Internet Mathematics, 12:221–238, 2016
Copyright © Taylor & Francis Group, LLC
ISSN: 1542-7951 print/1944-9488 online
DOI: 10.1080/15427951.2016.1164100

FAST LOW-COST ESTIMATION OF NETWORK PROPERTIES USING RANDOM WALKS

Colin Cooper, Tomasz Radzik, and Yiannis Siantos
Department of Informatics, King's College London, London, UK

Abstract *We study the use of random walks as an efficient method to estimate global properties of large connected undirected graphs. Typical examples of the properties of interest include the number of edges, vertices, and triangles, and more generally, the number of small fixed subgraphs. We consider two methods based on first returns of random walks: (1) the cycle formula of regenerative processes and (2) weighted random walks with edge weights defined by the property under investigation. We review the theoretical foundations for these methods and indicate how they can be adapted for the general nonintrusive investigation of large online networks.*
The expected value and variance of the time of the first return of a random walk decrease with increasing vertex weight, so for a given time budget, returns to high-weight vertices should give the best property estimates. We present theoretical and experimental results on the rate of convergence of the estimates as a function of the number of returns of a random walk to a given start vertex. We made experiments to estimate the number of vertices, edges, and triangles for two test graphs.

1. INTRODUCTION

Recent developments in technology have allowed the creation of large networks available globally via personal computers or, more recently, mobile phones. The original and most outstanding examples of such networks are the WWW and the email network. Relatively recently, many On-Line Social Networks (OLSN) such as Twitter and Facebook, or online video repositories such as YouTube have sprung up. The size, structure, and rate of growth of these networks is a question of natural interest. Because they are so large and access to them is often limited by the provider, we need methods to investigate them that are fast relative to the network size, are nonintrusive, and have low storage overheads.

We investigate how effective random-walk-based sampling methods are for estimating structural properties of a connected undirected graph (a model of a large network), such as the number of vertices, edges, and small subgraphs. We collect existing theoretical facts that are useful in designing random-walk-based methods, and evaluate the performance of these methods experimentally. The final application is practical, but we are guided by theory as far as possible. The methods we consider are based on randomized crawling by downloading pages from the network under investigation. Our assumption is that the network cannot be explored systematically by breadth-first search (BFS), either because of size or because the number of third-party accesses to the network is restricted.

A preliminary version of this article appeared in the Proceedings of WAW 2013 [8].

Address correspondence to Tomasz Radzik, Department of Informatics, King's College London, Strand, London WC2R 2LS, UK. E-mail: tomasz.radzik@kcl.ac.uk

Let $G = (V, E)$ be a connected undirected graph with $|V| = n$ vertices and $|E| = m$ edges. We consider a reversible random walk on G. We assume that the random walk on G is ergodic, i.e., the graph is connected and the walk is aperiodic. If the walk is periodic, it can always be made aperiodic by making it lazy (looping at the current vertex with probability 1/2). A walk is reversible if the transition probability along an edge (u, v) is proportional to a positive weight $w(u, v)$. If we take $w(u, v) = 1$, then this is a simple random walk. The expected first-return time T_u^+ of a random walk to a vertex u is given by $\mathbf{E}T_u^+ = 1/\pi_u$, where π_u is the stationary probability of vertex u. For a simple random walk, $\mathbf{E}T_u^+ = 2m/d(u)$, where $d(u)$ is the degree of u. If $d(u)$ is large, then $\mathbf{E}T_u^+$ is small and we can quickly obtain an estimate for m. For example, a graph generated by preferential attachment has $m = cn$ edges and vertices u with degree as large as $d(u) = \sqrt{n}$. We can use a walk starting from such a vertex to estimate m in $2m/d(u) = O(\sqrt{n})$ expected steps.

An important idea is that for graphs in which there is variation in degree sequence, it is possible to use a simple random walk to quickly and accurately estimate the number of edges based on first returns to high-degree vertices. If the graph is near regular, $\mathbf{E}T_u^+ = \Theta(n)$ for any start vertex u. This is not helpful if we want a quick answer. Such graphs may still exhibit variations in local structure, which we can exploit. For example, the number of triangles at a vertex may vary considerably. If so, we could use a random walk with vertex weight proportional to the number of triangles at the vertex. By starting from a high-weight vertex, we should be able to exploit this structural variation to count properties efficiently.

We discuss the following ideas, both theoretically and experimentally.

1. Global properties of graphs can be estimated using the times of the first returns of random walks. The general theory is given in Section 3 with respect to the cycle formula of regenerative processes and weighted random walks. For a given property, these approaches either keep a running total of the number of structures (e.g., triangles) observed by each excursion of the simple (uniform) random walk (the cycle-formula method), or use first-return times of random walks with edge weights proportional to the number of structures containing the edge (the weighted-random-walk method).
2. The use of the cycle formula of regenerative processes is discussed in Section 3.1. The use of weighted random walks is developed in Section 3.2 with respect to various examples such as the number of triangles, vertices, and arbitrary fixed subgraphs.
3. The quality of the methods depends on the distribution of first-return times to the start vertex. We review the theory relating to this in Section 3.3. Vertices with high degree (or more generally, with high vertex weight) have smaller expected value and (upper bound on) variance of return time, and should estimate properties more effectively. This is also discussed in Section 3.3.
4. Experimentally, as the walk proceeds, it naturally discovers high-weight vertices, and the estimates based on returns to these vertices are efficient after a reasonable number of steps. The performance of random walks on suitable test graphs is assessed in Section 4. We note that the article [2] gives techniques to directly locate high-degree vertices, which could be used in conjunction with our work.
5. The expected value and variance of the first-return time T_v^+ of a random walk to a given vertex v are known quantities, given by $\mathbf{E}T_v^+ = 1/\pi_v$ and $\mathbf{Var}\ T_v^+ =$

$(2Z_{vv} + \pi_v - 1)/\pi_v^2$, respectively, where $Z_{vv} = \sum_{t \geq 0}(P_v(v,t) - \pi_v)$ and $P_v(v,t)$ is the probability that a walk starting from v returns to v at step t. It is difficult to evaluate Z_{vv} directly, but we can bound Z_{vv}, and hence, the variance of our estimates, using the eigenvalue gap of the transition matrix. The variance of our estimates can also be estimated directly from the return time data using a result of [6]. See Section 3.3 for details.

The aim of this study is to collect available information on random-walk-based methods for estimating network properties and to compare and develop the techniques. Our original contributions are in the design of weighted random walks to estimate, e.g., number of vertices, triangles, and fixed motifs, and to detect clustering (see Section 3.2). We also provide theoretical and experimental methods to bound and estimate the variance of the first return time T_v^+ to the start vertex v (3.13); see methods M1 and M2 in Section 3.3. Our initial results on estimating graph properties by first-return times of weighted random walks were presented in [7]. Here, we investigate this method further by comparing it with the method based on the cycle formula of regenerative processes and by considering the variance of the obtained estimates.

