Estimating Clustering Coefficients and Size of Social Networks via Random Walk

Stephen J. Hardiman*
Capital Fund Management,
France
hardimas@tcd.ie

Liran Katzir
Advanced Technology Labs,
Microsoft Research, Israel
lirank@microsoft.com

ABSTRACT

Online social networks have become a major force in today's society and economy. The largest of today's social networks may have hundreds of millions to more than a billion users. Such networks are too large to be downloaded or stored locally, even if terms of use and privacy policies were to permit doing so. This limitation complicates even simple computational tasks. One such task is computing the clustering coefficient of a network. Another task is to compute the network size (number of registered users) or a subpopulation size. The clustering coefficient, a classic measure of network connectivity, comes in two flavors, global and network average. In this work, we provide efficient algorithms for estimating these measures which (1) assume no prior knowledge about the network; and (2) access the network using only the publicly available interface. More precisely, this work provides three new estimation algorithms (a) the first external access algorithm for estimating the global clustering coefficient; (b) an external access algorithm that improves on the accuracy of previous network average clustering coefficient estimation algorithms; and (c) an improved external access network size estimation algorithm.

The main insight offered by this work is that only a relatively small number of public interface calls are required to allow our algorithms to achieve a high accuracy estimation. Our approach is to view a social network as an undirected graph and use the public interface to retrieve a random walk. To estimate the clustering coefficient, the connectivity of each node in the random walk sequence is tested in turn. We show that the error of this estimation drops exponentially in the number of random walk steps. Another insight of this work is the fact that, although the proposed algorithms can be used to estimate the clustering coefficient of any undirected graph, they are particularly efficient on social network-like graphs. To improve the network size prior-art estimation algorithms, we count node collision one step before they actually occur. In our experiments we validate our algorithms on several publicly available social network datasets. Our results validate the theoretical claims and demonstrate the effectiveness of our algorithms.

*Research was conducted while the author was unaffiliated.

WWW2013, May. 13-17, Rio de Janeiro, Brazil
ACM 978-1-4503-2035-1/13/05.

Categories and Subject Descriptors

F.2.2 [**Theory of Computing**]: Analysis of Algorithms and Problem Complexity—*Nonnumerical Algorithms and Problems*

General Terms

Algorithms

Keywords

Estimation, Sampling, Clustering Coefficient, Social Network

1. INTRODUCTION

The popularity of online social networks has grown enormously in recent years. Users of the most popular social network, Facebook™, now number greater than a billion[1]. This popularity has increased interest in analyzing the properties of these networks. In [2, 13, 21] the authors investigate structural measures of online social networks, including degree distribution and clustering coefficient.

Large social networks, as well as search engines, provide a public interface as part of their service. Estimating structural measures of the network using only these public interfaces is a research question that has received much attention in recent studies. Search engine public interfaces have been used in [6, 8] to estimate corpus size, index freshness, and density of duplicates, and in [7] estimate the impressionrank of a webpage. Online social network public interfaces have been used in [13, 14, 25] to estimate the assortativity coefficient, degree distribution, and clustering coefficients of online social networks, as well as in [14, 15] to estimate the number of registered users.

In practical scenarios, the underlying social network may be available only through a public interface. The public interface of most social networks provides the ability to retrieve a list of a user's connections ("friends"). By applying this function iteratively to a random member of the connection list one can effectively perform a random walk on the network. Although the public interface allows us to store the social network locally, this practice is considered impractical due to high time/space/communication cost and often violates the terms of use agreement. In light of this, in this paper we proceed under the assumption that (1) only external access to the social network is available; and (2) only a small number of users/nodes can be sampled. The main

[1]http://newsroom.fb.com/News/457/One-Billion-People-on-Facebook

insight offered by this work is that, even under these limitations, our algorithms achieve a good estimation accuracy of the network's structural measures.

This work focuses on two particular structural measures. The first measure is called the clustering coefficient. The second measure is the size of the network. Namely, the number of registered users in the network[2].

The clustering coefficient comes in two main flavors, (1) the network average clustering coefficient [12]; and (2) the global clustering coefficient [12]. Both measures are important for the understanding of the network structure. First, we introduce the local clustering coefficient of a node in a graph as the ratio of the number of edges between its neighbors to the maximal possible number of such edges. The network average clustering coefficient of a graph is the local clustering coefficient averaged over the set of nodes in the graph. The global clustering coefficient of a graph is the ratio of the number of triangles (ordered triples of different nodes in which are all nodes connected) to the number of connected triplets (ordered triples of different nodes in which consecutive nodes are connected).

The size of the network is one of the basic structural measures. The network size can determine the worth of a network (for business development). For certain applications in business development and advertisement, the size of a social network subpopulation is extremely important. For example, the number of users of an online product or the number of potential users for a product. The subpopulation fraction (which can also be estimated efficiently [15]) and the network size can determine the size of the subpopulation. Although some networks report their size periodically, the difference between consecutive reports can be more than ten percent. Moreover, even if this number is reported every day, an unbiased independent estimate would be beneficial.

This work contains three main contributions. The first and principal contribution is the first external access estimator for the global clustering coefficient. The second contribution is an improved external access estimator for the network average clustering coefficient. The third contribution is an improved external access estimator for the network size.

The rest of this paper is organized as follows. Section 2 surveys related work. Section 3 provided preliminaries and notations. Section 4 details our clustering coefficient estimators. Section 5 details our network size estimator. Section 6 reports our experimental results. We conclude the paper in Section 7.

2. BACKGROUND AND PRIOR WORK

We consider the social network as an undirected graph where nodes and edges are represented by users and friendship connections. Although the algorithms presented in this paper are correct for general graphs, the structure of social networks renders them even more effective.

Both the network average and the global clustering coefficient (also known as transitivity) are a long studied classical computer science problem. The running time of the naive algorithm for computing them is $O(n^3)$ for dense graphs (where n is the number of nodes in the graph), and it is considered impractical for large graphs. For the global clustering coefficient, the most challenging part of the computation is counting the total number of triangles, since computing the number of connected triplets is done in linear time. To this end, the computation of global clustering coefficient and the computation of the number of triangles is equivalent.

We provide references for a partial list (most recent) for several directions for estimating the number of triangles. Alon et al. [3] provided an exact algorithm for the counting the number of triangles. The running time of this algorithm is $O(E^{\frac{2\omega}{\omega+1}}) = O(E^{1.41})$, where $\omega < 2.376$ is the exponent of matrix multiplication. Avron [4] provided an estimator based on numerical matrix-vector multiplication using $O(\log^2 n)$ samples, each of which requires $O(|E|)$ time (where E is the set of nodes in the graph). Both these algorithm access the entire graph.

Buriol et al. [10] provided an approximate solution to the global clustering coefficient in the streaming model. The streaming models allows the algorithm to have a single pass on the input while (1) reading the edges in arbitrary/vertex ordered appearance (different algorithms) and (2) use constant amount of space. Becchetti et al. [9] provided an algorithm for the network average clustering coefficient in the streaming model. In contrast to [3, 4] these works assume there is no random access to the graph. However, the streaming algorithms access each edge at least once.

Schank et al. [27], provided estimators for both the global and network average clustering coefficient which only uses a sample of the nodes. However, unlike our work, the algorithms assume there is an efficient way to sample nodes with distribution that is tailored to the clustering coefficient. Specifically, for the network average clustering coefficient the sampling distribution is the uniform distribution and for the global clustering coefficient each node v_i with degree d_i is sampled proportionally to $d_i(d_i - 1)$. In contrast, the algorithms provided in this work do not even assume the number of nodes is known and does not require a tailored sampling distribution.

Another research direction [13, 25] addresses the problem of estimating the local clustering coefficient with external access[3]. In these papers, the graph can only be accessed via the exploration of nodes that lie on the frontier of previously explored nodes. Ribeiro et al [25] explored the graph using a random walk. Gjoka et al. [13] explored the graph using Metropolis-Hastings random walk that generates uniform samples from the nodes set. In both these papers, the computation requires augmenting the set of explored nodes, S, with further exploration of S's ego network. An ego network of a set of users S, is the set of users S' that contains all the users in S and all their (immediate) friends [13, 28].

