–128, IEEE, 2011.

[11] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Walking in facebook: A case study of unbiased sampling of osns," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[12] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pp. 177–187, ACM, 2005.

[13] L. Lovász, "Random walks on graphs: A survey," *Combinatorics, Paul erdos is eighty*, vol. 2, no. 1, pp. 1–46, 1993.

[14] M. Gjoka, M. Kurant, C. T. Butts, and A. Markopoulou, "Practical recommendations on crawling online social networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1872–1892, 2011.

[15] M. J. Salganik and D. D. Heckathorn, "Sampling and estimation in hidden populations using respondent-driven sampling," *Sociological methodology*, vol. 34, no. 1, pp. 193–240, 2004.

[16] E. Volz and D. D. Heckathorn, "Probability based estimation theory for respondent driven sampling," *Journal of Official Statistics*, vol. 24, no. 1, p. 79, 2008.

[17] A. H. Rasti, M. Torkjazi, R. Rejaie, N. Duffield, W. Willinger, and D. Stutzbach, "Respondent-driven sampling for characterizing unstructured overlays," in *INFOCOM 2009, IEEE*, pp. 2701–2705, IEEE, 2009.

[18] M. H. Hansen and W. N. Hurwitz, "On the theory of sampling from finite populations," *The Annals of Mathematical Statistics*, vol. 14, no. 4, pp. 333–362, 1943.

[19] S. Chib and E. Greenberg, "Understanding the metropolis-hastings algorithm," *The american statistician*, vol. 49, no. 4, pp. 327–335, 1995.

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

[21] T. Wang, Y. Chen, Z. Zhang, P. Sun, B. Deng, and X. Li, "Unbiased sampling in directed social graph," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 401–402, 2011.

[22] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon, "Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs," *Bioinformatics*, vol. 20, no. 11, pp. 1746–1758, 2004.

[23] S. Wernicke, "Efficient detection of network motifs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 3, no. 4, pp. 347–359, 2006.

[24] P. Hu and W. C. Lau, "A survey and taxonomy of graph sampling," *arXiv preprint arXiv:1308.5865*, 2013.

[25] S. Hardiman, P. Richmond, and S. Hutzler, "Calculating statistics of complex networks through random walks with an application to the online social network bebo," *The European Physical Journal B*, vol. 71, no. 4, pp. 611–622, 2009.

[26] L. Katzir, E. Liberty, and O. Somekh, "Estimating sizes of social networks via biased sampling," in *Proceedings of the 20th international conference on World wide web*, pp. 597–606, ACM, 2011.

[27] D. P. Bertsekas, *Introduction to Probability: Dimitri P. Bertsekas and John N. Tsitsiklis*. Athena Scientific, 2002.

[28] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*, vol. 707. John Wiley & Sons, 2011.

[29] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[30] R. M. Feldman and C. Valdez-Flores, *Applied Probability and Stochastic Processes*. Springer Publishing Company, Incorporated, 2nd ed., 2010.