Bias Correction in Clustering Coefficient Estimation

Roohollah Etemadi, Jianguo Lu
School of Computer Science, University of Windsor
Windsor, ON, Canada
etemadir, jlu@uwindsor.ca

Abstract*—Clustering coefficient ($\mathcal{C}$) is an important structural property to understand the complex structure of a graph. Calculating $\mathcal{C}$ is a computationally intensive task. Thereby, sampling-based methods have attracted substantial research for estimating $\mathcal{C}$, and the closely related metric, the number of triangles. Unfortunately, widely used estimators for $\mathcal{C}$ are biased. We quantify the bias using Taylor expansion and find that the bias can be determined by the number of shared wedges and triangles in the sample. Based on the understanding of the bias, we give a new estimator that corrects the bias. The results are derived analytically and verified extensively in 56 networks ranging in different size and structure. The experiments reveal that the bias ranges widely from data to data. The relative bias can be as high as 4% or can be negative. For most of the graphs, the bias is small, although every graph does have a bias as quantified by our analytical results. Negative or small biases occur in online social networks where clustering coefficient is typically high. Positive and large biases typically occur in Web graphs, where there are nodes with high degrees but few neighboring nodes connecting with each other.

***Keywords*-Graph sampling; Clustering Coefficient; Estimating algorithms; Bias; Variance.**

I. Introduction

Analyzing very large graphs has been receiving increasing attention from both academia [1] and industry [2]. Clustering coefficient (hereafter $\mathcal{C}$), also called transitivity, is one of the key graph properties to characterize the complex structure of networks [3]. Loosely speaking, it measures the probability whether a friend's friend is also a friend in social networks. Since its incept in [3], it finds wide applications in a variety of areas, such as computer networks (spam detection [4]), social networks (community detection and blog analysis [5]), biology (DNA sequence analysis [6]), economy (risk study [7]), and many more.

There has been tremendous work on estimating graph properties, in general, using sampling techniques [1], [8]. Sampling is necessary when the graph is very large, or the graph in its entirety is not available, such as in the case of online social networks. The estimation of $\mathcal{C}$ property has been specifically addressed in [9]–[14]. Much more work has been directed to a closely related triangle counting problem [15]–[20].

Let Λ denote the number of wedges (or paths of length two), Δ the number of closed-wedges. Metric $\mathcal{C}$ is defined as a ratio between the two, i.e.,

$$\mathcal{C} = \frac{\Delta}{\Lambda}. \tag{1}$$

Note that here we restrict our discussion to the global $\mathcal{C}$ as defined above for the sake of simplicity. There are other notions of $\mathcal{C}$s, such as local and average $\mathcal{C}$, that are beyond the scope of this paper. However, our method can be easily extended to those $\mathcal{C}$s.

Suppose that $\widehat{\Delta}$ and $\widehat{\Lambda}$ are unbiased estimators for Δ and Λ, respectively. In other words, $\mathbb{E}(\widehat{\Delta}) = \Delta$, and $\mathbb{E}(\widehat{\Lambda}) = \Lambda$. It has been taken for granted, e.g., in [10], [12], [14] and [11], that the $\mathcal{C}$ estimator $\widehat{\mathcal{C}}$ is:

$$\widehat{\mathcal{C}} = \frac{\widehat{\Delta}}{\widehat{\Lambda}}. \tag{2}$$

Unfortunately, this is a biased estimator as we can see from the fact that $\mathbb{E}(X/Y) \neq \mathbb{E}(X)/\mathbb{E}(Y)$. More precisely, by applying expectation on the estimator, we have:

$$\mathbb{E}(\widehat{\mathcal{C}}) = \mathbb{E}\left(\frac{\widehat{\Delta}}{\widehat{\Lambda}}\right) \neq \frac{\mathbb{E}(\widehat{\Delta})}{\mathbb{E}(\widehat{\Lambda})} = \frac{\Delta}{\Lambda} = \mathcal{C}.$$

While it is easy to understand the existence of bias, quantifying and correcting the bias is a challenging task. Recently, Jha et al. [14] and Ahmad et al. [12] noticed the bias problem and left it as an open problem to solve. In 2015, Jha et al. discussed the bias problem again, but could not quantify it [21].