In this work, we perform a random walk but remove the requirement of exploring the ego network. This difference is illustrated in Figure 1. The random walk contains three nodes v_1, v_2, v_3. Our approach requires the exploration of the nodes v_1, v_2, v_3 (marked by a thick circle), the ego network approach requires additional exploration of the nodes v_4, v_5, v_6. In total the ego network requires exploration of all the nodes $v_1, v_2, \ldots, v_6$ (marked by solid fill). In section 6, we show that the algorithm provided in this paper

[2]Technically, the algorithm estimates the size of the largest connected component and isolated users are neglected.

[3]Ref [25] mistakenly refers to the global clustering coefficient, but provides an accurate definition of the network average clustering coefficient.

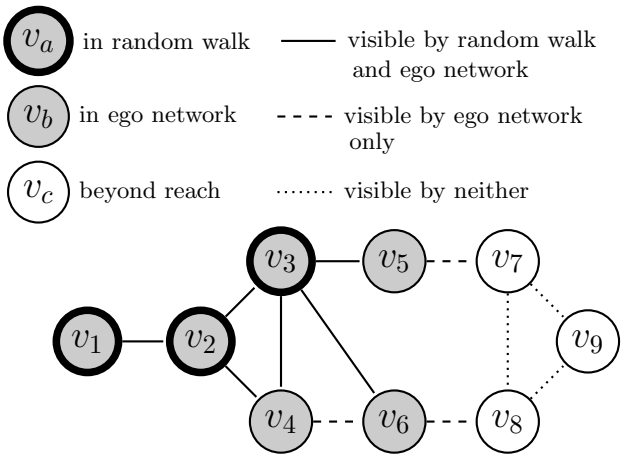

Figure 1: An example of a random walk with its corresponding ego network augmentation.

outperform competing approaches [25, 13] on all the social networks we study.

Another method for estimating the clustering coefficient from a random walk was presented in [14]. This algorithm uses only the ids of nodes visited by a random walk and does not assume any prior information. In contrast, the algorithms in this paper assume not only the node ids are visible, but also their list of friends (adjacency list). Practically, if this assumption holds, it renders [14] uncompetitive.

In this work, two estimators are provided for the clustering coefficient. The first for the network average clustering coefficient and the second for the global clustering coefficient. Both estimators use samples taken from a random walk on the graph. Namely, not only that the algorithms do not access the entire graph, they do not even have random access to the graph's nodes and edges. The only assumption is that a random walk can be performed via the public interface, and the visited node ids along with their list of friends (adjacency list). This is the case for many social networks. Indeed, the act of performing a random walk at all in an online social network typically necessitates having access to this information.

Both [14, 15] provide estimators for the total number of registered users in the network. These algorithms use only the node ids visited on the random walk and do not assume any prior information on the graph. The underlying idea in both papers is to count node collision, a pair of indices (k, l) such that the same node appears in the k^{th} and l^{th} location of the random walk. Nodes on the random walk are highly correlated when their index distances ($|k-l|$) are short, which increases the probability of a node collision. To ensure a unified probability of collision across all node pairs, a collision is counted only if the nodes appear a significant number of steps apart. These works differ in the way they select these pairs. In [15] the estimator chooses all pairs in which both k and l are a multiple of a parameter m, while [14] chooses all pairs in which $m \leq |k-l|$. Choosing all pairs [14] is practically better, but harder to analyze. The convergence of social network like graphs is very fast and depends on the degree distribution. For example, if the node degrees are distributed according to a Zipfian distribution with maximum degree of $\sqrt{n}$ and parameter $\alpha = 2$, then the number of samples needed to guarantee convergence for a fixed accuracy is $O(n^{1/4} \log n)$ [15].

In some applications the size of a subpopulation needs to be estimated. This subpopulation is defined by a property of the user's profile. For example, the number of registered users who use a specified online product. Estimating the size of a subpopulation requires multiplying the total size of the network by the ratio the target nodes to the total nodes which could also be estimated by the random walk [15]. In this work, we improve the network size estimation algorithms by using not only the visited node ids, but also the adjacency list of each of the visited nodes. This is done by counting node collision one step before they actually occur. Namely, two nodes on the random walk (enough nodes apart) that share a connection. We call this collision a neighbor collision.

3. PRELIMINARIES AND NOTATIONS

We denote by $G(V, E)$ the social network's underlying undirected graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of nodes (users) and E is the set of edges (friendship connections). Additionally, we denote by d_i the degree of node v_i and the sum of degrees by $D = \sum_{i=1}^{n} d_i = 2|E|$. The maximum degree of a node in the graph is noted by $d_{\max} = \max_{i=1}^{n} d_i$.

We denote by an $n \times n$ matrix A the adjacency matrix for graph G. Namely, $A_{i,k} = A_{k,i} = 1$ if node v_i is connected by an edge to node v_k and 0 otherwise. We assume no self loops, thus $A_{i,i} = 0$ for all i.

DEFINITION 1. *A triplet of nodes (v_j, v_i, v_k) is called connected if v_j is connected to v_i, v_i is connected to v_k, and $j < k$. Formally, if $A_{j,i} = 1$, $A_{i,k} = 1$, and $j < k$.*

DEFINITION 2. *A triangle is a connected triplet (v_j, v_i, v_k) in which v_j and v_k are connected. Formally, if $A_{j,k} = 1$.*

Following these definitions, a triplet of nodes is connected if $j < k$ and $A_{j,i}A_{i,k} = 1$ and it is a triangle if $j < k$ and $A_{i,j}A_{i,k}A_{j,k} = 1$. For a specific node v_i, the number of connected triplets (v_j, v_i, v_k) is thus $\sum_{j<k} A_{j,i}A_{i,k}$. Note that $\sum_{j<k} A_{j,i}A_{i,k} = d_i(d_i-1)/2$ since there are $d_i(d_i-1)/2$ choices for $j < k$ in which both $A_{j,i} = 1$ and $A_{i,k} = 1$. For a specific node v_i, the number of (v_j, v_i, v_k) triangles is denoted by $l_i = \sum_{j<k} A_{i,j}A_{i,k}A_{j,k}$ (it is also the number of edges between neighbors of v_i) .

DEFINITION 3. *The local clustering coefficient [12] for node v_i, denoted by c_i, is defined as the ratio of the number of (v_j, v_i, v_k) triangles to the number of (v_j, v_i, v_k) connected triplets. Formally,*

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

Note that $c_i \in [0, 1]$. In the case where $d_i = 1$ or $d_i = 0$, we have $c_i = 0$.

DEFINITION 4. *The network average clustering coefficient [12], denoted by $\mathfrak{c}_l$, is defined by*

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

DEFINITION 5. *The global clustering coefficient [12], denoted by $\mathfrak{c}_g$, is defined as the ratio of the total number of triangles to the total number of connected triplets. Formally,*

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

Note that a set of three nodes $\{v_j, v_i, v_k\}$ forms three different triangles[4] *one is counted in l_j, a second in l_i, and a third in l_k.*

The first step of the estimation algorithms is to generate a random walk. A random walk with r steps on G, denote by $R = (x_1, x_2, \ldots, x_r)$, is defined as follows: start from an arbitrary starting node v_{x_1}, then move to one of the neighboring nodes uniformly at random (with probability $\frac{1}{d_{x_i}}$) and repeat $r-1$ times. We use $\Pr[A]$ to denote the probability that event A occurred. We denote the distribution induced by R, as

$$\pi_R = (\Pr[x_r = 1], \Pr[x_r = 2], \ldots, \Pr[x_r = n]).$$

The probability $\Pr[x_r = i]$ after many random walk steps converges to $p_i \triangleq d_i/D$ and the vector $\pi = (p_1, p_2, \ldots, p_n)$ is called the stationary distribution of G.

In our estimators, we assume that x_1 is drawn from the stationary distribution[5]. This assumption is valid because we can always perform an initial random walk from an arbitrary node to draw a starting node from the stationary distribution.

