Estimating Clustering Coefficients and Size of Social Networks via Random Walk

LIRAN KATZIR, Microsoft Research, Advanced Technology Labs, Herzliya, Israel
STEPHEN J. HARDIMAN, Research was conducted while the author was unaffiliated

This work addresses the problem of estimating social network measures. Specifically, the measures at hand are the network average and global clustering coefficients and the number of registered users. The algorithms at hand (1) assume no prior knowledge about the network and (2) access the network using only the publicly available interface. More precisely, this work provides (a) a unified approach for clustering coefficients estimation and (b) a new network size estimator. The unified approach for the clustering coefficients yields the first external access algorithm for estimating the global clustering coefficient. The new network size estimator offers improved accuracy compared to prior art estimators.

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 drops exponentially in the number of random walk steps. For the network size estimation we offer a generalized view of prior art estimators that in turn yields an improved estimator. All algorithms are validated on several publicly available social network datasets.

Categories and Subject Descriptors: F.2.2 [**Theory of Computing**]: Analysis of Algorithms and Problem Complexity

General Terms: Algorithms

Additional Key Words and Phrases: Estimation, sampling, clustering coefficient, social network

ACM Reference Format:
Liran Katzir and Stephen J. Hardiman. 2015. Estimating clustering coefficients and size of social networks via random walk. ACM Trans. Web 9, 4, Article 19 (September 2015), 20 pages.
DOI: http://dx.doi.org/10.1145/2790304

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 Ahn et al. [2007], Gjoka et al. [2010], and Mislove et al. [2007] 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.

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

Authors' addresses: L. Katzir, Yad Harutsim 7 Street (Beit Ampa - Harel), Herzliya, Israel; email: lirank@cs.technion.ac.il; S. J. Hardiman, Capital Fund Management, 23 rue de l'Universite, Paris 75007, France; email: hardimas@tcd.ie.

DOI: http://dx.doi.org/10.1145/2790304

Search engine public interfaces have been used in Bar-Yossef and Gurevich [2008, 2011] to estimate corpus size, index freshness, and density of duplicates, and in Bar-Yossef and Gurevich [2009] estimate the impressionrank of a web page. Online social network public interfaces have been used in Gjoka et al. [2010], Hardiman et al. [2009], Ribeiro and Towsley [2010], and Wang et al. [2014] to estimate the assortativity coefficient, degree distribution, and clustering coefficients of online social networks, as well as in Hardiman et al. [2009], Katzir et al. [2011], Kurant et al. [2012], Massoulié et al. [2006], and Ye and Wu [2010] 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 article 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 insight offered by this work is that, even under these limitations, the estimators presented in this work 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 [Costa et al. 2006] and (2) the global clustering coefficient [Costa et al. 2006]. 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 that have an edge between any two nodes) to the number of connected triplets (ordered triples of different nodes in which consecutive nodes have an edge).

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 (e.g., 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 [Katzir et al. 2011]) 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 10% . Moreover, even if this number is reported every day, an unbiased independent estimate would be beneficial.

This work contains three main contributions. The first contribution is the first external access estimator for the global clustering coefficient. The second contribution is an in-depth analysis of the convergence guarantees for the clustering coefficient estimators. The third and principal contribution is an improved external access estimator for the network size.

The rest of this article is organized as follows. Section 2 surveys related work. Section 3 provided preliminaries and notations. Section 4 details the clustering coefficient estimators and their analysis. Section 5 details the new generalized collision network

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

size estimator. Section 6 reports our experimental results. We conclude the article 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 article 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 directly 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. [1997] provided an exact algorithm for 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 [2010] provided an estimator based on numerical matrix-vector multiplication using $O(\log^2 n)$ nonzero entries, each of which requires $O(|E|)$ time (where E is the set of nodes in the graph). Both these algorithms access the entire graph.

Buriol et al. [2006] provided an approximate solution to the global clustering coefficient in the streaming model. The streaming model 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 a constant amount of space. Becchetti et al. [2010] provided an algorithm for the network average clustering coefficient in the streaming model. In contrast to Alon et al. [1997] and Avron [2010] these works assume there is no random access to the graph. However, the streaming algorithms access each edge at least once.

Schank and Wagner [2005] 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.

The problem of estimating the network average clustering coefficient using external access is addressed by Gjoka et al. [2013] and Ribeiro and Towsley [2010].[3] Gjoka et al. [2013][4] provides an estimator for the clustering coefficient distribution per node degree using external access only (the estimator is named "Traversed Edges"). These per degree clustering coefficients can in turn estimate the network average clustering coefficient. Both estimators Gjoka et al. [2013] ("Traversed Edges") and Ribeiro and Towsley [2010] (Section 4.2.4) are identical and fall into the unified framework presented in this article. The clustering coefficient related contributions of this work are:

[3]Reference Ribeiro and Towsley [2010] mistakenly refers to the global clustering coefficient, but provides an accurate definition of the network average clustering coefficient.

[4]Gjoka et al. [2013] was published in parallel with the conference version of this article.

(a) the first estimator for the global clustering coefficient; (b) a rigorous proof that both the global and network average clustering estimators are consistent; and (c) analysis for the required number of samples to convergence.

Another method for estimating the clustering coefficient from a random walk was presented in Hardiman et al. [2009]. 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 article assume not only the node IDs are visible, but also their list of friends (adjacency list). Practically, if this assumption holds, it renders Hardiman et al. [2009] uncompetitive.

In this work, a unified approach is presented that captures two estimators for the clustering coefficient: the first estimator for the network average clustering coefficient and the second estimator for the global clustering coefficient. Both estimators use samples taken from a random walk on the graph. Namely, not only do the algorithms 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.

There are two different approaches in the literature for estimating the total number of registered users in the network (also known as graph size estimation): node collision based [Hardiman et al. 2009; Katzir et al. 2011; Massoulié et al. 2006; Ye and Wu 2010] and edge collision based [Kurant et al. 2012]. The estimators in Hardiman et al. [2009], Katzir et al. [2011], Massoulié et al. [2006], and Ye and Wu [2010] use only the node IDs visited on the random walk and do not assume any prior information on the graph, while Kurant et al. [2012] also assumes access to immediate friends list. Both Massoulié et al. [2006] and Ye and Wu [2010] sample nodes uniformly. In Katzir et al. [2011] it is shown that sampling from the graph's stationary distribution produces more accurate results. The underlying idea in Hardiman et al. [2009] and Katzir et al. [2011] 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. In contrast, the estimator provided in Kurant et al. [2012] counts edge collision. A pair of edges is considered as an edge collision if they share at least one end point. Nodes on the random walk are highly correlated when their index distances ($|k-l|$) are short, which increases the probability of a node/edge 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. The collision works differ in the way they select these pairs. In Katzir et al. [2011] the estimator chooses all pairs in which both k and l are a multiple of a parameter m, while Hardiman et al. [2009] and Kurant et al. [2012] choose all pairs in which $m \leq |k-l|$. Choosing all pairs [Hardiman et al. 2009] 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 $O(n^{1/4} \log n)$ [Katzir et al. 2011].