The analysis of the bias needs to be embedded in a concrete sampling method. We base our following discussions on random edge sampling. Random edge sampling has been widely used for estimating $\mathcal{C}$ [12], [14], [21], triangles [15]–[20], and other graph properties [8], [22], [23]. It is also closely related to other sampling methods. For example, random walk [10], [11], [13] is an approximation of random edge sampling in that their node sampling probabilities are asymptotically equal in undirected graphs. Random node sampling can also be associated with random edge sampling–when we sample node with probability proportional to its size (PPS), it is actually a kind of random edge sampling in the sense that sampling probability of the node is the same in two sampling schemes.

For this random edge sampling scheme, we quantify the bias using the 'power method' [24]. It involves a Taylor expansion that results in a long formula. The intuitive understanding is lost in the complex formula without simplification. Hence, we simplify the formula, and derive an adjusted estimator as follows:

$$\widehat{\mathcal{C}}^{*} = \frac{\widehat{\Delta}}{\widehat{\Lambda}}\left[1+\frac{r}{p}\right]^{-1}, \tag{3}$$

where p is the sampling probability, r is a constant determined by the graph topology that will be explained later in Section II. Roughly speaking, r is dominated by the ratio of the third moment and the square of the second moment of the degrees of the graph. The corrected estimator in Eq. 3 highlights the importance of the problem particularly in the age of big data: when the graph is very large, only a small fraction of the graph is needed to achieve high accuracy, resulting in a very small p. Although r, in general, is a very small number, r/p may not be neglectable in this case. We will show that r/p can be as high as 0.04 in certain cases.

Eq. 3 is good for understanding the nature of the bias problem. However, it cannot be used for estimation in practice because r is unknown from the sample. In other words, we also need to estimate r to correct the bias. Thus, we derive a corrected estimator for random edge sampling as below:

$$\widehat{\mathcal{C}}^{+} = \frac{\Delta_g}{\Lambda_g}\left[1+r_g\right]^{-1}, \tag{4}$$

where $r_g = 2\Psi_g/\Lambda_g^2 - \Omega_g/\Delta_g\Lambda_g$. Variable Ψ_g is the number of shared wedges, Ω_g is the number of shared wedges and closed-wedges, all in sample graph g. We show that the result can be simplified further by taking the first term only in the above formula, assuming that the graph is large. Based on this, the bias can be quantified by the second and third moments of the degrees of the nodes in the graph. Since the simplified result is derived using several approximations, we need to empirically evaluate the approximation using real graphs. The result is confirmed and explained on 56 real-world graphs.

II. The Bias Problem

A. Clustering Coefficient

Let $G(\mathcal{V},\mathcal{E})$ be a simple graph, where $\mathcal{V}$ and $\mathcal{E}$ are the set of nodes and the set of edges, respectively. Let $N=|\mathcal{V}|$, $M=|\mathcal{E}|$, and each node is labeled as $1,2,\ldots,N$. Let d_i denote the degree of node i, for $i \in \{1,2,\ldots,N\}$. A wedge $\mathcal{W}$ is a triplet (u,v,w), where $u,v,w \in \mathcal{V}$ are three distinct nodes, $(u,v) \in \mathcal{E}$, and $(v,w) \in \mathcal{E}$. Wedge $\mathcal{W}$ is closed if $(u,w) \in \mathcal{E}$. Otherwise it is open. A triangle consists of three (closed) wedges. Let $\Lambda_i = d_i(d_i-1)/2$ denote the number of wedges for node i, and Δ_i the number of closed-wedges for node i. Clustering coefficient is defined as the proportion of the wedges that are closed [9], [25], i.e.,

$$\mathcal{C} = \frac{\sum_{i=1}^{N}\Delta_i}{\sum_{i=1}^{N}\Lambda_i} = \frac{\Delta}{\Lambda}. \tag{5}$$