The actual number of steps needed to converge to the stationary distribution depends on the mixing time of G. There are several definitions of mixing time, many of which are known to be equivalent up to constant factors. All definitions take an ϵ parameter to measure the distance between the stationary and the induced distribution. Both the book [17] and the survey [19] provide excellent overview on random walks and mixing times. We denote the mixing time of graph G by $\tau(\epsilon)$ or τ (ϵ is assumed to be a small constant). We use the following definition:

DEFINITION 6. *Let $R = (x_1, x_2, \ldots, x_r)$ be a random walk. Then, let the distance between π and π_R be the maximum difference between the probability of drawing a specific node x_r over all possible choices of nodes x_1 and x_r. Namely,*

$$d(r) = \max_{x_1=1}^{n} \max_{i=1}^{n} |p_i - Pr[x_r = i]|.$$

We have $\tau(\epsilon) = \min\{r \mid d(r) \le \epsilon\}$.

Social network graphs are known to have low mixing times and constant clustering coefficients (which are not extremely small). Recently, Addario-Berry et al [1] proved rigorously that the mixing time of Newman-Watts [23, 24] small world networks is $\Theta(\log^2 n)$. Mohaisen et al. [22] provide numerical evaluation of the mixing time of several networks. The authors claim that "the mixing time is much larger than anticipated". However, Table 1 and Figure 2 in their paper show that to have $d(r) \approx 0$, the number of steps should be $r = \log^2 n$ for the Facebook network, $r = 3\log^2 n$ for the DBLP and youtube networks, and $r = 10\log^2 n$ for the Live Journal network. Both the low mixing time and the relatively high value of the clustering coefficients enable the clustering coefficient estimation algorithms in this paper to provide accurate result with relatively low number of samples. Notations are summarized in Table 1.

G	underlying undirected graph
n	number of nodes in the graph
A	adjacency matrix for G
v_i	node in G
d_i	degree of node v_i
D	the sum all nodes degrees $\sum_{i=1}^{n} d_i$
r	total number of steps in the random walk
x_k	the index of k^{th} node in the random walk
p_i	$p(x_k = i) = \frac{d_i}{D}$
π	the stationary distribution $(p_1, p_2, \ldots, p_n)$
l_i	number of edges between neighbors of v_i
$\mathfrak{c}_l$	network average (local) clustering coefficient
$\mathfrak{c}_g$	global clustering coefficient
$\hat{\mathfrak{c}}_l$	$\mathfrak{c}_l$ estimation
$\hat{\mathfrak{c}}_g$	$\mathfrak{c}_g$ estimation
$\hat{n}$	n estimation
$\tau(\epsilon)$	mixing time
$d_{\max}$	$\max_{i=1}^{n} d_i$

Table 1: Summary of notations

4. CLUSTERING COEFFICIENT ESTIMATION

We now present the main observation used in both network average and global clustering coefficient estimators. Given a random walk $(x_1, x_2, \ldots, x_r)$, we define a new variable $\phi_k = A_{x_{k-1}, x_{k+1}}$ for every $2 \le k \le r-1$. For any function $f(x_k)$ the following holds[6]:

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

The first equality holds due to the law of total expectation. The second equality holds because there are d_i^2 equal probability combinations of (x_{k-1}, v_i, x_{k+1}) out of which only $2l_i$ form a triangle (v_j, v_i, v_k) or a reverse triangle (v_k, v_i, v_j). Notice that in a triangle or a reverse triangle v_j is connected to v_k ($A_{j,k} = 1$). The third equality holds due to algebraic manipulation.

4.1 Network average clustering coefficient

To estimate $\mathfrak{c}_l$, we introduce two variables. First, we define Φ_l as a weighted sum of ϕ_js, $\Phi_l = \frac{1}{r-2}\sum_{k=2}^{r-1} \phi_k \frac{1}{d_{x_k}-1}$. Second, we define Ψ_l as the sum of the sampled nodes reciprocal degrees, $\Psi_l = \frac{1}{r}\sum_{k=1}^{r} \frac{1}{d_{x_k}}$.

[4] In some references a triangle is defined by an unordered set of three nodes, in which case $\mathfrak{c}_g$ is defined by three times the ratio of the total number of triangles to the total number of connected triplets.

[5] This is not necessary in practice. However, the running time bound is tighter with this assumption.

[6] We choose $f(v_i) = 1/(d_i - 1)$ for the network average clustering and $f(v_i) = d_i$ for the global clustering estimator.

Using linearity of expectation and Eq (1) it is easy to compute Φ_l and Ψ_l expectation.

$$\mathrm{E}[\Phi_l] = \mathrm{E}\left[\phi_k \frac{1}{d_{x_k}-1}\right] = \sum_{i=1}^{n} \frac{1}{D}\frac{2l_i}{d_i(d_i-1)} = \frac{1}{D}\sum_{i=1}^{n} c_i$$

$$\mathrm{E}[\Psi_l] = \mathrm{E}\left[\frac{1}{d_{x_k}}\right] = \sum_{i=1}^{n} \frac{d_i}{D}\frac{1}{d_i} = \frac{n}{D}$$

From the above equations we can isolate $\mathfrak{c}_l$ and get that:

$$\mathfrak{c}_l = \frac{1}{n}\sum_{i=1}^{n} c_i = \frac{\mathrm{E}[\Phi_l]}{\mathrm{E}[\Psi_l]}$$

Intuitively, both Φ_l and Ψ_l converge to their expected values and the estimator Φ_l/Ψ_l converges to $\mathfrak{c}_l$ as well.

DEFINITION 7. *Let $\hat{\mathfrak{c}}_l$ be the estimator for $\mathfrak{c}_l$, defined as follows:*

$$\hat{\mathfrak{c}}_l \triangleq \frac{\Phi_l}{\Psi_l}.$$

LEMMA 1. *For any $\epsilon \le 1/8$ and $\delta \le 1$ we have:*

$$\Pr[\mathfrak{c}_l(1-\epsilon) \le \hat{\mathfrak{c}}_l \le \mathfrak{c}_l(1+\epsilon)] \ge 1-\delta$$

when the number of samples, r, satisfies:

$$r \ge r_l \in O\left(\frac{D}{n\mathfrak{c}_l}\tau(\epsilon)\right).$$

PROOF. The proof first finds the number of step, r_l, which guarantees both Φ_l and Ψ_l be within $\epsilon/3$ approximations to their expected values with probability at least $1-\delta/2$. See Appendix A for more details. Since the probability of Φ_l or Ψ_l deviating from their expected value is at most $\delta/2$, the probability of either Φ_l or Ψ_l deviating is at most δ (using the union bound). Then, we use the fact that

$$(1-\epsilon)\mathfrak{c}_l \le \frac{(1-\frac{\epsilon}{3})}{(1+\frac{\epsilon}{3})}\frac{\mathrm{E}[\Phi_l]}{\mathrm{E}[\Psi_l]} \le \frac{\Phi_l}{\Psi_l} \le \frac{(1+\frac{\epsilon}{3})}{(1-\frac{\epsilon}{3})}\frac{\mathrm{E}[\Phi_l]}{\mathrm{E}[\Psi_l]} \le (1+\epsilon)\mathfrak{c}_l$$

to complete the proof. □

Note that for social network like graph the mixing time is assumed to be relatively low (for Newman-Watts networks $\tau(\epsilon) = O(\log^2 n)$ [1]), $D = O(n)$ and $\mathfrak{c}_l$ is a small constant. Thus, the number of steps needed is linear in the mixing time, $\tau(\epsilon)$.

4.2 Global Clustering Coefficient

To estimate $\mathfrak{c}_g$, we introduce two variables. First, we define Φ_g as a weighted sum of ϕ_js, $\Phi_g = \frac{1}{r-2}\sum_{k=2}^{r-1}\phi_k d_{x_k}$. Second, we define Ψ_g as the sum of the sampled nodes degrees minus one, $\Psi_g = \frac{1}{r}\sum_{k=1}^{r} d_{x_k} - 1$.