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 of the target nodes to the total nodes, which could also be estimated by the random walk [Katzir et al. 2011].

In this work, we describe a new generalized collision estimator that contains a tuning parameter. A specific value of this parameter renders the generalized estimator equivalent to the node collision estimator and another value renders equivalence to the edge collision estimator. Thus, the two prior-art estimators are as a special case of the new generalized collision estimator. Finally, we present an adaptation of the

jackknife technique to random walk samples. In turn, we use this technique to tune the parameter value.

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. We denote by A_i the set of vertices adjacent to node v_i. Thus, $A_i \cap A_j$ is the set of neighbor nodes common to v_i and v_j.

Definition 3.1. A triplet of nodes (v_j, v_i, v_k) is called connected if there is an edge between v_j and v_i, there is an edge between v_i and v_k, and $j < k$. Formally, if $A_{j,i} = 1$, $A_{i,k} = 1$, and $j < k$.

Definition 3.2. A triangle is a connected triplet (v_j, v_i, v_k) in which there is an edge between v_j and v_k. Formally, if (v_j, v_i, v_k) is a connected triplet and $A_{j,k} = 1$. Namely, $A_{j,i} = 1$, $A_{i,k} = 1$, $A_{j,k} = 1$, and $j < k$.

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 3.3. The local clustering coefficient [Costa et al. 2006] 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 3.4. The network average clustering coefficient [Costa et al. 2006], denoted by $\mathfrak{c}_l$, is defined by

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

Definition 3.5. The global clustering coefficient [Costa et al. 2006], 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:[5] one is counted in l_j, a second in l_i, and a third in l_k.

[5]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.

The first step of the estimation algorithms is to generate a random walk. A random walk with r steps on G, denoted 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.[6] 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 [Levin et al. 2008] and the survey [Lovász and Winkler 1998] provide excellent overview on random walks and mixing times. We denote the mixing time of graph G by $\tau_{mix}(\epsilon)$. We use the following definition:

Definition 3.6. 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 $x_r = i$ and the probability of drawing i in the stationary distribution over all i and starting nodes x_1. Namely,

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

We have $\tau_{mix}(\epsilon) \triangleq \min\{r \mid d(r) \leq \epsilon\}$, and $\tau_{mix} \triangleq \tau_{mix}(1/8)$.

The following inequality $\tau_{mix}(\epsilon) \leq \lceil \log_2 1/\epsilon \rceil \tau_{mix}$ states that the value of ϵ effects the value of the mixing time by (at most) a logarithmic amount [Levin et al. 2008]. Social network graphs are known to have low mixing times and constant clustering coefficients (which are not extremely small). Recently, Addario-Berry and Lei [2012] proved rigorously that the mixing time of Newman and Watts [1999a, 1999b] small world networks is $\Theta(\log^2 n)$. Mohaisen et al. [2010] provide numerical evaluation of the mixing time of several networks. The empirical evidence provided by Mohaisen et al. [2010] supports the claim that the theoretical argument by Addario-Berry and Lei [2012] extends to real world social networks. Specifically, Table 1 and Figure 2 in Mohaisen et al. [2010] 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 article to provide accurate results with a relatively low number of samples. Notations are summarized in Table I.

4. CLUSTERING COEFFICIENT ESTIMATION

The main intuition behind the estimator is that while traveling along a random walk, we get many connected triplets. Testing if these triplets are triangles with the right weighting gives an approximation for the clustering coefficients. We now present the

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

Table I. Summary of Notations

G	underlying undirected graph
n	number of nodes in the graph
A	adjacency matrix for G
v_i	i^{th} node in G
A_i	the set of v_i's neighbors
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
$\widehat{\mathfrak{c}}_l$	$\mathfrak{c}_l$ estimation
$\widehat{\mathfrak{c}}_g$	$\mathfrak{c}_g$ estimation
$\hat{n}$	n estimation
$\tau_{mix}(\epsilon)$	mixing time
τ_{mix}	$\tau_{mix}(1/8)$
$d_{\max}$	$\max_{i=1}^{n} d_i$

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:

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

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.

The unified approach defines two random variables. First, Φ is defined as a weighted sum of ϕ_js, $\Phi = \frac{1}{r-2} \sum_{k=2}^{r-1} \phi_k f(x_k)$. Second, Ψ is defined as follows: $\Psi = \frac{1}{r} \sum_{k=1}^{r} f(x_k) \frac{d_{x_k} - 1}{d_{x_k}}$. The ratio of these two variables yields the desired estimate. As you will see, the estimator for the global and network average clustering coefficients differ only in the choice of $f(x_k)$. Specifically, $f(x_i) = 1/(d_i - 1)$ for the network average clustering and $f(x_i) = d_i$ for the global clustering estimator.

4.1. Network Average Clustering Coefficient

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

expectation and Equation (1) it is easy to compute Φ_l and Ψ_l expectation.

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

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

From the preceding 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 4.1. Let $\widehat{\mathfrak{c}}_l$ be the estimator for $\mathfrak{c}_l$, defined as follows:

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

LEMMA 4.2. *For any $\epsilon \le 1/8$ and $\delta \le 1$ we have*

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

when the number of samples, r, satisfies

$$r \ge r_l \in O\left(\frac{D}{n\mathfrak{c}_l}\log\left(\frac{1}{\delta}\right)\frac{\lceil \log 1/\epsilon \rceil}{\epsilon^2}\tau_{mix}\right).$$

PROOF. The proof first finds the number of steps, r_l, which guarantees both Φ_l and Ψ_l are 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

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

to complete the proof. □

Note that for a social-network-like graph the mixing time is assumed to be relatively low (for Newman-Watts networks $\tau_{mix}(\epsilon) = O(\log^2 n)$ [Addario-Berry and Lei 2012]), $D = O(n)$ and $\mathfrak{c}_l$ is a small constant.