Table I: Summary of the notations

Notation	Meaning
$G(\mathcal{V},\mathcal{E})$	Input graph (undirected and no self-edges)
g	A subgraph of G
N, M	Number of nodes and edges in G
n	Sample size
d_i	Degree of node i in G
Λ	# wedges in G
Δ	# closed-wedges in G
Λ_g	# wedges in g
Δ_g	# closed-wedges in g. Closeness checked on G
Λ_i	# wedges of node i
Δ_i	# closed-wedges of node i
Ψ	# pairs of shared wedges in G
Ω	# pairs of a wedge and a closed-wedge sharing one edge in G
Ψ_g	Ψ counted in g
Ω_g	Ω counted in g
$\langle d\rangle$	Average degree of G. $\langle d\rangle = \frac{1}{N}\sum_{i=1}^{N} d_i$.
$\langle d^2\rangle$	Second moment. $\langle d^2\rangle = \frac{1}{N}\sum_{i=1}^{N} d_i^2$.
$\langle d^3\rangle$	Third moment. $\langle d^3\rangle = \frac{1}{N}\sum_{i=1}^{N} d_i^3$.

Table I summarizes a list of notions used in this paper.

B. The Sampling Scheme

Our sampling scheme is based on edge sampling. It selects n distinct edges from the original graph G uniformly at random to generate a subgraph g. When interpreted as a node sampling process, it is the same as PPS sampling, where nodes are sampled with Probability Proportional to Size. In this sense, random walk sampling is an approximation to random edge sampling.

Let Λ_g be the count of wedges in g, and Δ_g denotes the number of closed-wedges restricted to the wedges of g in which their closenesses are checked based on the original graph G. The expectations of Λ_g and Δ_g are

$$\mathbb{E}(\Lambda_g) = \Lambda p^2, \quad \mathbb{E}(\Delta_g) = \Delta p^2. \tag{6}$$

Hence, the unbiased estimator for Λ and Δ [20] are

$$\widehat{\Lambda} = \frac{\Lambda_g}{p^2}, \quad \widehat{\Delta} = \frac{\Delta_g}{p^2}.$$

Under this sampling scheme, the biased estimator in Eq. 2 is instantiated as

$$\widehat{\mathcal{C}} = \frac{\Delta_g}{\Lambda_g}. \tag{7}$$

C. The Bias

We quantify the bias using the classic power method [24]. By Taylor expansion, the quadratic approximation of x/y in

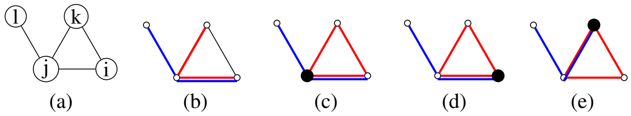


Figure 1: Illustration of shared wedges and closed-wedges. (a) A sample graph; (b) Wedge (l, j, i) shares with wedge (k, j, i); (c-d) Wedge (l, j, i) shares with closed-wedges (i, j, k), (k, i, j); (e) Wedge (l, j, k) shares with closed-wedge (j, k, i). The large node in the plot indicates the centre node of the closed-wedge. E.g., in Panel (c) the closed-wedge is (k, j, i).

the neighbourhood of (a, b) is

$$\frac{x}{y} \approx \frac{a}{b} + \left(\frac{1}{b}(x-a) - \frac{a}{b^2}(y-b)\right) + \left(\frac{a}{b^3}(y-b)^2 - \frac{1}{b^2}(x-a)(y-b)\right).$$

Applying expectation on both sides of the equation yields:

$$\mathbb{E}\left(\frac{x}{y}\right) \approx \mathbb{E}\left(\frac{a}{b}\right) + \frac{1}{b}(\mathbb{E}(x)-a) - \frac{a}{b^2}(\mathbb{E}(y)-b) + \frac{a}{b^3}\mathbb{E}(y-b)^2 - \frac{1}{b^2}\mathbb{E}(x-a)(y-b). \tag{8}$$