Using linearity of expectation and Eq (1) it is easy to compute Φ_g and Ψ_g expectation.

$$\mathrm{E}[\Phi_g] = \mathrm{E}[\phi_k d_{x_k}] = \sum_{i=1}^{n}\frac{1}{D}\frac{2l_i}{d_i}d_i = \frac{1}{D}\sum_{i=1}^{n} 2l_i$$

$$\mathrm{E}[\Psi_g] = \mathrm{E}[d_{x_k}-1] = \sum_{i=1}^{n}\frac{d_i}{D}(d_i-1) = \frac{1}{D}\sum_{i=1}^{n} d_i(d_i-1)$$

From the above equations we can isolate $\mathfrak{c}_g$ and get that:

$$\mathfrak{c}_g = \frac{1}{\sum_{i=1}^{n} d_i(d_i-1)}\sum_{i=1}^{n} 2l_i = \frac{\mathrm{E}[\Phi_g]}{\mathrm{E}[\Psi_g]}.$$

Intuitively, both Φ_g and Ψ_g converge to their expected values and the estimator Φ_g/Ψ_g converges to $\mathfrak{c}_l$ as well.

DEFINITION 8. *Let $\hat{\mathfrak{c}}_g$ be the estimator for $\mathfrak{c}_g$, defined as follows:*

$$\hat{\mathfrak{c}}_g \triangleq \frac{\Phi_g}{\Psi_g}.$$

LEMMA 2. *For any $\epsilon \le 1/8$ and $\delta \le 1$ we have:*

$$\Pr[\mathfrak{c}_g(1-\epsilon) \le \hat{\mathfrak{c}}_g \le \mathfrak{c}_g(1+\epsilon)] \ge 1-\delta$$

when the number of samples, r, satisfies:

$$r \ge r_g \in O\left(\frac{Dd_{\max}}{\mathfrak{c}_g\sum_{i=1}^{n} d_i(d_i-1)}\tau(\epsilon)\right).$$

The proof is similar to the proof of Lemma 1, except the number of steps r_g that guarantees convergences for Φ_g and Ψ_g is different. See Appendix B for more details.

Both estimators presented in this section are consistent. Formally, as the number of samples, r, grows the estimators converge to the true value. This also implies the estimators are asymptotically unbiased.

5. NETWORK SIZE ESTIMATION

In this section we present an estimator for the graph size (number of nodes). The estimator uses observations of node pairs which are "far away" from each other in the random walk (as in Ref [14]). This assumption is needed to ensure both nodes in a pair are (approximately) uncorrelated: each drawn from the stationary distribution[7]. Specifically, the estimator examines node pairs whose index distance is greater than a threshold m. Formally,

$$I = \{(k,l) \mid m \le |k-l| \wedge 1 \le k,l \le r\}.$$

The estimator counts weighted neighbor collisions. A neighbor collision is a pair of indices (k,l) such that v_{x_k} and v_{x_l} share a common neighbor. Formally, let A_i be the set of vertices adjacent to v_i. Thus, $A_i \cap A_j$ is the set of nodes neighboring both v_i and v_j. Given a random walk $(x_1, x_2, \ldots, x_r)$, we define a new variable $\phi_{k,l} = |A_{x_k} \cap A_{x_l}|$. Note that if $(k,l) \in I$, then

$$\mathrm{E}\left[\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}\right] = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{d_i}{D}\frac{d_j}{D}|A_i \cap A_j|\frac{1}{d_i d_j} = \sum_{j=1}^{n}\left(\frac{d_j}{D}\right)^2.$$

To see why $\sum_{i=1}^{n}\sum_{j=1}^{n}|A_i \cap A_j| = \sum_{j=1}^{n} d_j^2$ consider the following combinatorial proof. For a node v_k, the number of connected triplets (v_i, v_k, v_j) with no restrictions on i and j is d_k^2. Thus, the total number of connected triplets is $\sum_{k=1}^{n} d_k^2$. Alternatively, for nodes v_i and v_j the number of connected triplets (v_i, v_k, v_j) is $|A_i \cap A_j|$. Thus, the total number of connected triplets can also be expressed by $\sum_{i=1}^{n}\sum_{j=1}^{n}|A_i \cap A_j|$.

Next, we define Φ_n to be the averaged value of $\phi_{k,l}\frac{1}{d_{x_k}d_{x_l}}$ over all possible choices of $(k,l) \in I$. Namely,

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

[7]The larger the value of m, the smaller the bias in the estimate introduced by this correlation, but increasing m means fewer observations of node pairs and a larger estimator variance. However, note that we again benefit from the fast-mixing nature of social graphs, and m need only be of the order $O(\log^2 n)$.

Let Ψ_n be the averaged sum of $\frac{d_{x_k}}{d_{x_l}}$ over all possible choices of $(k,l) \in I$. Formally,

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

Due to linearity of expectation, we have

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

$$\mathrm{E}[\Psi_n] = \mathrm{E}\left[\frac{d_{x_k}}{d_{x_l}}\right] = \sum_{i=1}^{n}\sum_{j=1}^{n}\frac{d_i}{D}\frac{d_j}{D}\frac{d_j}{d_i} = n\sum_{j=1}^{n}\left(\frac{d_j}{D}\right)^2$$

Notice that $n = \mathrm{E}[\Psi_n]/\mathrm{E}[\Phi_n]$. Intuitively, both Ψ_n and Φ_n converge to their expected values and the estimator Ψ_n/Φ_n converges to n as well.

DEFINITION 9. *Let $\hat{n}$ be the estimator for n, defined as follows:*

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

Prior art algorithm [14, 15] count the number of node collisions, C, and estimates n by Ψ_n/C. A node collision is a pair of indices (k,l) such that such that $x_k = x_l$. In contrast Φ_n counts neighbor collision and estimates n by Ψ_n/Φ_n.

LEMMA 3. *The neighbor collision estimator, $\hat{n}$ (definition 9), has confidence intervals tighter than the node collision estimator.*

PROOF. Formally, $C = \frac{1}{|I|}\sum_{(k,l)\in I} 1_{x_k=x_l}$ where $1_{x_k=x_l}$ is 1 if $x_k = x_l$ and 0 otherwise. The key observation is that

$$\mathrm{E}\left[1_{x_{k+1}=x_{l+1}} \mid x_k, x_l\right] = \phi_{k,l}\frac{1}{d_{x_k} d_{x_l}}.$$

This stems from the combinatorial argument that (a) there are $d_{x_k} d_{x_l}$ equally likely joint node transitions from x_k and x_l to x_{k+1} and x_{l+1}; and (b) in only $\phi_{k,l} = |A_{x_k} \cap A_{x_l}|$ of them $x_{k+1} = x_{l+1}$ holds. Note that, x_k is uncorrelated with x_l when $(k,l) \in I$. Using this observation we have,

$$\Phi_n = \frac{1}{|I|}\sum_{(k,l)\in I}\mathrm{E}\left[1_{x_{k+1}=x_{l+1}} \mid x_k, x_l\right].$$

This is the Conditional Monte Carlo estimator[8] of C, which guarantees $\mathrm{Var}[C] \geq \mathrm{Var}[\Phi_n]$ [26](Section 5.4). □

5.1 Implementation notes

The straight forward computation of Ψ_n and Φ_n running time is $O(r^2)$ and $O(r^2 d_{\max}^2)$ respectively. However, a careful implementation can reduce this complexity to $O(r)$ and $O(r d_{\max})$ respectively. For Φ_n the expected running can be reduced to $O(r\sum_{i=1}^{n}\frac{d_i^2}{D})$.

First, we define $(l+m)^+$ to be $\min\{r, l+m\}$ and $(l-m)^-$ to be $\max\{l-m, 1\}$. For the computation of Ψ_n instead of multiplying the value of $\frac{1}{d_{x_l}}$ by each d_{x_k} separately, it is multiplied by the sum of $\sum_{k=(l-m)^-}^{(l+m)^+} d_{x_k}$. The sum in turn, can be efficiently computed for every k in $O(1)$, using a cumulative sum precomputation. Specifically, if $B_q = \sum_{k=1}^{q} d_{x_k}$, then

$$|I|\,\Psi_n = \sum_{l=1}^{r}\frac{1}{d_{x_l}}\left(B_r - B_{(l+m)^+} + B_{(l-m)^-}\right).$$

To compute Φ_n one must first construct an inverted index of neighboring nodes. In document-term view, each node is a document containing adjacent nodes as terms. Specifically if v_j is a neighbor of x_k then k is a term in v_j. The running time of creating an inverted index is linear in the number of terms ($O(r d_{\max})$ worst case and $O(r\sum_{i=1}^{n}\frac{d_i^2}{D})$ expected). Then, the entry for v_j holds a list L_j of all indices in which v_j is a neighbor. Thus, $|I|\,\Phi_n = \sum_{j=1}^{n} C_j$, where $C_j = \sum_{(k,l)\in I | k\in L_j \wedge l\in L_j}\frac{1}{d_{x_k} d_{x_l}}$. To efficiently compute C_j in $O(|L_j|)$, a precomputation $B_q(j) = \sum_{q\geq k\in L_j}\frac{1}{d_{x_k}}$ should be used (similarly to the computation of Ψ_n).