4.2. Global Clustering Coefficient

To estimate $\mathfrak{c}_g$, Φ_g is written $\Phi_g = \frac{1}{r-2}\sum_{k=2}^{r-1}\phi_k d_{x_k}$ and Ψ_g as the sum of the sampled node degrees minus one, $\Psi_g = \frac{1}{r}\sum_{k=1}^{r} d_{x_k} - 1$. Using linearity of expectation and Equation (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 preceding 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 4.3. Let $\widehat{\mathfrak{c}}_g$ be the estimator for $\mathfrak{c}_g$, defined as follows:

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

LEMMA 4.4. *For any $\epsilon \le 1/8$ and $\delta \le 1$ we have*

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

when the number of samples, r, satisfies

$$r \ge r_g \in O\left(\frac{Dd_{\max}}{\mathfrak{c}_g \sum_{i=1}^{n} d_i(d_i-1)} \log\left(\frac{1}{\delta}\right) \frac{\lceil \log 1/\epsilon \rceil}{\epsilon^2} \tau_{mix}\right).$$

The proof is similar to the proof of Lemma 4.2, 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.

4.3. Conditional Monte Carlo Improved Estimation

Both $\widehat{\mathfrak{c}}_l$ and $\widehat{\mathfrak{c}}_g$ estimators use $\phi_k = A_{x_{k-1},x_{k+1}}$. To improve the estimator accuracy, we replace ϕ_k with its conditional expectation. Namely, $\mathrm{E}[\phi_k | x_{k-1} = j, x_k = i]$. This process is known as Conditional Monte Carlo [Rubinstein and Kroese 2007] (Section 5.4) and it guarantees the accuracy is at least as good as the original estimator. This version of the estimators requires not only to test friendship between nodes (edge), but also to retrieve the entire friends list (adjacency list) of any node in the random walk.

Formally, the number of neighbors v_i and v_j share is $|A_i \cap A_j|$. We use the observation that

$$\mathrm{E}[\phi_k | x_{k-1} = j, x_k = i] = \frac{1}{d_i} |A_i \cap A_j|.$$

To see why, consider iterating over all possible and equally probable choices of x_{k+1}. There are d_i such choices and in only $|A_i \cap A_j|$ (number of common neighbors) of them $\phi_k = 1$.

The updated equations are summarized next.

$$\widetilde{\phi}_k = \frac{1}{d_{x_k}} |A_{x_{k-1}} \cap A_{x_k}|,$$

$$\widetilde{\Phi}_l = \frac{1}{r-1} \sum_{k=2}^{r} \tilde{\phi}_k \frac{1}{d_{x_k}-1} = \frac{1}{r-1} \sum_{k=2}^{r} \frac{|A_{x_{k-1}} \cap A_{x_k}|}{d_{x_k}(d_{x_k}-1)},$$

$$\widetilde{\Phi}_g = \frac{1}{r-1} \sum_{k=2}^{r} \tilde{\phi}_k d_{x_k} = \frac{1}{r-1} \sum_{k=2}^{r} |A_{x_{k-1}} \cap A_{x_k}|.$$

Note that $\mathrm{E}[\phi_k | x_{k-1} = j, x_k = i]$ is not symmetric and depends on the order of x_{k-1} and x_k. A random walk on an undirected graph induces a reversible Markov chain.

Therefore, the probability of a walk $R = (x_1, x_2, \ldots, x_r)$[7] is equal to the probability of the reversed walk $R'(x_r, x_{r-1}, \ldots, x_1)$. Thus, one can use R and R' to produce an estimate. This results in the following formula for $\widetilde{\Phi}_l$:

$$\widetilde{\Phi}_l = \frac{1}{2(r-1)} \sum_{k=2}^{r} \frac{|A_{x_{k-1}} \cap A_k|}{d_{x_k}(d_{x_k}-1)} + \frac{|A_{x_{k-1}} \cap A_k|}{d_{x_{k-1}}(d_{x_{k-1}}-1)}.$$

The Conditional Monte Carlo (CMC) estimators for the network average and global clustering coefficient are $\widetilde{\Phi}_l/\Psi_l$ and $\widetilde{\Phi}_g/\Psi_g$, respectively.

5. NETWORK SIZE ESTIMATION

In this section, we present an estimator for the graph size (number of nodes). We refer to the new estimator as the generalized collision estimator since it holds prior work algorithms as a special case. The estimator counts node pairs (collisions) that appear in two distinct locations of the random walk. The key difference between the generalized estimator and prior work is the use of a weight function from the nodes set to scalars, $w : \{v_1, v_2, \ldots, v_n\} \to \Re$, which gives a different weight for each collision.

The estimator uses observations of node pairs that are "far away" from each other in the random walk (as in Hardiman et al. [2009]). This assumption is needed to ensure both nodes in a pair are (approximately) uncorrelated: each drawn from the stationary distribution. Specifically, the estimator examines node pairs whose index distance is greater than a threshold m. We describe how to choose m in the simulation section.[8] Formally,

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

Given a random walk $(x_1, x_2, \ldots, x_r)$, we define a new variable $\phi_{k,l}$, which is equal to 1 if the node x_k is equal to the node x_l. Namely,

$$\phi_{k,l} = 1_{\{x_k = x_l\}}.$$

Note that if $(k,l) \in I$, then

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

To see why, notice that the probability of $x_k = v_j$ is d_j/D, the probability of $x_l = v_j$ is d_j/D, and these events are independent under the assumption that m is at least the mixing time. Next, we define $\Phi_n(w)$ to be the weighted averaged value of $w(x_k)\phi_{k,l}$ over all possible choices of $(k,l) \in I$. Namely,

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

Let $\Psi_n(w)$ be the weighted average of $w(x_k)d_{x_k}/d_{x_l}$ over all possible choices of $(k,l) \in I$. Formally,

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

[7]The probability of drawing R from the stationary distribution is $\frac{d_{x_1}}{D}\frac{1}{d_{x_1}}\frac{1}{d_{x_2}}\cdots\frac{1}{d_{x_{r-1}}} = \frac{1}{D}\frac{1}{d_{x_2}d_{x_3}\cdots d_{x_{r-1}}}$.

[8]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 $O(\log^2 n)$.