The complexity measures we use to present our results are somewhat crude, because the processing load per walk step varies both locally and on the remote site for the different walks we use. Our basic measures are the number of steps and the number of returns to the start vertex. By choosing a high-weight start vertex, the expected first-return time can be made sublinear (see ideas 3, 4). The variance of this quantity can also be bounded as outlined.

2. NETWORK SAMPLING METHODS BASED ON RANDOM WALKS

The simplest way to study a network is to inspect it completely using, e.g., breadth first search. Failing this, a simple statistical method of sampling vertices uniformly at random (u.a.r) can be considered. In practice, for large networks such as the WWW or OLSNS, neither of these methods is feasible, but the network can still be queried by some limited form of crawling or interaction with the network through its API. Our assumption is that query results are held locally on a single processor. The selection of the next vertex to visit (query) is based on a random walk run on the query data. The random walk is used as a randomized algorithm to determine the next step in the query process. We measure the computational complexity as the number of steps made by the walk, and our aim is to obtain results in a number of steps sublinear in the network size, which is assumed unknown. There are various ways to study network properties using a random walk, including:

1. using a random walk as a surrogate for uniform sampling (an outline of this is given in Section 2.1),
2. estimates based on running totals sampled by the walk (an outline of this is given in Section 2.2), and
3. estimates based on the times of the first returns of random walks (this is discussed in detail in Section 3).

We suppose that as we crawl a graph, we record the value of some function $f(X_i)$ of the vertices visited by a random walk $(X_i : i = 1, \ldots, t)$, and keep the running total $S(t)$ of the sampled values. Thus,

$$S(t) = \sum_{i=1}^{t} f(X_i). \tag{2.1}$$

Here, X_i is the vertex position of the walk at step i, and $f(X_i)$ is a function evaluated at this vertex. For example, the value of $f(v)$ at vertex v could be an indicator function for a vertex of degree at least d, ($f(v) = 1$, iff $d(v) \geq d$); or an indicator function that is set to $f(u) = 1$ whenever the walk returns to its start vertex u (i.e., $f(v) = 0,\ v \neq u$).

For a walk in the stationary distribution π, $\mathbf{E}_\pi f(X) = \sum_{v \in V} \pi_v f(v)$. In order to estimate the average value $\overline{f} = (1/|V|) \sum_{v \in V} f(v)$, it would be necessary to replace $f(v)$ by $g(v) = f(v)/\pi_v$ in (2.1). For a simple random walk $\pi_v = d(v)/2m$, which supposes we know the number of edges m or can estimate it from, e.g., the expected first-return time $\mathbf{E}T_v^+ = 1/\pi_v$. An alternative method that avoids estimating the number of edges m is to use a renormalization such as respondent driven sampling (see [12]).

2.1. Using a Random Walk as a Surrogate for Uniform Sampling

Sampling the elements of a set uniformly at random with replacement can be used to estimate the set size in sublinear time by the method of *sample and collide*. The use of this method to estimate network size is described in [4]. If we sample uniformly at random with replacement from a population of size n, then, by the birthday paradox, the expected number of trials required for the first repetition is $\sqrt{2n}$ [10]. The expected number of repetitions in s samples is $s(s-1)/2n$. Thus, if the first repetition occurs at sample R, then $\overline{n} = R^2/2$ is an estimate of the network size.

The method of sample and collide requires u.a.r. samples from the population. To obtain a sample from a network using a random walk, we can do the following. Run the walk for $t \geq T$ steps before sampling, where T is a suitable mixing time. In this case, the walk is in near-stationarity and $P_u(X_t = x) \sim \pi_x$, where X_t is the position of the walk at step t, and π_x is the stationary distribution of the walk. However, for a simple random walk $\pi_x = d(x)/2m$, where $d(x)$ is degree of x and m is the number of edges. Thus, unless the graph is regular, the sample is not uniform.

To use a sample from the stationary distribution, we need to unbias the walk. There are several ways to do this, for example:

(1) One method is to use the approach of [19] and [11], who use a continuous-time random walk; random waiting time at a vertex x, which is negative exponential with mean $1/d(x)$; and a fixed stopping time T. In this way, the obtained stationary distribution is uniform.
(2) The discrete equivalent (reweighted random walk) is to walk for a fixed number of steps T and then sample the vertex x occupied by the walk. To unbias the sample, retain the sample with probability $1/d(x)$. This gives a uniform sampling probability of $1/2m$. If $d(x)$ is large, this gives a slow sampling rate.
(3) Another method is to use a Metropolis–Hastings random walk with target stationary distribution $\pi_x = 1/n$. One way to do this is to use a transition probability $1/M$ where $M \geq \Delta(G)$, the maximum degree of G. This effectively converts G into an M-regular multigraph. Thus, for $y \in N(x)$, the transition probability is $p(x, y) = 1/M$, and $p(x, x) = 1 - d(x)/M$. It follows from detailed balance $\pi_x p(x, y) =$

$\pi_y p(y, x)$ that $\pi_x = 1/n$; the stationary distribution is uniform. The problem with this approach, is that it slows the walk by a factor of $\Theta(M/\delta)$, where δ is the minimum degree.

An alternative approach to uniform sampling is developed by [13]. A simple random walk is used in conjunction with the birthday paradox, and the statistical bias arising from the nonuniform stationary distribution is approximately corrected.

2.2. Estimates Based on Running Totals Sampled by the Walk

The following section concerns the empirical accuracy of the sampled running total $S(t)$ as given by (2.1). The total $S(t) = \sum_{i=1}^{t} f(X_i)$ depends on a function $f(X_i)$ evaluated at the vertices X_i visited by a random walk during steps $i = 1, \dots, t$. Theorems 2.1–2.3 provide Chernoff-type bounds for the concentration of the random variable $S(t)$. Theorem 2.1 seems the most straightforward in application but requires the walk to start from the stationary distribution. The bounds in Theorems 2.2 and 2.3 allow any initial distribution of the random walk.

The following result from [14] concerns sampling a function $f(v) : v \in V$ on a graph $G = (V, E)$ using a reversible ergodic random walk starting from stationarity. The function $f(v)$ evaluated at any vertex v is restricted to take values in the interval [0, 1]. Let π be the stationary distribution of the walk, and $\mu = \mathbf{E}_\pi f(X) = \sum_{v \in V} \pi_v f(v)$ the expected value of $f(X)$ w.r.t the stationary distribution. Let $\lambda_0 = \max(\lambda, 0)$, where λ is the second largest eigenvalue of the transition matrix P of the random walk. Let $\Pr_\pi$ denote the probability of an estimate for a walk starting from stationarity. Then, the following theorem holds.

Theorem 2.1. [14]. *Consider an ergodic and reversible Markov chain $(X_0, X_1, \dots)$ on a graph $G = (V, E)$, starting from the stationary distribution π. Let $S(t) = \sum_{i=1}^{t} f(X_i)$ be the running total. Then, for any $\epsilon > 0$ such that $\mu + \epsilon < 1$,*

$$\Pr_\pi(|S(t) - \mu t| \geq \epsilon t) \leq 2 \exp\left(-2 \frac{1 - \lambda_0}{1 + \lambda_0} t \epsilon^2\right). \tag{2.2}$$

An alternative approach used by [9], and [17] is to estimate such probabilities starting from a given initial distribution $q = (\Pr(X_0 = v : v \in V)$ on vertices. Theorems 2.2 and 2.3 assume that the function f is mean centered, i.e., $\mu = \mathbf{E}_\pi f = 0$.