6. EXPERIMENTAL EVALUATION

6.1 Networks with public dataset

We demonstrate the effectiveness of the estimators by experimenting with social networks with known structure. Datasets statistics are enclosed in Table 2.

Network	n	D/n	$\mathfrak{c}_l$	$\mathfrak{c}_g$
DBLP	977,987	8.457	0.7231	0.1868
Orkut	3,072,448	76.28	0.1704	0.0413
Flickr	2,173,370	20.92	0.3616	0.1076
Live Journal	4,843,953	17.69	0.3508	0.1179

Table 2: Networks statistics

In all our datasets we perform the following: (1) if the original network is directed, the direction is removed (the edge is made undirected); (2) only the network's largest connected component is retained and the rest of the nodes/users are dropped. All the datasets we use are publicly available[9].

DBLP In the "Digital Bibliography and Library Project" (DBLP[18]) dataset each entry is a reference to a paper which contains a title and a list of authors. In the corresponding network each node is an author and an edge between two authors represent co-authorship of one or more papers. We used a snapshot taken Oct 01, 2012.

Orkut Orkut is a general purpose social network. The dataset contains a partial snapshot (11.3% of the nodes) taken during 2006 by [21]. In this social network the friendship connections (edges) are undirected.

Flickr Flickr is an online social network with focus on photo sharing. The dataset contains a partial snapshot taken during 2006–2007 by [20]. In this social network the friendship connections (edges) are directed.

[8]Note that if $(k,l) \in I$, then $(k-1, l-1) \in I$ except for $k = 1$ which holds only for a negligible fraction of the pairs.

[9]The DBLP, Orkut, Flickr, and LiveJournal are publicly available at http://dblp.uni-trier.de/xml/ and http://konect.uni-koblenz.de/networks/{orkut-links,flickr-growth,soc-LiveJournal1} [16], respectivly.

LiveJournal LiveJournal is an on-line social network with focus on journals and blogs. The dataset contains a partial snapshot of the nodes taken by [5]. In this social network the friendship connections (edges) are directed.

The x-axis in our figures is the percentage of mined nodes (number of mined nodes over the total number of network nodes). The y-axis is the relative estimated value (estimate value over the true value). We display $[5\%, 95\%]$-confidence intervals for all figures. A $[5\%, 95\%]$-confidence interval of random variable z, is defined as the interval $[L, U]$ such that $\Pr[z \leq L] = 0.05$ and $\Pr[z \leq U] = 0.95$. Thus, $\Pr[z \in [L, U]] = 0.9$. To estimate the confidence interval, each simulation was run independently 100,000 times. The values L and U are estimated by the 5^{th} and 95^{th} percentile values respectively.

In subsections 6.2 and 6.3 we compare the prior art algorithms method with the random walk approach described in this work. For comparison we consider the following approaches: (1) the estimator based on random walk combined with ego network exploration described in [25] (labeled RW Ego network); and (2) the estimator based on Metropolis-Hastings sampling with ego network exploration described in [13] (labeled MH Ego Network). The estimator described in subsection 4.1 is labeled random walk. In the random walk estimator (our approach) the number of mined nodes is exactly the random walk's length, while in the Ego network algorithms (prior art) the mined nodes include the (sampled) walk nodes as well as their neighbors.

In subsection 6.4 we compare prior art node collision estimator [14, 15] (labeled node collision) with the new proposed neighbor collision estimator (labeled neighbor collision).

6.2 Network average clustering coefficient

Figure 2 displays confidence intervals for all algorithms and datasets. The proposed random walk estimator significantly outperforms ego network estimators. Specifically, using only 1% of the network size, the confidence intervals of the random walk estimator are about fifty percent tighter for the DBLP network and four times as tight for the Orkut, Flickr, and LiveJournal networks. The exact numbers are enclosed in Table 3.

Network	random walk	MH Ego	RW Ego
DBLP	[0.967, 1.033]	[0.942, 1.051]	[0.910, 1.073]
Orkut	[0.916, 1.085]	[0.583, 1.468]	[0.426, 1.658]
Flickr	[0.891, 1.111]	[0.557, 1.415]	[0.064, 2.023]
LiveJ	[0.951, 1.054]	[0.816, 1.200]	[0.645, 1.329]

Table 3: Network average clustering [5%,95%]-confidence interval for 1% mined nodes.

6.3 Global clustering coefficient

In this subsection there is no prior art algorithm for comparison. To have a baseline, we retrofit the ego network estimator for computing the global clustering coefficient. The global clustering coefficient can be viewed as a weighted sum of local clustering coefficients. The ego network sampling estimators multiplies each observed c_{x_k} by $w_k = d_{x_k}(d_{x_k} - 1)$ and divide the total by the sum $W = \sum_k w_k$.

Figure 3 displays confidence intervals for all algorithms and datasets. The proposed random walk estimator significantly outperforms ego network estimators by an even greater margin when compared with the network average clustering coefficient estimators. The curve for metropolis hasting ego network in missing in the Flickr graph because all the values are greater than 8, which demonstrate the estimator's inefficiency. In the LiveJournal graph, one can see the upper 95% curves are even increasing. These curves converge only after 5% of the network is sampled. Using only 1% of the network size, the confidence intervals of the random walk estimator are about three times tighter for the DBLP network and ten times tighter for the Orkut network. The ego network estimators for the Flickr and LiveJournal networks are extremely inaccurate in the $[0.1\%, 2\%]$ range. The exact numbers are enclosed in Table 4.

Network	random walk	MH Ego	RW Ego
DBLP	[0.869, 1.180]	[0.659, 1.919]	[0.609, 1.485]
Orkut	[0.892, 1.130]	[0.424, 2.711]	[0.317, 3.068]
Flickr	[0.922, 1.078]	[0.212, 10.07]	[0.176, 1.588]
LiveJ	[0.620, 1.523]	[0.235, 4.275]	[0.246, 3.051]

Table 4: Global clustering [5%,95%]-confidence interval for 1% mined nodes.

6.4 Network size

In this subsection we compare the node collision and neighbor collision estimators. In all estimators the number of mined nodes is exactly the random walk's length. We used $m = 2.5\%r$ as the separation parameter for all estimators. Namely, we used about 95% of the maximum number of (k, l) pairs ($|I| \approx 0.95r^2$). In Figure 4 we see that the neighbor collision estimator outperforms the node collision estimator. The node collision estimator and neighbor collision estimator are Ψ_n/C and Ψ_n/Φ_n respectively. The performance of the estimators depend on the variance of Ψ_n, C, and Φ_n. The performance of the neighbor collision reduces the variance of one factor, but retains the variance of Ψ_n. Therefore, we see a different performance impact on these datasets. Moreover, the fact that $\frac{1}{1\pm x} \approx 1 \mp x + x^2 \mp \cdots + x^{2k}$ explains why the neighbor collision estimator has a greater impact on performance in the early stages of convergence when r is small.

Using only 1% of the network size, there was a significant accuracy improvement in the DBLP network, a noticeable improvement for the Orkut network, and negligible improvement for the Flickr and LiveJ networks. The exact number are enclosed in Table 5. The second column is prior art node-collision estimator; the third column is the proposed new neighbor collision estimator; and the fourth column is the confidence bound improvement[10].

7. CONCLUSIONS

We presented algorithms for estimating the (1) network average clustering coefficient; (2) global clustering coefficient; and (3) the number of registered users. These algorithms use the information collected by random walk, namely, the ids of the visited nodes along with their adjacency list.

[10]In the DBLP network the change from 1.384 to 1.221 in the 95% confidence implies a $(0.384 - 0.221)/0.384$ improvement and the change in the 5% confidence from 0.752 to 0.815 implies a $(0.815 - 0.752)/(1 - 0.752)$ improvement.