Due to linearity of expectation, we have

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

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

$$= n\sum_{j=1}^{n} w(v_j)\left(\frac{d_j}{D}\right)^2.$$

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

Definition 5.1. Let $\widehat{n}(w)$ be the estimator for n given w, defined as follows:

$$\widehat{n}(w) \triangleq \frac{\Psi_n(w)}{\Phi_n(w)}.$$

5.1. Conditional Monte Carlo Improved Estimation

The conditional Monte Carlo method [Rubinstein and Kroese 2007] (Section 5.4) (used in Section 4.3) can also be applied to the size estimator. To improve the estimator accuracy, we replace $\phi_{k,l}$ with its conditional expectation. Namely, $\phi_{k,l}$ is replaced with $\mathrm{E}[\phi_{k,l}|x_k = j, x_{l-1} = i]$. This method guarantees the accuracy is at least as good as the original estimator. This version of the estimators requires retrieval of the entire friends list (adjacency list) of any node in the random walk.

Formally, the value of $A_{i,j}$ is 1 if and only if node v_i is connected to node v_j. We use the observation that

$$\mathrm{E}[\phi_{k,l}|x_k = j, x_{l-1} = i] = \frac{1}{d_i}A_{i,j}.$$

To see why, consider iterating over all possible and equally probable choices of x_l. If there is no edge between v_i and v_j ($A_{i,j} = 0$) the expected value is 0. If there is an edge ($A_{i,j} = 1$), there are d_i such choices and in only one of them $\phi_{k,l} = 1$.

The updated equations are summarized next.

$$\widetilde{\phi}_{k,l} \triangleq \mathrm{E}[\phi_{k,l}|x_k = j, x_{l-1} = i] = \frac{1}{d_{x_{l-1}}}A_{x_k,x_{l-1}},$$

$$\widetilde{\Phi}_n(w) = \frac{1}{|I|}\sum_{(k,l)\in I} w(x_k)\tilde{\phi}_{k,l} = \frac{1}{|I|}\sum_{(k,l)\in I}\frac{w(x_k)}{d_{x_{l-1}}}A_{x_k,x_{l-1}}.$$

The CMC estimator for n is given by $\widehat{n}_{cmc}(w) \triangleq \Psi_n(w)/\widetilde{\Phi}_n(w)$.

5.2. Optimizing the Weight Function

The formal problem of estimating the optimal weight function w_{opt} (that which minimizes the variance of the estimate) is statistically much harder than estimating n. To this end, we restrict w to a parametric form $w_\alpha(v_i) = d_i^\alpha$ for some constant α.

We treat $\widehat{n}_{cmc}(w_{\alpha_1}), \widehat{n}_{cmc}(w_{\alpha_2}), \ldots, \widehat{n}_{cmc}(w_{\alpha_a})$ as a set of a different estimators, whose variance is $\sigma_1^2, \sigma_2^2, \ldots, \sigma_a^2$, respectively. First, we estimate the variance for each value of α, denoted by $\widehat{\sigma}_1^2, \widehat{\sigma}_2^2, \ldots, \widehat{\sigma}_a^2$. Then, α is chosen to be the value that induced the least variance. Namely, $\alpha = \mathrm{argmin}_{\alpha_p}\widehat{\sigma}_p^2$.

The estimation of the variance ($\widehat{\sigma}_p^2$) is done using blockwise jackknife method [Künsch 1989].[9] Formally, the random walk $R = (x_1, x_2, \ldots, x_r)$ is partitioned into k-blocks of roughly the same size $R = (b_1, b_2, \ldots, b_k)$. Then, the i'th pseudosample is $R_{-i} = (b_1, b_2, \ldots, b_{i-1}, b_{i+1}, \ldots, b_k)$. To estimate the variance of an estimator $\widehat{n}$, an estimate is produced for each pseudosample $\widehat{n}_i$ as a function of R_{-i}[10]. Next, the average estimate $\widehat{n}_{(\cdot)} = \frac{1}{k}\sum_{i=1}^{k}\widehat{n}_i$ is computed. Finally, the variance $\widehat{\sigma}^2 \triangleq \frac{1}{k}\sum_{i=1}^{k}(\widehat{n}_i - \widehat{n}_{(\cdot)}^2)$.

We conclude by noting that the estimated $\widehat{n}_{cmc}(w)$ generalize prior work. When $\alpha = 0$ ($w_\alpha(v_j) = d_j^0 = 1$), we get that $\widehat{n}_{cmc}(w_0)$ and the induced edges estimator [Kurant et al. 2012] (Eq. (24)) are identical. When $\alpha = 1$ ($w_\alpha(v_j) = d_j$), we get that $\widehat{n}_{cmc}(w_1)$ and the Conditional Monte Carlo version of node collision estimator [Hardiman et al. 2009; Katzir et al. 2011] are identical.

5.3. Implementation Notes

The input for $\widehat{n}_{cmc}$ is the random walk $R = (x_1, x_2, \ldots, x_r)$ and the list of friends for each x_i. We denote the size of the friends input by $L = \sum_{i=1}^{r} d_{x_i}$ (the expected value of L is $\sum_{i=1}^{n}\frac{d_i^2}{D}$). The straightforward computation of Ψ_n and Φ_n running times are $O(r^2)$ and $O(rL)$, respectively. However, a careful implementation can reduce the running times to $O(r)$ and $O(L)$, respectively.

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 $O(L)$. Then, the entry for v_j holds a list L_j of all indices in which v_j is a neighbor. Thus, $|I|\Phi_n = \sum_{k=1}^{n} C_k$, where $C_k = \sum_{(k,l)\in I | l\in L_{x_k}}\frac{w(x_k)}{d_{x_{l-1}}}$. To efficiently compute C_k in $O(|L_k|)$, a precomputation $B_q(k) = \sum_{q\geq l, q\in L_k}\frac{1}{d_{x_{q-1}}}$ 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. Dataset statistics are shown in Table II.

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.[11]

[9]Reference Efron and Tibshirani [1993] is a great resource for jackknife and bootstrapping methods. Blockwise bootstrapping can also be applied [Härdle et al. 2003].

[10]Technically, we found that pseudosamples that remove two different blocks $R_{-i,j}$ produce a slightly better estimate.

[11]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} [Kunegis 2012], respectively.

Table II. Network Statistics

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

DBLP. In the "Digital Bibliography and Library Project" (DBLP [Ley 2002]) dataset each entry is a reference to a paper that contains a title and a list of authors. In the corresponding network each node is an author, and an edge between two authors represents coauthorship of one or more papers. We used a snapshot taken Oct. 1, 2012.

Orkut. Orkut is a general-purpose social network. The dataset contains a partial snapshot (11.3% of the nodes) taken during 2006 by Mislove et al. [2007]. In this social network the friendship connections (edges) are undirected.

Flickr. Flickr is an online social network with a focus on photo sharing. The dataset contains a partial snapshot taken during 2006–2007 by Mislove et al. [2008]. In this social network the friendship connections (edges) are directed.