The following Chernoff-type bound for Markov chains is from [17].

Theorem 2.2. [17]. *Consider an ergodic and reversible Markov chain $(X_0, X_1, \dots)$ on a graph $G = (V, E)$. Let $S(t) = \sum f(X_i)$ be the running total. Assume $\max_v |f(v)| \leq 1$. Let λ be the second largest eigenvalue of the transition matrix P of the walk. Then, for any initial distribution q of X_0, any positive integer t and all $0 \leq \gamma \leq 2/5$,*

$$\Pr_q(S(t) \geq t\gamma) \leq e^{(1-\lambda)/5} \cdot \sum_{u \in V} \frac{(\Pr(X_0 = u))^2}{\pi(u)} \cdot \exp\left(-\frac{t\gamma^2(1-\lambda)}{4}\left(1 - \frac{5\gamma}{2}\right)\right).$$

A recent study [20] gives simplified bounds for Lezaud-type inequalities given the variance $\sigma^2 = \sum_{v \in V} \pi_v f^2(v)$ of f. An example of this is given by Theorem 2.3.

Theorem 2.3. [20]. *Consider an ergodic and reversible Markov chain $(X_0, X_1, \dots)$ on a graph $G = (V, E)$. Let $S(t) = \sum f(X_i)$ be the running total. Let $\lambda_0 = \max(\lambda, 0)$, where λ is the second eigenvalue of the transition matrix P of the walk. Assume $\max_v |f(v)| \leq 1$. Then, for any initial distribution q of X_0, any positive integer t, and all $\gamma > 0$,*

$$\Pr_q(S(t) \geq t\gamma) \leq e^{\frac{1-\lambda_0}{1+\lambda_0}\frac{\gamma^2}{\sigma^2+\gamma}} \cdot \sum_{u \in V} \frac{(\Pr(X_0 = u))^2}{\pi(u)} \cdot \exp\left(-\frac{t}{8\sigma^2}(1-\lambda_0)\gamma^2\right).$$

3. ESTIMATES BASED ON FIRST-RETURN TIME OF A RANDOM WALK

For a random walk starting at vertex v, the first-return time to v is defined as

$$T_v^+ = \min\{t > 0 : X_t = v\},$$

where X_t is the position of the walk at step t ($X_0 = v$). If the walk is ergodic, it has a well-defined stationary distribution π_v at any vertex v, and the expected value of the first-return time $\mathbf{E}T_v^+$ is given by $\mathbf{E}T_v^+ = 1/\pi_v$.

We describe two methods to estimate properties of networks based on first-return times of random walks. The methods are in no sense mutually exclusive and can indeed be used together. The first method, the cycle formula of regenerative processes, has typically been used with simple random walks (e.g., [19]). The second method uses first-return times of weighted random walks. Both methods are equally viable for estimating a given property.

An important point for either method is that high-weight vertices perform well as start vertices for random walk property estimates. For a simple random walk, the weight of vertex u is the vertex degree $d(u)$, and this feeds into the first return time $\mathbf{E}T_u^+ = 2m/d(u)$. Thus, in regular graphs, all vertices are equivalent start points. By reweighting the walk, we can artificially create high-weight vertices suitable for estimating a given property.

To give an example of this, consider a regular graph that contains many triangles (copies of K_3), distributed in nonuniform clusters. In a simple random walk, because vertex weight is proportional to degree, this graph has no high-weight vertices. By weighting edges proportional to the number of triangles they are contained in, vertices with many triangles assume a high weight. First returns to these vertices can provide a good estimate for the total number of triangles.

For an (ergodic) weighted random walk, the expected value of the time of the first return is equal to the reciprocal of the stationary probability, as in the case of the simple unweighted walk:

$$\mathbf{E}T_v^+ = \frac{1}{\pi_v}, \tag{3.1}$$

but the stationary probability now is $\pi_v = w(v)/w_G$, where $w(v)$ is the weight of vertex v and w_G is the total weight of the graph.

3.1. Estimates Based on the Cycle Formula of Regenerative Processes

The cycle formula of regenerative processes can be summarized as follows. Consider a random walk starting from vertex u and let $f(X_t)$ be a vertex valued function. Then,

$$\mathbf{E}_u\left(\sum_{t=0}^{T_u^+-1} f(X_t)\right) = \mathbf{E}T_u^+ \sum_{v\in V} \pi_v f(v). \tag{3.2}$$

This identity is a consequence of the result (see, e.g., [1] Chapter 2, Lemma 6) that

$$\mathbf{E}_u(\text{number of visits to } v \text{ before time } T_u^+) = \frac{\pi_v}{\pi_u} = \mathbf{E}T_u^+\ \pi_v.$$

Identity (3.2) was used [19] to count network size using a simple random walk. Putting $f(v) = 1/d(v)$ removes the degree bias from π_v so that $\sum_{v\in V} \pi_v f(v) = n/2m$, and the right-hand side of (3.2) equals $n/d(u)$.

Let $\overline{\phi} = \sum_{v\in V} \phi(v)$ be the quantity that we want to estimate. Following [19], we put $f(v) = \phi(v)/w(v)$ when generalizing to weighted random walks, with $\pi_v = w(v)/w_G$. Denote by R_u the random variable $\sum_{t=0}^{T_u^+-1} f(X_t)$, with expectation $\mathbf{E}R_u$ given by (3.2). Then, as $\mathbf{E}T_u^+ = 1/\pi_u$,

$$\mathbf{E}R_u = \mathbf{E}T_u^+ \times \sum_{v\in V} \pi_v \frac{\phi_v}{w(v)} = \frac{\overline{\phi}}{w(u)}. \tag{3.3}$$

An important point experimentally is that $\overline{\phi}$ obtained from (3.3) does not depend on the total weight w_G, but only on $w(u)$, which is a known quantity.

3.2. Estimates Based on Return Times of Weighted Random Walks

This technique generalizes the following observation. For a simple random walk, the stationary distribution of vertex u is $\pi_u = d(u)/2m = 1/\mathbf{E}T_u^+$. Thus, the first-return time T_u^+ can be used to estimate the number of edges m of a graph. Let $Z(k) = \sum_{i=1}^{k} Z_i$ be the time of the kth return to vertex u. The random variable

$$\widehat{m} = \frac{Z(k)d(u)}{2k} \tag{3.4}$$

estimates the total number of edges m.

The basic idea is to design the stationary distribution to reveal the required network property. To do this we fix the edge weights for the walk transitions at any vertex in such a way that the required answer is contained in the graph weight w_G. The total weight w_G can be obtained from the stationary distribution $\pi_v = w(v)/w_G$ of the start vertex v, which by (3.1) is the reciprocal of the expected return time. The larger $w(v)$, the smaller $\mathbf{E}_v T_v^+$, giving us more rapidly k samples for (3.4).