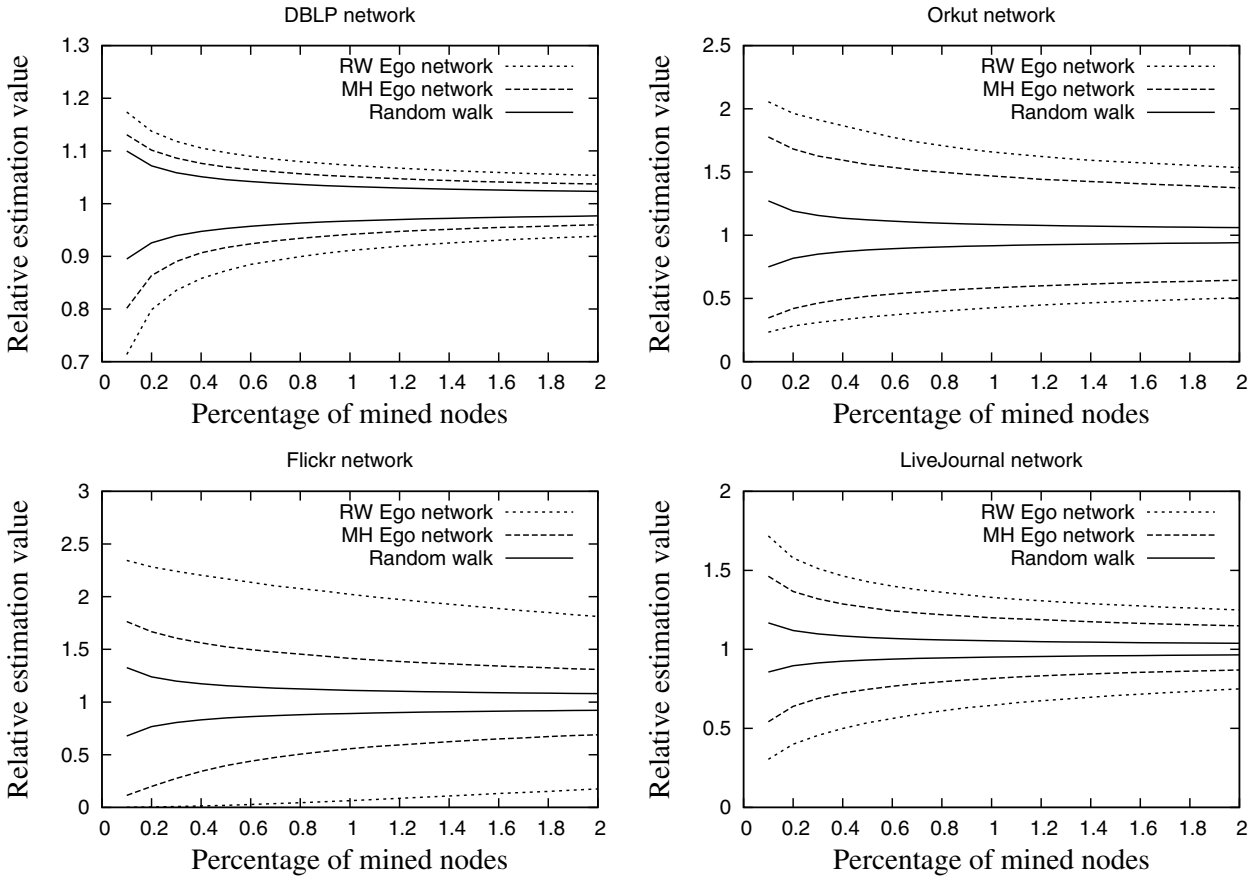

Figure 2: Estimation of the network average clustering coefficient confidence interval vs. the percentage of mined nodes.

Network	Node	Neighbor	improvement
DBLP	[0.752, 1.384]	[0.815, 1.221]	[25.4%, 42.5%]
Orkut	[0.849, 1.187]	[0.860, 1.161]	[7.30%, 13.9%]
Flickr	[0.846, 1.203]	[0.843, 1.208]	[1.91%, 2.40%]
LiveJ	[0.780, 1.232]	[0.785, 1.218]	[2.27%, 6.03%]

Table 5: Network size [5%,95%]-confidence interval for 1% mined nodes.

For the clustering coefficients algorithms we showed that (1) for social-network like graphs these algorithms considerably outperform prior art (sampling the ego network of each sampled node); and (2) an analytic bound on the number of steps required for convergence. For the number of registered users algorithm we showed, both analytically and experimentally, that the new suggested algorithm is strictly more accurate than prior art node collision algorithms.

Ego network algorithms sample all the adjacency lists of nodes in the random walk, while the random walk estimator samples only two nodes from this list (previous and next node of the random walk). Investigating between these two extremes might give rise to further improvement.

8. REFERENCES

[1] L. Addario-Berry and T. Lei. The mixing time of the newman–watts small world. In *SODA*, pages 1661–1668, 2012.

[2] Y.-Y. Ahn, S. Han, H. Kwak, S. B. Moon, and H. Jeong. Analysis of topological characteristics of huge online social networking services. In *WWW*, pages 835–844, 2007.

[3] N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.

[4] H. Avron. Counting triangles in large graphs using randomized matrix trace estimation. In *Large-Scale Data Mining: Theory and Applications (KDD Workshop)*, 2010.

[5] L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD*, pages 44–54, 2006.

[6] Z. Bar-Yossef and M. Gurevich. Random sampling from a search engine's index. *J. ACM*, 55(5), 2008.

[7] Z. Bar-Yossef and M. Gurevich. Estimating the impressionrank of web pages. In *WWW*, pages 41–50, 2009.

[8] Z. Bar-Yossef and M. Gurevich. Efficient search engine measurements. *TWEB*, 5(4):18, 2011.

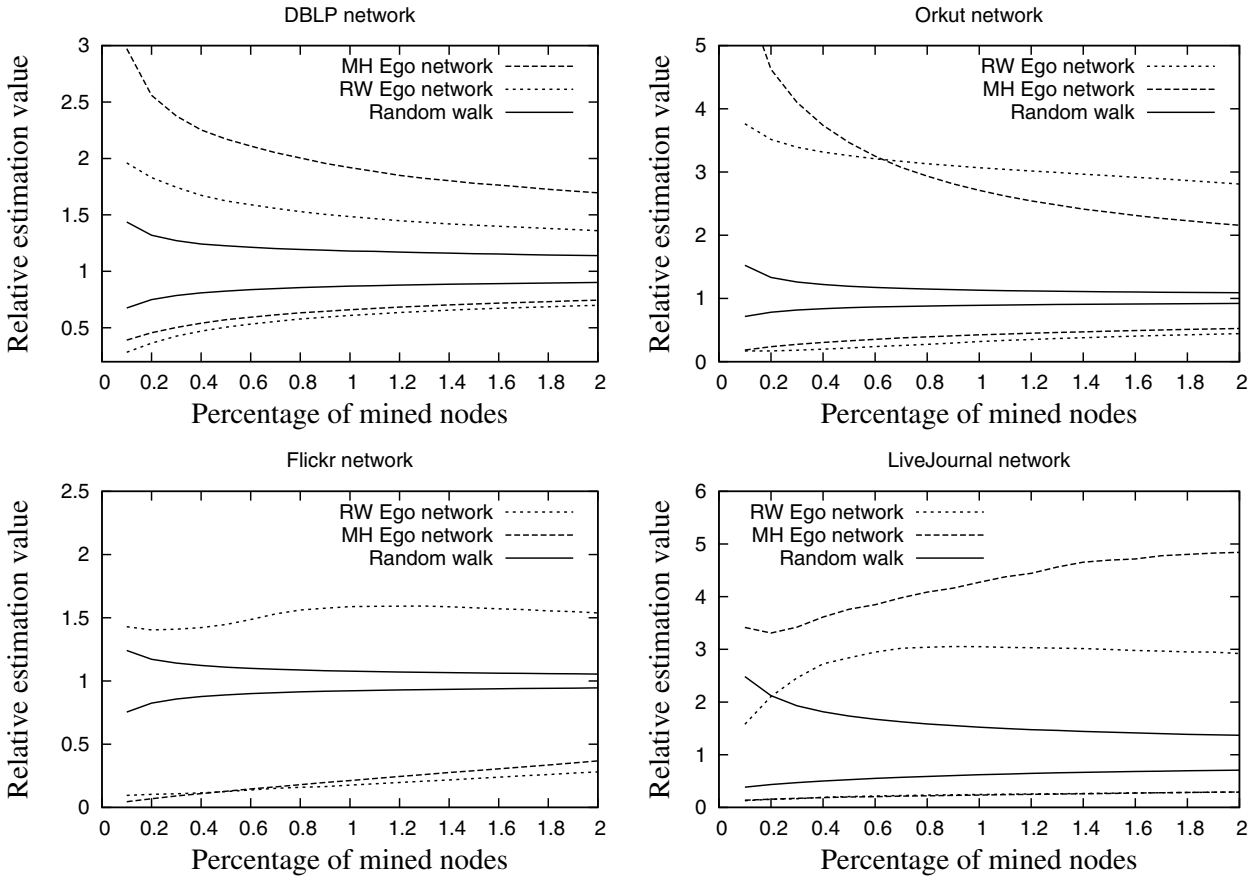

Figure 3: Estimation of the global clustering coefficient confidence interval vs. the percentage of mined nodes.