LiveJournal. LiveJournal is an online social network with a focus on journals and blogs. The dataset contains a partial snapshot of the nodes taken by Backstrom et al. [2006]. In this social network the friendship connections (edges) are directed.

For all figures, the y axis is the normalized Root Mean Squared Error (RMSE). Namely, $(\mathrm{E}[(\frac{\widehat{n}}{n}-1)^2])^{\frac{1}{2}}$. To estimate the RMSE, each simulation was run independently 10,000 times. In Section 6.2 we corroborate the theoretical convergence claims for Section 4.1 and Section 4.2. In Section 6.3 we compare prior-art node collision estimator [Hardiman et al. 2009; Katzir et al. 2011; Kurant et al. 2012] with the new generalized estimator.

6.2. Clustering Coefficients Coefficient

We run the simulations for the improved version of the clustering coefficient estimators described in Section 4.3. The estimators are labeled as "Random Walk CMC." The x axis in for these figures is the percentage of mined nodes (number of mined nodes over the total number of network nodes). Figures 1 and 2 display RMSE for $\widehat{\mathfrak{c}}_l$ and $\widehat{\mathfrak{c}}_g$ with CMC, respectively. The theoretical results show the error has a $\frac{\lceil \log 1/\epsilon \rceil}{\epsilon^2}$ term, which implies that the number of steps is roughly proportional to the error squared. This can be observed for all networks. Table III contains the RMSE for 1% mined nodes.

6.3. Network Size

We run the simulations for three estimators: (a) the "Node Collision" [Hardiman et al. 2009; Katzir et al. 2011] with the CMC enhancement, (b) the "Induced Edges" estimator [Kurant et al. 2012], and (c) the "Generalized Collision" described in this article. 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 ($|I| \approx 0.95r^2$).

Figure 3 shows three curves of the generalized estimator RMSE for 0.5%, 1.0%, and 2.0% mined nodes as a function of α. When $\alpha = 0$ the values coincide with the induced edges RMSE and when $\alpha = 1$ the values coincide with the node collision RMSE. One can see that the optimal value of α changes as a function of the number of mined nodes. Specifically, the optimal values of α are summarized in Table IV. In the general case, the error is a nonconvex function of α as can be seen from the LiveJournal graph. The

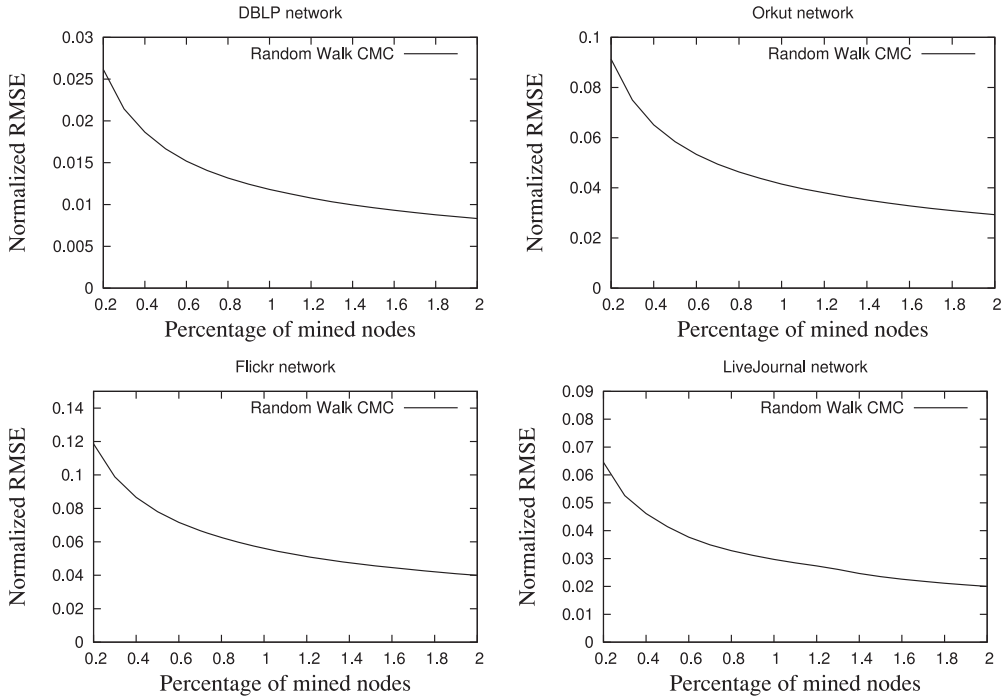

Fig. 1. Estimation of the network average clustering coefficient RMSE vs. the percentage of mined nodes.

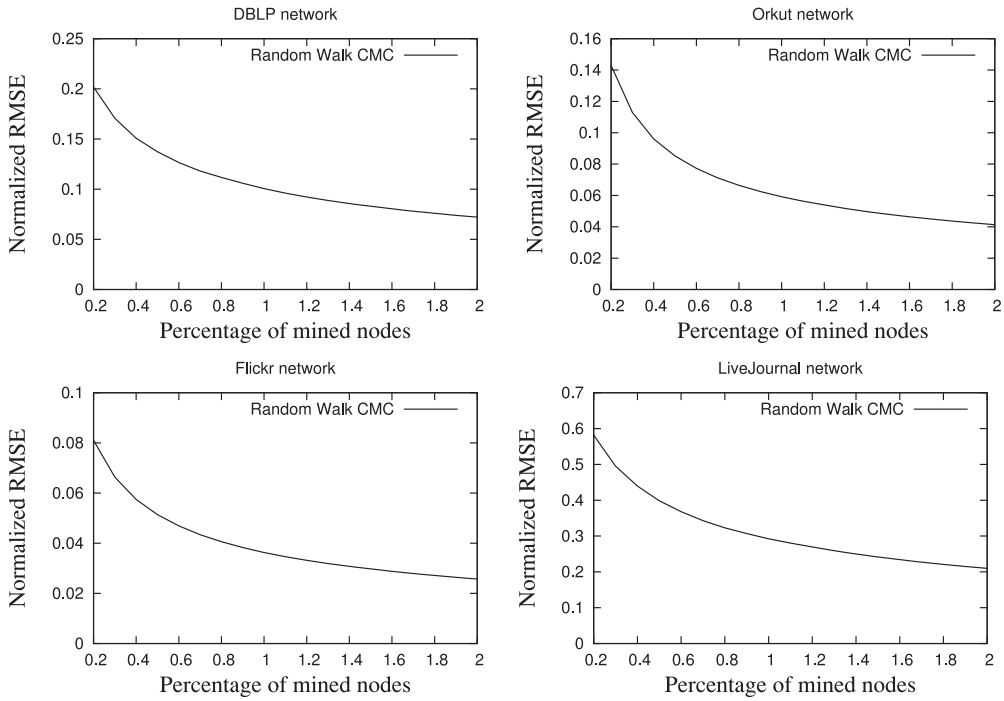

Fig. 2. Estimation of the global clustering coefficient RMSE vs. the percentage of mined nodes.