The remainder of this section is arranged as follows. First, we summarize the properties of weighted random walks. Second, we give examples of using weighted random walks to estimate the total number of triangles t in the network (a litmus test for social networks) and to estimate the size n of the network. Third, we explain the general framework for estimating the number of small fixed subgraphs ("motifs" such as triangles, cliques, cycles, etc.), and for detecting clustering within a given set of vertices S.

3.2.1. Weighted random walks. Given a graph $G = (V, E)$ with m edges and a positive weight function $w(u, v)$ on edges $\{u, v\} \in E$, we can define a Markov chain with state space $S = V$ and a transition matrix with elements,

$$p_{uv} = \begin{cases} \dfrac{w(u, v)}{w(u)}, & \text{if } \{u, v\} \in E, \\ 0, & \text{otherwise,} \end{cases}$$

where $w(u) = \sum_{\{u,v\}\in E} w(u, v)$ is the weight of a vertex u, and $w_G = \sum_{u\in V} w(u) = 2\sum_{\{u,v\}\in E} w(u, v)$, is the weight of the graph G. See [1] for details.

We refer to this chain as a weighted random walk on G. The stationary distribution is

$$\pi_u = \frac{w(u)}{w_G}. \tag{3.5}$$

A special, but important case of a weighted random walk is the simple random walk, where $w(u, v) = 1$ for all $\{u, v\} \in E$. For this case,

$$p_{uv} = \begin{cases} \dfrac{1}{d(u)}, & \{u, v\} \in E, \\ 0, & \text{otherwise,} \end{cases}$$

$$\pi_u = \frac{d(u)}{2m}.$$

3.2.2. Estimating the number of triangles. For each edge e, we assign the weight $1 + t(e)$, where $t(e)$ is the number of triangles containing e. Let $t(v)$ be the number of triangles containing v and $t(G)$ the total number of triangles in G. Then,

$$\pi_u = \frac{w(u)}{w_G} = \frac{d(u) + 2t(u)}{2m + 6t(G)},$$

and

$$\begin{aligned} t(G) &= \frac{d(u) + 2t(u)}{6\pi_u} - \frac{m}{3} \\ &= \mathbf{E}_v T_v^+ \frac{d(u) + 2t(u)}{6} - \frac{m}{3}. \end{aligned}$$

Let $Z(k) = \sum_{i=1}^{k} Z_i$ be the time of the kth return to vertex u. We estimate the number of triangles $t(G)$ by

$$\hat{t} = \max\left\{0, \frac{Z(k)(d(u) + 2t(u))}{6k} - \frac{m}{3}\right\}, \tag{3.6}$$

where m can be estimated by (3.4).

3.2.3. Estimating the network size. We now use inversely degree-biased weighted random walks, setting the edge weight

$$w(u, v) = \frac{1}{d(u)} + \frac{1}{d(v)}.$$

It can be shown that $w_G = 2n$, so the stationary distribution is

$$\pi_u = \frac{w(u)}{w_G} = \frac{1 + \sum_{v \in N(u)} \frac{1}{d(v)}}{2n}. \tag{3.7}$$

Let $Z(k) = \sum_{i=1}^{k} Z_i$ be the time of the kth return to vertex u, as before, and let $w(u)$ be as shown in (3.7). We use the following estimate for the number of vertices:

$$\widehat{n} = \frac{Z(k)w(u)}{2k}. \tag{3.8}$$

3.2.4. Estimating the number of occurrences of an arbitrary fixed subgraph. Using a weighted random walk to estimate the number of edges $m(G)$ or triangles $t(G)$ in a graph G are special cases of the following problem. Let $\mathcal{S}$ be a set of unlabeled graphs. For each $M \in \mathcal{S}$, we want to count the number of distinct labeled copies of M in the graph G. The cases edges and triangles given above correspond to $\mathcal{S} = \{K_2\}$ and $\mathcal{S} = \{K_2, K_3\}$, respectively. For each $e \in E(G)$, we put $w(e) = \sum_{M \in \mathcal{S}} N(M, e)$, where $N(M, e)$ is the number of distinct subgraphs H isomorphic to M, which contain e. The simplest case (after $\mathcal{S} = \{K_2\}$) is $\mathcal{S} = \{K_2, M\}$, where M can be any connected subgraph, e.g., K_k, $K_{k,\ell}$, a chordless cycle of length 4, or some specific (small) tree. In this case, we have the following:

$$w_G = 2\sum_e w(e) = 2\sum_e (1 + N(M, e)) = 2m + 2\nu\mu(G), \tag{3.9}$$

where $\nu = |E(M)|$ and $\mu(G)$ is the number of distinct copies of M in G. As $\pi_v = w(v)/w_G$, and $w(v)$ and ν are known, we can use the method of first returns to estimate $\mu(G)$.

As an experimental heuristic, we can use weighted walks with edge weight

$$w(e) = 1 + cN(M, e). \tag{3.10}$$

We have $c = 1$ in (3.9), but any value of $c > 0$ is valid. The parameter c can be chosen smaller than 1 (e.g., $c = 1/10$) in order to stop large values of $N(M, e)$ distorting the eigenvalue gap and, hence, the mixing rate of the walk. We adopted this approach with some success for counting triangles in the Google web graph (see Section 4).

3.2.5. Detecting edges within a given set *S*. For a subset of vertices $S \subseteq V$, let $d(S) = \sum_{v \in S} d(v)$ and $d_S(S) = \sum_{v \in S} d_S(v)$ be the total degree and the "internal" degree of S, respectively. The ratio of $d_S(S)$ to $d(S)$ can be considered as a measure of evidence for clustering.

Let $S \subseteq V$ be given. The values $d_S(S)$ and $d(S)$ can be estimated using appropriate weighted random walks. We use the following edge weights for estimating $d(S)$, where $c > 0$ is a constant. For edge $e = \{u, v\}$, let $w(e) = 1$ if neither vertex is in S, let $w(e) = 1 + c$ if exactly one vertex is in S, and let $w(e) = 1 + 2c$ if both vertices are in S. It follows that $w_G = 2m + 2cd(S)$. For estimating $d_S(S)$, use the weight $1 + c$ for all edges with both ends in S and the weight 1 otherwise.

Using these weighted random walks for estimating $d_S(S)$ and $d_V(S)$ can be viewed as a special case of estimating the number of occurrences of a fixed *colored* subgraph. For example, let all vertices in S be colored red and all vertices in $V \setminus S$ colored white. The number of occurrences in G of K_2 with both ends colored red is equal to $d_S(S)/2$.