[9] L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient algorithms for large-scale local triangle counting. *TKDD*, 4(3), 2010.

[10] L. S. Buriol, G. Frahling, S. Leonardi, A. Marchetti-Spaccamela, and C. Sohler. Counting triangles in data streams. In *PODS*, pages 253–262, 2006.

[11] K.-M. Chung, H. Lam, Z. Liu, and M. Mitzenmacher. Chernoff-hoeffding bounds for markov chains: Generalized and simplified. In *STACS*, pages 124–135, 2012.

[12] L. F. Costa, F. A. Rodriguez, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, Aug. 2006.

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

[14] S. J. Hardiman, P. Richmond, and S. Hutzler. Calculating statistics of complex networks through random walks with an application to the on-line social network bebo. *European Physics Journal B*, 71(4):611–622, 2009.

[15] L. Katzir, E. Liberty, and O. Somekh. Estimating sizes of social networks via biased sampling. In *WWW*, pages 597–606, 2011.

[16] J. Kunegis. KONECT – the Koblenz Network Collection. http://konect.uni-koblenz.de/, 2012.

[17] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008.

[18] M. Ley. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *Proc. Int. Symp. on String Processing and Information Retrieval*, pages 1–10, 2002.

[19] L. Lovász and P. Winkler. Mixing times. microsurveys in discrete. In *DimacsWorkshop*, 1998.

[20] A. Mislove, H. S. Koppula, K. P. Gummadi, P. Druschel, and B. Bhattacharjee. Growth of the flickr social network. In *Proceedings of the 1st ACM SIGCOMM Workshop on Social Networks (WOSN'08)*, August 2008.

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

[22] A. Mohaisen, A. Yun, and Y. Kim. Measuring the mixing time of social graphs. In *Internet Measurement Conference*, pages 383–389, 2010.

[23] M. Newman and D. Watts. Renormalization group analysis of the small-world network model. *Physics Letters A*, 263:341–346, 1999.

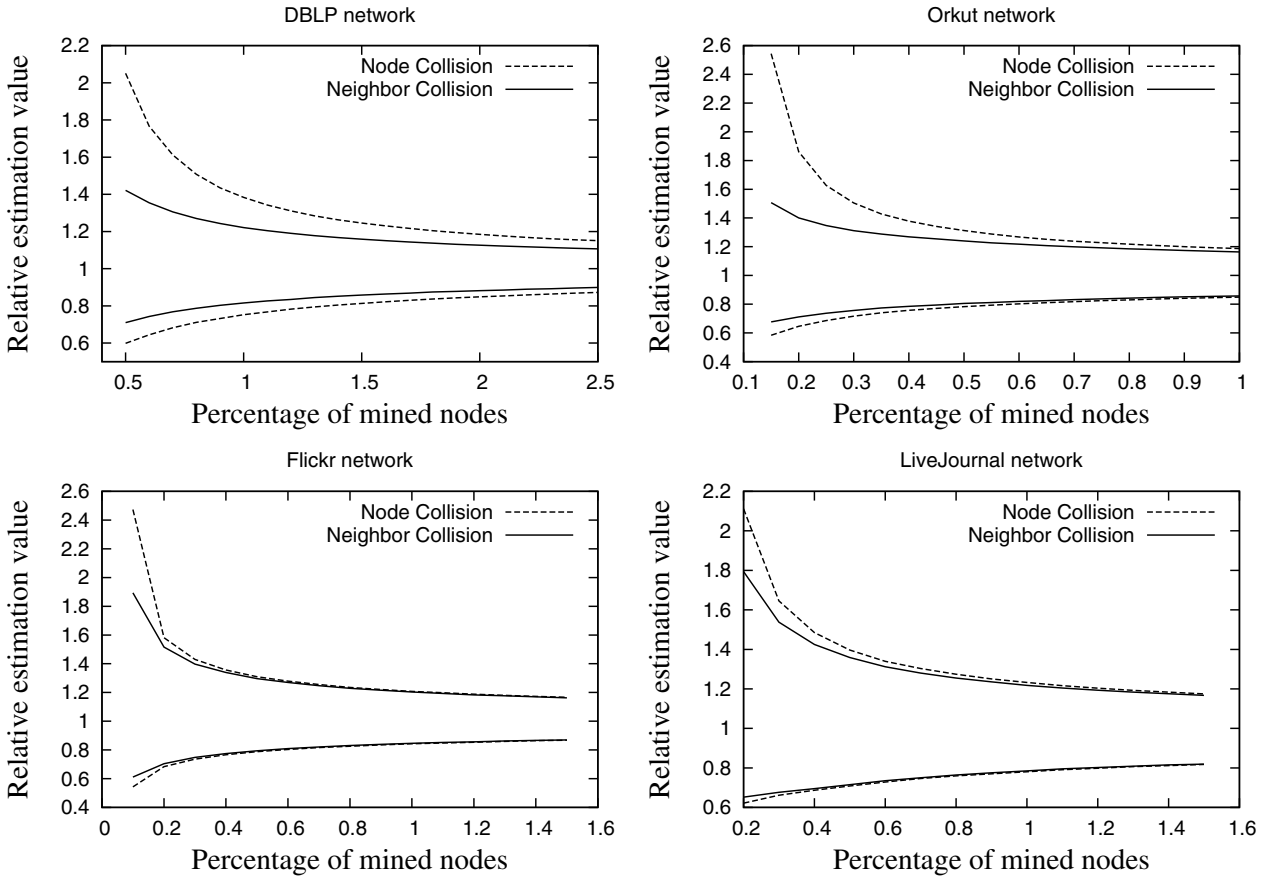

Figure 4: Estimation of the network size confidence interval vs. the percentage of mined nodes.

[24] M. Newman and D. Watts. Scaling and percolation in the small-world network model. *Physical Review E*, 60:7332–7342, 1999.

[25] B. F. Ribeiro and D. F. Towsley. Estimating and sampling graphs with multidimensional random walks. In *Internet Measurement Conference*, pages 390–403, 2010.

[26] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method*. Wiley Series in Probability and Statistics, 2 edition, 2007.

[27] T. Schank and D. Wagner. Approximating clustering coefficient and transitivity. *J. Graph Algorithms Appl.*, 9(2):265–275, 2005.

[28] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

APPENDIX

A. CONCENTRATION OF Ψ_L AND Φ_L

In the proof of Lemma 1 we required that the variables Ψ_l and Φ_l give an $\epsilon/3$ approximation to their expected values with probability at least $1-\delta/2$.