Table III. Clustering Coefficients Normalized RMSE for 1% Mined Nodes

Network	$\widehat{c}_l$ RMSE	$\widehat{c}_g$ RMSE
DBLP	0.0118	0.1006
Orkut	0.0415	0.0591
Flickr	0.0560	0.0363
LiveJ	0.0297	0.2924

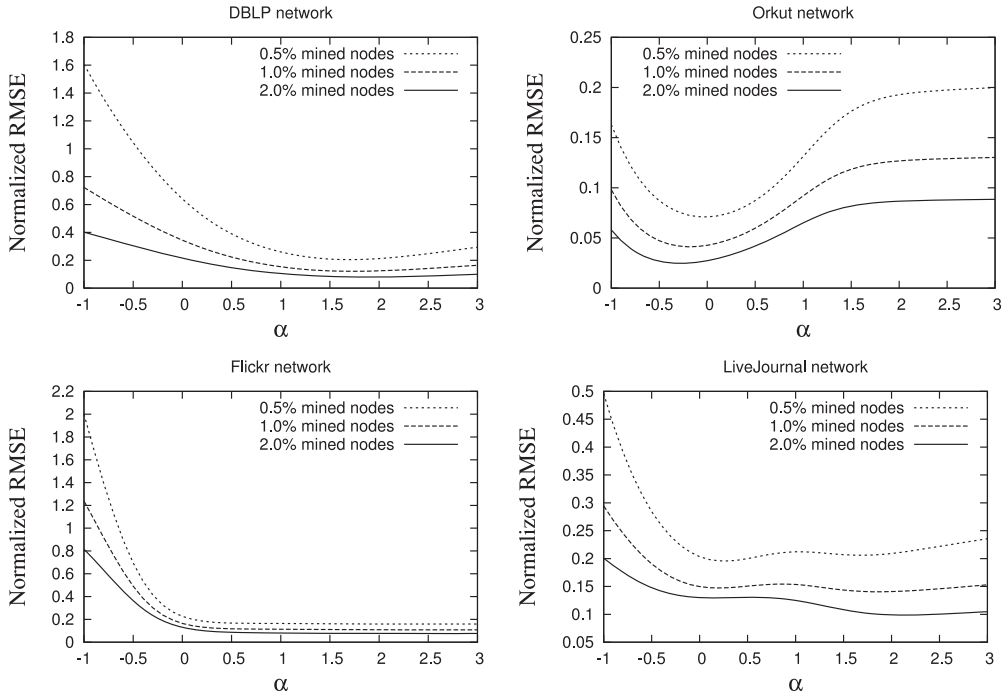

Fig. 3. The generalized collision RMSE vs. the α parameter.

Table IV. Optimal Values of Alpha

Network	0.5%	1.0%	2.0%
DBLP	1.7	1.7	1.9
Orkut	0.0	−0.2	−0.3
Flickr	2.2	2.7	2.7
LiveJ	0.3	1.9	2.1

main observation of these graphs is that a "good" choice of the weights can have a tremendous effect on the accuracy of the estimator.

In Figure 4 there are curves for the node collision, induced edges, and the generalized collision estimators. The x axis in these figures is the percentage of mined nodes (number of mined nodes over the total number of network nodes). The performance gaps can be explained by observing Figure 3 RMSE vs α functions.

DBLP. The RMSE resembles a quadratic function with a minimum around $\alpha = 2$. One can see that indeed the generalized collision outperforms the node collision, which greatly outperforms the induced edges estimator.

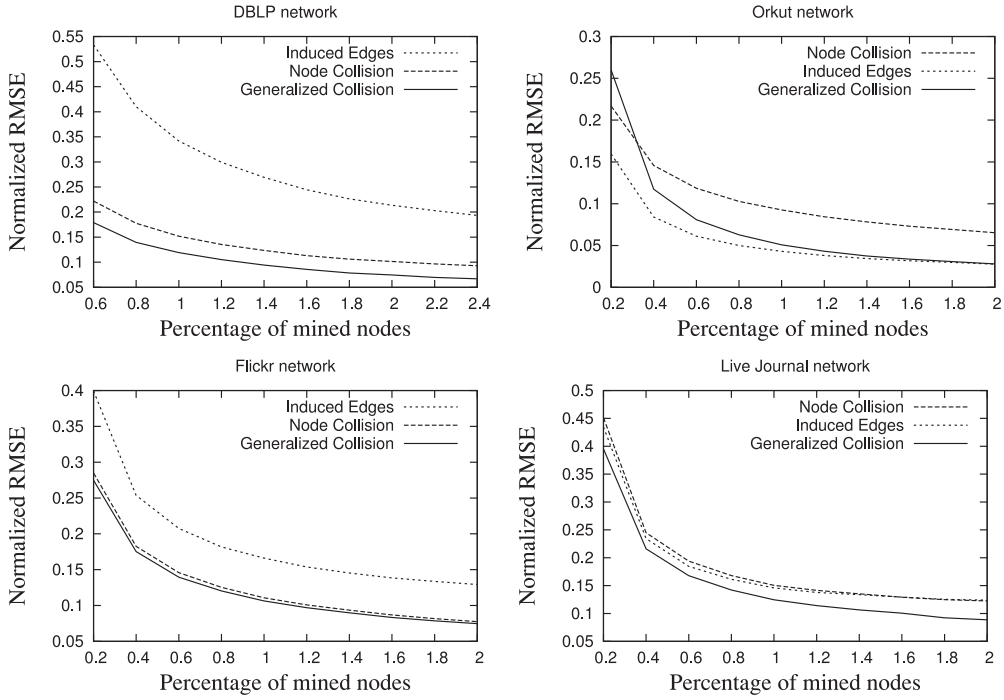

Fig. 4. Estimation of the network size RMSE vs. the percentage of mined nodes.