3.3. Distributional Properties of First-Return Times

As stated earlier, the expected value of the first-return time T_v^+ to a vertex v is $\mathbf{E}T_v^+ = 1/\pi_v$; see, e.g., [1]. The variance of T_v^+ is given by

$$\mathbf{Var}\ T_v^+ = \frac{2\mathbf{E}_\pi T_v + 1}{\pi_v} - \frac{1}{\pi_v^2}. \tag{3.11}$$

Here, $\mathbf{E}_\pi T_v$ is the expected hitting time of v from stationarity, i.e.,

$$\mathbf{E}_\pi T_v = \sum_{u \in V} \pi_u \mathbf{E}_u T_v,$$

where $\mathbf{E}_u T_v$ is the expected time to hit v starting from u. The quantity $\mathbf{E}_\pi T_v$ can be expressed as $\mathbf{E}_\pi T_v = Z_{vv}/\pi_v$, where

$$Z_{vv} = \sum_{t \geq 0} (P_v(v,t) - \pi_v), \tag{3.12}$$

and $P_v(v,t)$ is the probability that a walk starting from v returns to v at step t. Thus,

$$\mathbf{Var}\ T_v^+ = \frac{2Z_{vv} + \pi_v - 1}{\pi_v^2}. \tag{3.13}$$

The quantity Z_{vv} can be bounded by $1 - \pi_v \leq Z_{vv} \leq 1/(1 - \lambda_2)$, where $1 - \lambda_2$ is the eigenvalue gap of the transition matrix (see (3.14) and (3.15) for a proof of the upper bound). The lower bound follows from the fact that $P_v(v,t)$ tends to π_v from above (see [16] Proposition 10.18). Thus, for rapidly mixing random walks (e.g., walks on graphs with positive eigenvalue gap) $Z_{vv} = C$ constant. Because $\pi_v = w(v)/w_G$, both the expected first-return time $1/\pi_v$ and the variance $\sim (2C-1)/\pi_v^2$ of first-return time decrease with increasing vertex weight $w(v)$. This implies that returns to high-weight vertices should make the best estimates for w_G and that they will return sample values more often and more reliably.

3.3.1. Bounds on number of visits from stationarity. For a walk starting from stationarity, let $N(t)$ be the number of visits to vertex v in t steps. By using (2.2) of Theorem 2.1 with $f(v) = 1/2$ and $f(u) = 0,\ u \neq v$, and $\epsilon = \delta\pi_v < 1/2$, we obtain

$$\Pr(|N(t) - t\pi_v| \geq 2\delta\pi_v t) \leq 2\exp\left(-t(1-\lambda_0)(\delta\pi_v)^2\right).$$

This inequality can be useful only once $t = \omega((m/d(v))^2)$. This highlights once again the value of starting the walk from a high-degree vertex.

3.3.2. Methods used in the experimental plots. The quantity $\mathbf{E}_\pi T_v$ can be bounded in various ways to give estimates of Z_{vv} and $\mathbf{Var}\ T_v^+$. We give two methods: (M1) an estimate based on eigenvalue gap and (M2) a direct estimate from the return time data. One standard deviation of the sample mean for estimates of the number of edges was derived by these methods. This is illustrated in Figure 1(a): the outer dashed curve is obtained using method (M1) and the inner dashed curve using method (M2). The graph used in those experiments is described in Section 4.

M1. From (3.12) we have

$$Z_{vv} = \sum_{t \geq 0} (P_v(v,t) - \pi_v) \leq \sum_{t \geq 0} |P_v(v,t) - \pi_v|. \tag{3.14}$$

Using the result that $|P_v(v,t) - \pi_v| \leq \lambda_2^t$ (see, e.g., [18]) gives

$$Z_{vv} \leq \frac{1}{1-\lambda_2}. \tag{3.15}$$

M2. We estimate Z_{vv} directly from the first-return time data. Let T be a mixing time of a random walk on a graph G. The method was originally described in [6] for regular graphs, but was subsequently generalized over a series of articles. It states that, subject to Z_{vv} being constant, $T\pi_v = o(1)$ and $T\pi_v = \Omega(1/n^2)$, then for $t > T \log n$ the probability $\rho(t)$ that a first return to v has not occurred by t is of the form

$$\rho(t) \sim \exp\left(-t/\mathbf{E}_\pi T_v^+\right).$$

Replacing $1 - \rho(t)$ by the proportion $y(t)$ of returns at or before step t estimates Z_{vv}. For the plot in Figure 1(a), an estimate of $\widehat{Z}_{vv} = 1.6$ was obtained.

4. EVALUATION OF RANDOM-WALK-BASED METHODS

Figures 1–3 show the convergence of our experiments as a function of the number of returns k to the start vertex of the walk. The test networks included here are a triangle-closing preferential attachment graph and a sample from the WWW (the Google web sample).

The figures are presented as follows. The horizontal axis is k – the number of returns to the start vertex. The vertical axis is the estimate of the property. The data points plotted are based on 10 independent experiments. The underlying data points appear close to each other because there can be several returns within a short period, followed by a long wait for the next return. In all plots, the thick line, "Experiments Average," is our estimate – the sample mean as a function of $10k$ (the kth return in 10 experiments); the two dotted lines, "Experiments Deviation," show one standard deviation of the sample mean; and the horizontal line through the middle of the plot area shows the true value of the property. For the estimates based on the times of the first returns of weighted random walks, we plot also the standard deviation of the sample mean estimated using method M2 (the dashed curves "Z_{vv} Deviation"). Finally, for the estimates of the number of edges, we also computed an upper bound on the standard deviation using method M1. This bound is shown in Figure 1(a) by the two outer dashed curves "Bound on Deviation," but is outside of the visible area in the plot in Figure 2.

4.1. Hybrid Triangle Closing Model

We use a hybrid triangle closing model, which extends the preferential attachment model of [3] and generates graphs as follows. At each step we add a new vertex v with r edges to the existing graph. To add a vertex v, we first attach to an existing vertex x chosen preferentially. The remaining $r - 1$ edges from v are added as follows. With probability p we attach to a vertex chosen by preferential attachment, and with probability $1 - p$ we add an edge from v to a random neighbor of vertex x. Using this approach, we are able to control the number of triangles generated while maintaining the power-law degree distribution to be asymptotic to 3.

We generated a graph using this model with $n = 600,000, m \sim 1.8 * 10^6$ and $p = 0.6$. At each step, $r = 3$ edges were added. The total number of triangles was 550,499.