To prove both Ψ_l or Φ_l are concentrated we first restate a theorem from Chung et al. [11]:

THEOREM 4 (THEOREM 3.1 [11]). *Let M be an ergodic Markov chain with state space $[n]$ and stationary distribution π. Let $\tau = \tau(\epsilon)$ be its ϵ-mixing time for $\epsilon \leq \frac{1}{8}$. Let $(x_1, x_2, \ldots, x_r)$ denote an r-step random walk on M starting from an initial distribution φ on $[n]$, i.e., $x_1 \leftarrow \varphi$. Let $\|\varphi\|_\pi = \sum_{i=1}^n \frac{\varphi_i^2}{\pi_i}$. For every $k \in [r]$, let $f_k : [n] \to [0,1]$ be a weight function at step k such that the expected weight $E_{v \leftarrow \pi}[f_k(x_k)] = \mu$ for all k. Define the total weight of the walk $(x_1, x_2, \ldots, x_r)$ by $Z \triangleq \sum_{k=1}^r f_k(x_k)$. There exists some constant c (which is independent of μ, δ and ϵ) such that for $0 < \delta < 1$*

$$Pr[|Z - \mu r| > \epsilon \mu r] \leq c \|\varphi\|_\pi e^{-\epsilon^2 \mu r/72\tau},$$

or equivalently

$$Pr\left[\left|\frac{Z}{r} - \mu\right| > \epsilon\mu\right] \leq c \|\varphi\|_\pi e^{-\epsilon^2 \mu r/72\tau}.$$

LEMMA 5. *There is a constant value, ξ, such that if $r \geq r_{\Psi_l} = \xi \frac{D}{n} \tau(\epsilon)$, we have*

$$Pr\left[|\Psi_l - E[\Psi_l]| \leq \frac{\epsilon E[\Psi_l]}{3}\right] \geq 1 - \frac{\delta}{2}$$

PROOF. Let $f_k(x_k) = f(x_k) = \frac{1}{d_{x_k}}$. We assume that $\varphi \approx \pi$, and thus $\|\varphi\|_\pi = 1$. We have, $\mathrm{E}[\Psi_l] = \mathrm{E}\left[\frac{1}{d_{x_k}}\right] = \frac{n}{D}$. From Theorem 4,

$$\Pr\left[|\Psi_l - \mathrm{E}[\Psi_l]| > \frac{\epsilon}{3}\mathrm{E}[\Psi_l]\right] \leq c e^{-\epsilon^2 nr/9 \cdot 72 \cdot \tau D}$$

Extracting r_{Ψ_l} for which $\frac{\delta}{2} = ce^{-\epsilon^2 nr/9\cdot 72\cdot \tau D}$, we have $r_{\Psi_l} \leq \tilde{\xi}\log(\delta)\frac{1}{\epsilon^2}\frac{D}{n}\tau(\epsilon)$. Since ϵ and δ are constants, this ends the proof. □

LEMMA 6. *There is a constant value, ξ, such that if $r \geq r_{\Phi_l} = \xi\frac{D}{n\mathfrak{c}_l}\tau(\epsilon)$, we have*

$$Pr\left[|\Phi_l - E[\Phi_l]| \leq \frac{\epsilon E[\Phi_l]}{3}\right] \geq 1 - \frac{\delta}{2}$$

PROOF. For this bounds, we cannot apply Therorem 4 directly since f_j depends on previously visited node. However, since $\frac{A_{x_k,x_{k+2}}}{d_{x_k}-1}$ only depends on a 3-nodes history, we observe a related Markov chain that remembers the last three visited nodes. To this end, $\tilde{M}$ has $\tilde{n} = n \times n \times n$ nodes, and $(x_1, x_2, x_3) \leftarrow (x_2, x_3, x_4)$ with the same transition probability of x_3 to x_4 in M. Let $f_k(\tilde{x}_k) = \frac{A_{x_{k-1},x_{k+1}}}{d_{x_k}-1}$. We assume that $\varphi \approx \pi$, and thus $\|\varphi\|_\pi = 1$. We have, $\mathrm{E}[\Phi_l] = \mathrm{E}\left[\phi_k\frac{1}{d_{x_k}-1}\right] = \frac{1}{D}\sum_{i=1}^n c_i = \frac{n}{D}\mathfrak{c}_l$. From Theorem 4,

$$\Pr\left[|\Phi_l - \mathrm{E}[\Phi_l]| > \frac{\epsilon}{3}\mathrm{E}[\Phi_l]\right] \leq ce^{-\epsilon^2 n\mathfrak{c}_l(r-2)/9\cdot 72\cdot\tilde{\tau}D}$$

Extracting r_{Φ_l} for which $\frac{\delta}{2} = ce^{-\epsilon^2 n\mathfrak{c}_l(r-2)/9\cdot 72\cdot\tilde{\tau}D}$, we have $r_{\Phi_l} \leq \tilde{\xi}\log(\delta)\frac{1}{\epsilon^2}\frac{D}{n\mathfrak{c}_l}\tilde{\tau}$. Since ϵ and δ are constants, this ends the proof.

Note that $\tilde{\tau}(\epsilon) \leq \tau(\epsilon)$. To see this, in the true stationary distribution the probability of drawing x_{k-1}, x_k, x_{k+1} is $\frac{d_{x_{k-1}}}{D}\frac{1}{d_{x_k}}\frac{1}{d_{x_{k+1}}}$. After $\tau(\epsilon)$ steps, the probability of drawing x_{k-1} is at most ϵ distance away. Therefore, probability of drawing x_{k-1}, x_k, x_{k+1} is $\left(\frac{d_{x_{k-1}}}{D} \pm \epsilon\right)\frac{1}{d_{x_k}}\frac{1}{d_{x_{k+1}}}$, and thus the difference is bounded by $\epsilon\frac{1}{d_{x_k}}\frac{1}{d_{x_{k+1}}} \leq \epsilon$. □

To conclude we combine Lemma 5 and 6, and choose $r_l = \max\{r_{\Psi_l}, r_{\Phi_l}\}$.

B. CONCENTRATION OF Ψ_G AND Φ_G

In the proof of Lemma 2 we require that the variables Ψ_g and Φ_g give an $\epsilon/3$ approximation to their expected values with probability at least $1 - \delta/2$.

LEMMA 7. *There is a constant value, ξ, such that if $r \geq r_{\Psi_g} = \xi\frac{Dd_{\max}}{\sum_{i=1}^n d_i(d_i-1)}\tau(\epsilon)$, we have*

$$Pr\left[|\Psi_g - E[\Psi_g]| \leq \frac{\epsilon E[\Psi_g]}{3}\right] \geq 1 - \frac{\delta}{2}$$

PROOF. Let $f_k(x_k) = f(x_k) = \frac{d_{x_k}-1}{d_{\max}}$ (all values in $[0,1]$). We assume that $\varphi \approx \pi$, and thus $\|\varphi\|_\pi = 1$. We have, $\frac{1}{d_{\max}}\mathrm{E}[\Psi_g] = \mathrm{E}\left[\frac{d_{x_k}-1}{d_{\max}}\right] = \frac{1}{Dd_{\max}}\sum_{i=1}^n d_i(d_i-1)$. From Theorem 4,

$$\Pr\left[\left|\frac{\Psi_g}{d_{\max}} - \frac{\mathrm{E}[\Psi_g]}{d_{\max}}\right| > \frac{\epsilon}{3}\frac{\mathrm{E}[\Psi_l]}{d_{\max}}\right] \leq ce^{-\frac{\epsilon^2\sum_{i=1}^n d_i(d_i-1)r}{9\cdot 72\cdot\tau Dd_{\max}}}$$

Extracting r_{Ψ_g} for which $\frac{\delta}{2} = ce^{-\epsilon^2\sum_{i=1}^n d_i(d_i-1)r/9\cdot 72\cdot\tau Dd_{\max}}$, we have $r_{\Psi_g} \leq \tilde{\xi}\log(\delta)\frac{1}{\epsilon^2}\frac{Dd_{\max}}{\sum_{i=1}^n d_i(d_i-1)}\tau(\epsilon)$. Since ϵ and δ are constants, this ends the proof. □

LEMMA 8. *There is a constant value, ξ, such that if $r \geq r_{\Phi_g} = \xi\frac{Dd_{\max}}{c_g\sum_{i=1}^n d_i(d_i-1)}\tau(\epsilon)$, we have*

$$Pr\left[|\Phi_g - E[\Phi_g]| \leq \frac{\epsilon E[\Phi_g]}{3}\right] \geq 1 - \frac{\delta}{2}$$

PROOF. The proof combines the division by $d_{\max}$ of lemma 7 and the the 3-node history markov chain $\tilde{M}$ of lemma 6. □