Orkut. The optimal value of α is 0 when there are fewer samples, but shifts to -0.3 as the number of samples increases. This fact makes the induced edges outperform the two other estimators. However, as the percentage of mined nodes increases, two factors are at work: (a) the optimal value of α decreases to -0.3 with increasing walk length; this has a negative effect on the accuracy of the induced edges estimator for walks of length 2% of mined nodes; and (b) the generalized collision algorithm estimates the optimal value of α better with longer walk lengths. Eventually, the generalized collision and the induced edges estimator have comparable performance.

Flickr. The RMSE resembles a constant line when $\alpha \geq 0.5$ with a quadratic shape when $\alpha < 0.5$. One can see that indeed the generalized collision has comparable performance to the node collision and both outperform the induced edges estimator.

LiveJournal. The RMSE is roughly the same for $\alpha = 0$, $\alpha = 1$, and around the optimal value of α. However, as the number of sampled nodes increases, the RMSE around $\alpha > 1$ gets lower. One can see that indeed all three estimators have comparable performance when the number of mined nodes is small, but as the number of samples increases the performance gap in favor of the generalized collision increases.

Using only 1% of the network size, the generalized collision estimator outperforms both algorithms for all except the Orkut network. The improvements versus the best out of the two estimators are 21%, -18%, 4.5%, and 15% for the DBLP, Orkut, Flickr, and LiveJournal, respectively. The average improvement versus the node collision estimator is 22% and versus the induced edges is 24%. Table V contains the RMSE for 1% mined nodes.

Table V. Size Estimation Normalized RMSE for 1% Mined Nodes

Network	Generalized	Induced edges	Node collision
DBLP	0.120	0.342	0.152
Orkut	0.051	0.043	0.093
Flickr	0.106	0.166	0.111
LiveJ	0.124	0.146	0.150

In all but one network the generalized collision estimator outperforms both prior-art estimators. Moreover, when comparing across networks the generalized collision algorithm obtains more than 20% improvement in accuracy.

In practice, one would seek an estimate of studied parameters within 10% of their true values. To this end, an RMSE of 0.05 would have a 95% guarantee, as all estimators are asymptotically normal. For the networks and parameters at hand, all but the global clustering coefficient for the LiveJournal Network are in the practical region.

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. For the clustering coefficient algorithms we showed an analytic bound on the number of steps required for convergence. For the number of registered users algorithm we showed that the new suggested generalized collision estimator is more accurate than prior-art estimators.

APPENDIX

A. CONCENTRATION OF Ψ_l AND Φ_l

In the proof of Lemma 4.2 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. [2012]:

THEOREM A.1 (THEOREM 3.1 [CHUNG ET AL. 2012]). *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 $[n]$, that is, $x_1 \leftarrow \varphi$. 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,

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

We will use the fact that $\tau_{mix}(\epsilon) \leq \lceil \log_2 1/\epsilon \rceil \tau_{mix}$ [Levin et al. 2008].

LEMMA A.2. *There is a constant value, ξ, such that if $r \geq r_{\Psi_l} = \xi \frac{D}{n} \log(\frac{1}{\delta}) \frac{\lceil \log 1/\epsilon \rceil}{\epsilon^2} \tau_{mix}$, 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, $\mathrm{E}[\Psi_l] = \mathrm{E}[\frac{1}{d_{x_k}}] = \frac{n}{D}$. From Theorem A.1,

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

Extracting r_{Ψ_l} for which $\frac{\delta}{2} = ce^{-\epsilon^2 nr/9\cdot 72\cdot \tau D}$, we have $r_{\Psi_l} \leq \tilde{\xi}\frac{1}{\epsilon^2}\frac{D}{n}\log(\frac{1}{\delta})\frac{\lceil \log 1/\epsilon\rceil}{\epsilon^2}\tau_{mix}$. □

LEMMA A.3. *There is a constant value, ξ, such that if $r \geq r_{\Phi_l} = \xi\frac{D}{nc_l}\log(\frac{1}{\delta})\frac{\lceil \log 1/\epsilon\rceil}{\epsilon^2}\tau_{mix}$, we have*

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

PROOF. For these bounds, we cannot apply Theorem A.1 directly since f_j depends on a previously visited node. However, since $\frac{A_{x_k\cdot x_{k+2}}}{d_{x_k}-1}$ only depends on a three-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 $f_k(\tilde{x}_k) = \frac{A_{x_{k-1}\cdot x_{k+1}}}{d_{x_k}-1}$. We assume that $\varphi \approx \pi$, and thus $\|\varphi\|_\pi = 1$. We have, $\mathrm{E}[\Phi_l] = \mathrm{E}[\phi_k\frac{1}{d_{x_k}-1}] = \frac{1}{D}\sum_{i=1}^{n} c_i = \frac{n}{D}c_l$. From Theorem A.1,

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

Extracting r_{Φ_l} for which $\frac{\delta}{2} = ce^{-\epsilon^2 nc_l(r-2)/9\cdot 72\cdot \tilde{\tau}(\epsilon)D}$, we have $r_{\Phi_l} \leq \tilde{\xi}\log(\frac{1}{\delta})\frac{\lceil \log 1/\epsilon\rceil}{\epsilon^2}\frac{D}{nc_l}\tilde{\tau}$.

Note that $\tilde{\tau}(\epsilon) \leq \tau(\epsilon) + 2$. 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, the probability of drawing x_{k-1}, x_k, x_{k+1} is $(\frac{d_{x_{k-1}}}{D} \pm \epsilon)\frac{1}{d_{x_k}}\frac{1}{d_{x_{k+1}}}$, 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 Lemmas A.2 and A.3, and choose $r_l = \max\{r_{\Psi_l}, r_{\Phi_l}\}$.

B. CONCENTRATION OF Ψ_G AND Φ_G

In the proof of Lemma 4.4, 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 B.1. *There is a constant value, ξ, such that if $r \geq r_{\Psi_g} = \xi\frac{Dd_{\max}}{\sum_{i=1}^{n} d_i(d_i-1)}\log(\frac{1}{\delta})\frac{\lceil \log 1/\epsilon\rceil}{\epsilon^2}\tau_{mix}$, 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, $\frac{1}{d_{\max}}\mathrm{E}[\Psi_g] = \mathrm{E}[\frac{d_{x_k}-1}{d_{\max}}] = \frac{1}{Dd_{\max}}\sum_{i=1}^{n} d_i(d_i-1)$. From Theorem A.1,

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

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 $r_{\Psi_g} \leq \tilde{\xi}\frac{Dd_{\max}}{\sum_{i=1}^{n} d_i(d_i-1)}\log(\frac{1}{\delta})\frac{\lceil \log 1/\epsilon\rceil}{\epsilon^2}\tau_{mix}$. □

LEMMA B.2. *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 B.1 and the the three-node history Markov chain $\tilde{M}$ of Lemma A.3. □

REFERENCES

Louigi Addario-Berry and Tao Lei. 2012. The mixing time of the Newman–Watts small world. In *SODA*. 1661–1668.