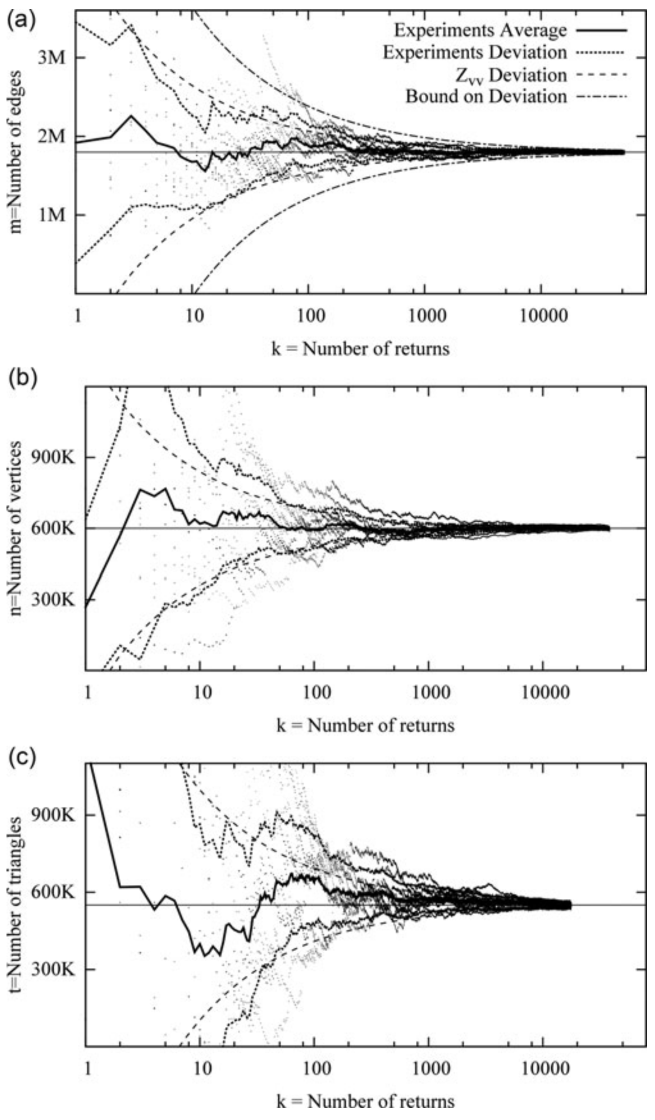

Figure 1 Triangle-closing preferential-attachment graph. Estimate of number of edges, vertices, and triangles. The key from (a) applies to all plots.

The graph has a power-law coefficient of 2.9. The second eigenvalue of the transition matrix (simple random walk) is 0.88265, making the eigenvalue gap 0.1175.

Figure 1 shows the convergence of the edge $\widehat{m}$, vertex $\widehat{n}$, and triangle $\widehat{t}$ estimates computed using the weighted random walk method described in Section 3.2. The random walks started at a vertex u of degree $d(u) = 61{,}824$, and belonging to $t(u) = 70{,}045$ triangles. The weights of this vertex are $w_{SRW}(u) = d(u) = 61{,}824$, for the simple random walk used to estimate the number of edges; $w_{TRW}(u) = d(u) + 2t(u) = 201{,}914$, for the weighted random walk used to estimate the number of triangles; and $w_{VRW}(u) = 15{,}201$,

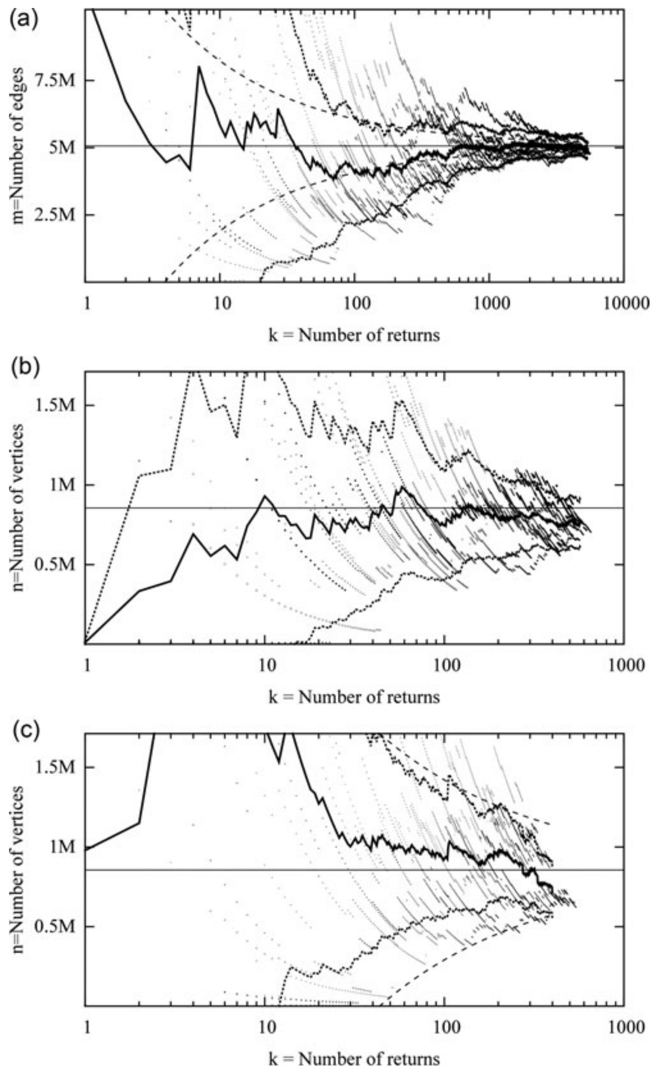

Figure 2 Google web graph. Estimate of the number of edges (a), the number of vertices by cycle formula (b), and the number of vertices by return times of weighted random walks (c). The key is the same as that in Figure 1.

for the weighted random walk used to estimate the number of vertices. The expected first-return times to vertex u are 58, 34, and 79, respectively.

All 10 experiments gave reasonable estimates of all three parameters after roughly 100 to 1,000 returns to the start vertex, that is after at most $n/10$ samples (visits to a vertex). Figure 1(a) shows good rate of convergence of the edge estimate $\widehat{m}$, and a good match between the standard deviation of the experimental data (the dotted "Experiments Deviation" lines) and the standard deviation obtained by estimating the parameter Z_{vv} (the "Z_{vv} Deviation" curves). The estimates using the cycle formula, as described in Section 3.1, were similar, so we omit the details.

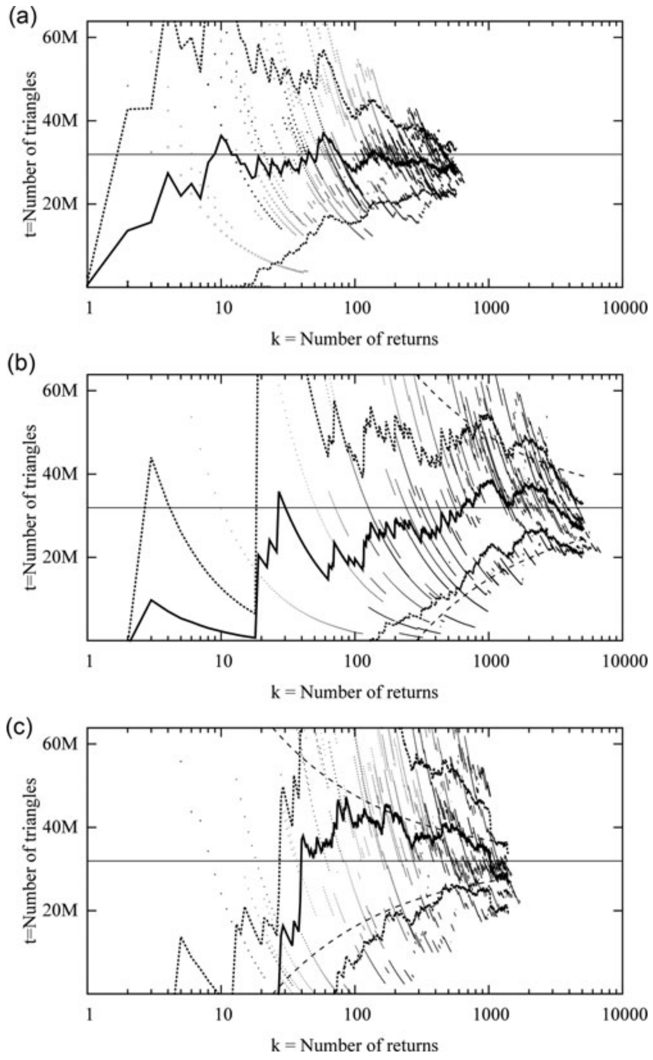

Figure 3 Google web graph. Estimate of number of triangles: cycle formula (a), return times of weighted random walks with the weight factor $c = 1$ (b), and $c = 0.1$ (c). The key is the same as that in Figure 1.

4.2. Google Web Sample

We used a sample from the Google web graph, which was released for the purposes of the Google programming contest in 2002 [15]. This dataset consists of 855,802 vertices, 5,066,842 edges, and 31,356,298 triangles. By direct computation, we found that the second eigenvalue of the transition matrix (simple random walk) is 0.99970, making the eigenvalue gap 3×10^{-4}. For this network, the estimates converged slower than in the generated test graphs, with much more variation around the expected values. The structure of the graph is very inhomogeneous; presumably this is why the dataset is made available.

In our experiments, random walks started at a vertex u of degree $d(u) = 6{,}353$, with $t(u) = 53{,}371$ triangles. For the simple random walk, which is used for estimating the number of edges, the expected first-return time to the start vertex u is equal to 1,595. The computed estimates for the number of edges are given in Figure 2(a). The convergence is slow and the theoretical standard deviation bound is outside the figure. We have to wait for about 1,000 returns to the start vertex to get a reasonable estimate, which means that the number of samples (visits to a vertex) is roughly of the same order as the number of vertices n.

We compare the performance of the cycle-formula method and the weighted-random-walk method for estimating the number of vertices and the number of triangles in the Google web graph. (Observe that these two methods are exactly the same when used for estimating the number of edges: $f(v) \equiv 1$ for the cycle-formula method, and $w(v,u) \equiv 1$ for the weighted-random-walk method.) For the weighted-random-walk method, the weights of the start vertex u are $w_{TRW}(u) = d(u) + 2t(u) = 113{,}095$ and $w_{VRW}(u) = 855$. The expected first-return times to vertex u are 1,753 and 2,002, respectively.

Figure 2(b) and (c) show the estimates of the number of vertices computed by the cycle-formula method and the weighted-random-walk method. Figure 3(a) and (b) show the estimates of the number of triangles. The convergence is slow for both methods, but the estimates given by the cycle formula are more accurate, especially for the number of triangles.

To see if we can improve the performance of the weighted-random-walk method for estimating the number of triangles, we varied the value of the parameter c in the edge-weight formula (3.10) to avoid distorting the already small eigenvalue gap even further. The edge weight is $w(e) = 1 + ct(e)$, the vertex weight is $w(u) = d(u) + 2ct(u)$, and the graph weight is $2m + 6ct(G)$. To simplify the experiments, we used the correct value of m. The plots for $c = 1$ and $c = 1/10$ are given in Figure 3(b) and (c), respectively. The value $c = 1/10$ worked quite well, giving clearly better results than the value $c = 1$ (after n steps, the standard deviation estimated using the method M2 is four times smaller for $c = 1/10$), but not fully matching the performance of the cycle-formula method. The value of c can be optimized by further experiments.

Figures 4 and 5 show the estimates of the number of vertices and the number of triangles in the Google web graph, based on running totals sampled by a simple random walk. The plots are broadly consistent with the concentration theorems discussed in Section 2.2 and the eigenvalue gap 3×10^{-4} of this graph: the convergence should become noticeable when the number of steps becomes larger than the reciprocal of the eigenvalue gap. The estimates obtained by the running-totals method look considerably better than the estimates based on the first-returns methods, but a direct comparison of these two approaches is not fair. For example, in the case of estimating the number of vertices, the running-totals method estimates only $n/(2m)$. We then use the *exact* value m to obtain an estimate of the number of vertices. If we used an estimate $\hat{m}$ obtained from the first returns, the convergence would be more comparable with what we see in the plots in Figure 2.

5. GENERAL DISCUSSION OF RESULTS AND CONCLUSIONS

Although the methods work reasonably well on all our test graphs, some real graphs such as the Google web sample showed more fluctuation in the results. It is generally accepted that online networks are expanders, but they are not as good expanders as our random graphs test models, which might adversely affect the mixing rate of the walks.

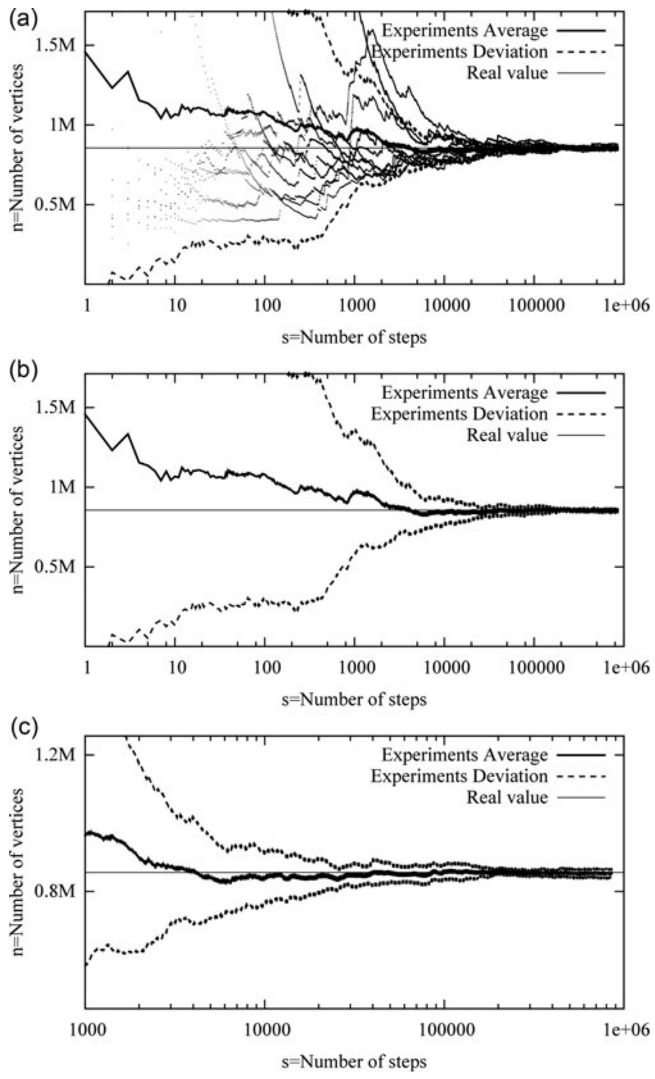

Figure 4 Google web graph. Estimates of the number of vertices by the running totals: all 10 experiments, the average and the deviation (a); the average and the deviation (b); from step 1,000 (c).

However, even if the estimates do fluctuate, their tendency to track the correct value is reassuring.