Yong-Yeol Ahn, Seungyeop Han, Haewoon Kwak, Sue B. Moon, and Hawoong Jeong. 2007. Analysis of topological characteristics of huge online social networking services. In *WWW*. 835–844.

Noga Alon, Raphael Yuster, and Uri Zwick. 1997. Finding and counting given length cycles. *Algorithmica* 17, 3 (1997), 209–223.

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

Lars Backstrom, Daniel P. Huttenlocher, Jon M. Kleinberg, and Xiangyang Lan. 2006. Group formation in large social networks: Membership, growth, and evolution. In *KDD*. 44–54.

Ziv Bar-Yossef and Maxim Gurevich. 2008. Random sampling from a search engine's index. *J. ACM* 55, 5, Article 24 (Oct. 2008).

Ziv Bar-Yossef and Maxim Gurevich. 2009. Estimating the impressionrank of web pages. In *WWW*. 41–50.

Ziv Bar-Yossef and Maxim Gurevich. 2011. Efficient search engine measurements. *TWEB* 5, 4, Article 18 (Oct 2011).

Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. 2010. Efficient algorithms for large-scale local triangle counting. *TKDD* 4, 3, Article 13 (Oct. 2010).

Luciana S. Buriol, Gereon Frahling, Stefano Leonardi, Alberto Marchetti-Spaccamela, and Christian Sohler. 2006. Counting triangles in data streams. In *PODS*. 253–262.

Kai-Min Chung, Henry Lam, Zhenming Liu, and Michael Mitzenmacher. 2012. Chernoff-Hoeffding bounds for Markov chains: Generalized and simplified. In *STACS*. 124–135.

Luciano da F. Costa, Francisco A. Rodrigues, Gonzalo Travieso, and Paulino R. Villas Boas. 2006. Characterization of complex networks: A survey of measurements. *Adv. Phys.* 56, 1 (Aug. 2006), 167–242. http://dx.doi.org/10.1080/00018730601170527

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Minas Gjoka, Maciej Kurant, Carter T. Butts, and Athina Markopoulou. 2010. Walking in Facebook: A case study of unbiased sampling of OSNs. *Proceedings of IEEE INFOCOM 2010*, 1–9.

Minas Gjoka, Maciej Kurant, and Athina Markopoulou. 2013. 2.5K-Graphs: From sampling to generation. In *Proceedings of IEEE INFOCOM'13*.

Stephen James Hardiman, Peter Richmond, and Stefan Hutzler. 2009. Calculating statistics of complex networks through random walks with an application to the on-line social network Bebo. *Eur. Phys. J. B* 71, 4 (2009), 611–622.

Wolfgang Härdle, Joel Horowitz, and Jens Peter Kreiss. 2003. Bootstrap methods for time series. *Int. Stat. Rev.* 71, 2 (Aug. 2003), 435–459.

Liran Katzir, Edo Liberty, and Oren Somekh. 2011. Estimating sizes of social networks via biased sampling. In *WWW*. 597–606.

Jerome Kunegis. 2012. KONECT—The Koblenz Network Collection. http://konect.uni-koblenz.de/.

Hans R. Künsch. 1989. The jackknife and the bootstrap for general stationary observations. *Ann. Stat.* 17, 1217–1241.

Maciej Kurant, Carter T. Butts, and Athina Markopoulou. 2012. Graph size estimation. *CoRR* abs/1210.0460.

David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. 2008. *Markov Chains and Mixing Times*. American Mathematical Society.

Michael Ley. 2002. The DBLP computer science bibliography: Evolution, research issues, perspectives. In *Proceedings of the International Symposium on String Processing and Information Retrieval*. 1–10.

László Lovász and Peter Winkler. 1998. Mixing times. Microsurveys in discrete. In *DimacsWorkshop*.

Laurent Massoulié, Erwan Le Merrer, Anne-Marie Kermarrec, and Ayalvadi Ganesh. 2006. Peer counting and sampling in overlay networks: Random walk methods. In *Proceedings of the 25th Annual ACM Symposium on Principles of Distributed Computing (PODC'06)*. ACM, New York, NY, 123–132. DOI:http://dx.doi.org/10.1145/1146381.1146402

Alan Mislove, Hema Swetha Koppula, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2008. Growth of the flickr social network. In *Proceedings of the 1st ACM SIGCOMM Workshop on Social Networks (WOSN'08)*.

Alan Mislove, Massimiliano Marcon, P. Krishna Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Internet Measurement Comference*. 29–42.

Abedelaziz Mohaisen, Aaram Yun, and Yongdae Kim. 2010. Measuring the mixing time of social graphs. In *Internet Measurement Conference*. 383–389.

Mark E. J. Newman and Duncan J. Watts. 1999a. Renormalization group analysis of the small-world network model. *Phys. Lett. A* 263, 341–346.

Mark E. J. Newman and Duncan J. Watts. 1999b. Scaling and percolation in the small-world network model. *Phys. Rev. E* 60, 7332–7342.

Bruno F. Ribeiro and Donald F. Towsley. 2010. Estimating and sampling graphs with multidimensional random walks. In *Internet Measurement Conference*. 390–403.

Reuven Y. Rubinstein and Dirk P. Kroese. 2007. *Simulation and the Monte Carlo Method* (2nd. ed.). Wiley Series in Probability and Statistics.

Thomas Schank and Dorothea Wagner. 2005. Approximating clustering coefficient and transitivity. *J. Graph Algorithms Appl.* 9, 2 (2005), 265–275.

Pinghui Wang, John C. S. Lui, Bruno F. Ribeiro, Don Towsley, Junzhou Zhao, and Xiaohong Guan. 2014. Efficiently estimating motif statistics of large networks. *TKDD* 9, 2, Article 8 (Nov. 2014). DOI:http://dx.doi.org/10.1145/2629564

Shaozhi Ye and Felix Wu. 2010. Estimating the size of online social networks. In *2010 IEEE 2nd International Conference on Social Computing (SocialCom)*. 169–176. DOI:http://dx.doi.org/10.1109/SocialCom.2010.32

Received November 2014; revised April 2015; accepted June 2015