The true reason for the under performance on the Google sample is not clear, but could be due to the structure of the Google graph. It might be the case that some walks reached pendant tree-like subgraphs from which it was hard to return. The work by [5] gives a detailed view of the structure of the web and point to the existence of such parts (referred to as "tendrils"). Coupled with this, direct inspection suggested the high-degree vertices were very clustered with poor conductance between them and the rest of the graph.

The conclusion seems to be straightforward: the cycle-formula method performed well on a range of test graphs (not all given here) and outperformed the weighted-random-

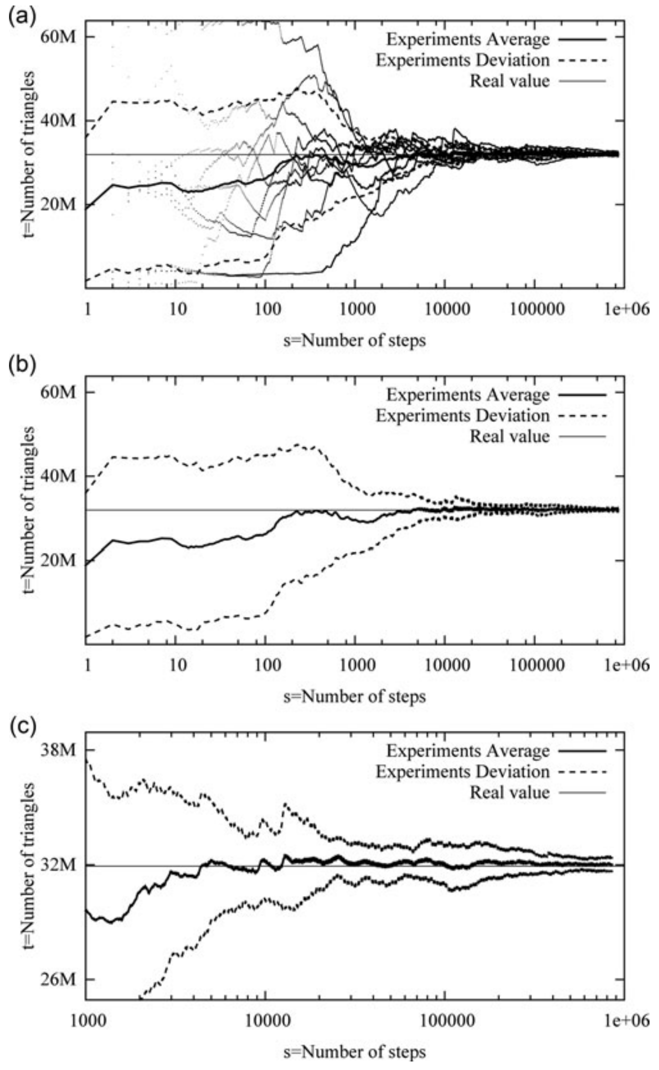

Figure 5 Google web graph. Estimates of the number of triangles by the running totals: all 10 experiments, the average and the deviation (a); the average and the deviation (b); from step 1,000 (c).

walks method on the more difficult Google web graph. However, with some tuning of the weight parameter *c* the weighted random walk performed best for the difficult case of estimating the number of triangles. Both methods returned good estimates in a reasonable time and have practical value. An important point, it seems, is to start the walk from a high-weight vertex.

FUNDING

Research supported in part by EPSRC grant EP/J006300/1 and Samsung Global Outreach Project “Fast low cost methods to learn structure of large networks.”

REFERENCES

[1] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. Available online (http://stat-www.berkeley.edu/pub/users/aldous/RWG/book.html), 2002.

[2] K. Avrachenkov, N. Litvak, M. Sokol, and D. Towsley. "Quick Detection of Nodes with Large Degrees." *Internet Mathematics* 10 (2014), 1–19.

[3] A. Barabasi and R. Albert. "Emergence of Scaling in Random Networks." *Science* 5439:286 (1999), 509–512.

[4] M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani. "Estimating aggregates on a peer-to-peer network." Technical Report, CS Dept, Stanford University, 2003.

[5] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. "Graph Structure in the Web." *Computer Networks* 33 (2000), 309–320.

[6] C. Cooper and A. Frieze. "The Cover Time of Random Regular Graphs." *SIAM Journal of Discrete Mathematics* 18:4 (2005), 728–740.

[7] C. Cooper, T. Radzik, and Y. Siantos. "Estimating Network Parameters Using Random Walks." In *Proceedings of 2012 Fourth International Conference on Computational Aspects of Social Networks* (CASoN 2012), pp. 33–40. New York, NY: IEEE, 2012.

[8] C. Cooper, T. Radzik, and Y. Siantos. "Fast Low-Cost Estimation of Network Properties Using Random Walks." In *Proceedings of Algorithms and Models for the Web Graph – WAW 2013*, LNCS 8305, pp. 130–143. London, UK: Springer, 2013.

[9] I. Dinwoodie, "A Probability Inequality for the Occupation Measure of a Reversible Markov Chain." *Ann. Appl. Probab.* 5 (1995), 37–43.

[10] W. Feller. *An Introduction to Probability Theory and Its Applications*, Vol. I. Hoboken, NJ: John Wiley & Sons, Inc., 1968.

[11] A. Ganesh, A-M. Kermarrec, E. Le Merrer, and L. Massoulie. "Peer Counting and Sampling in Overlay Networks Based on Random Walks." *Distrib. Comput.* 20 (2007), 267–278.

[12] S. Goel and M. Salganik. "Respondent Driven Sampling as Markov Chain Monte Carlo." *Statistics in Medicine* 28:17 (2009), 2202–2229.

[13] L. Katzir, E. Liberty, and O. Somekh. "Estimating Sizes of Social Networks via Biased Sampling." In *Proceedings of the 20th International Conference on World Wide Web* (WWW 2011), pp. 597–606. New York, NY: ACM, 2011.

[14] C. Léon and F. Perron. "Optimal Hoeffding Bounds for Discrete Reversible Markov Chains." *The Annals of Applied Probability* 14:2 (2004), 958–970.

[15] J. Leskovec. "Stanford Network Analysis Package." Available online (http://snap.stanford.edu/), 2009.

[16] D. Levin, Y. Peres, and E. Wilmer. *Markov Chains and Mixing Times*. Providence, RI: AMS, 2009.

[17] P. Lezaud. "Chernoff-Type Bound for Finite Markov Chains." *The Annals of Applied Probability* 8:3 (1998), 849–867.

[18] L. Lovasz. "Random Walks on Graphs: A Survey." *Bolyai Society Mathematical Studies* 2 (1996), 353–397.

[19] L. Massoulie, E. Le Merrer, A-M. Kermarrec, and A. Ganesh. "Peer Counting and Sampling in Overlay Networks: Random Walk Methods." In *PODC 2006*, pp. 123–132. New York: ACM, 2006.

[20] R. Wagner. "Tail Estimates for Sums of Variables Sampled by a Random Walk." *Combinatorics, Probability, and Computing* 17:2 (2008), 307–316.