Take $a = \mathbb{E}(\Delta_g)$, $b = \mathbb{E}(\Lambda_g)$, $x = \Delta_g$, and $y = \Lambda_g$. Note that by definition $cov(\Delta_g, \Lambda_g) = \mathbb{E}((\Delta_g - \mathbb{E}(\Delta_g))(\Lambda_g - \mathbb{E}(\Lambda_g)))$, and $var(\Lambda_g) = \mathbb{E}(\Lambda_g - \mathbb{E}(\Lambda_g))^2$. Eq. 8 can be rewritten as:

$$\mathbb{E}\left(\frac{\Delta_g}{\Lambda_g}\right) \approx \frac{\mathbb{E}(\Delta_g)}{\mathbb{E}(\Lambda_g)} + \frac{\mathbb{E}(\Delta_g)var(\Lambda_g)}{\mathbb{E}(\Lambda_g)^3} - \frac{cov(\Delta_g, \Lambda_g)}{\mathbb{E}(\Lambda_g)^2}. \tag{9}$$

Applying the fact that $\mathcal{C} = \Delta/\Lambda = \mathbb{E}(\Delta_g)/\mathbb{E}(\Lambda_g)$, the above equation can be reformulated as

$$\mathbb{E}(\widehat{\mathcal{C}}) \approx \mathcal{C}\left(1 + \frac{var(\Lambda_g)}{\mathbb{E}(\Lambda_g)^2} - \frac{cov(\Delta_g, \Lambda_g)}{\mathbb{E}(\Delta_g)\mathbb{E}(\Lambda_g)}\right). \tag{10}$$

We can see that the bias hinges on the variance of Λ_g and covariance between Δ_g and Λ_g. Before going into the derivation of the variance and covariance, we need to understand the dependency between two wedges, and the dependency between a wedge and a closed-wedge as illustrated in Fig. 1. Panel (a) is an example graph. In the graph, two wedges (k,j,i) and (l,j,i) share a common edge (j,i). We use Ψ to denote the number of such sharing in a graph. Panel (c) is an example of sharing between a wedge and a closed-wedge: wedge (l,j,i) and closed-wedge (k,j,i) share a common edge (j,i). Panels (d) and (e) are similar to (c), except that the center node in the wedges is changed. Here we showcase the following distinction between a closed-wedge and a triangle: a closed-wedge is similar to a triangle except that each triangle has three closed-wedges, and correspondingly, a closed-wedge has a center node. We denote the number of such sharing using Ω. In a graph, Ψ and Ω can be very large, much larger than Λ and Δ.

Let λ_i be an indicator for the i^{th} wedge in the original graph G. Indicator λ_i is 1 if the wedge is sampled in g; otherwise it is 0. Let $1, 2, .., \Lambda$ be the labels for all wedges in G. $\Lambda_g = \sum_{i=1}^{\Lambda} \lambda_i$. By applying var on both sides on the equation, we have

$$var(\Lambda_g) = var(\sum_{i=1}^{\Lambda} \lambda_i) = \sum_{i=1}^{\Lambda}\sum_{j=1}^{\Lambda} cov(\lambda_i, \lambda_j) = \sum_{i=1}^{\Lambda} var(\lambda_i) + \sum_{i \neq j} cov(\lambda_i, \lambda_j). \tag{11}$$

Variable λ_i follows a Bernoulli distribution with probability p^2 and $var(\lambda_i) = p^2 - p^4$. For the covariance, we need to consider the cases of dependent wedges. Wedges λ_i and λ_j are dependent when they have a shared edge in graph G as illustrated in Panel (b) of Fig. 1. In such a case, $\mathbb{E}(\lambda_i\lambda_j) = p^3$, hence $cov(\lambda_i, \lambda_j) = \mathbb{E}(\lambda_i\lambda_j) - \mathbb{E}(\lambda_i)\mathbb{E}(\lambda_j) = p^3 - p^4$. There are 2Ψ cases (each $cov(\lambda_i, \lambda_j)$ has an equivalent $cov(\lambda_j, \lambda_i)$) , we have the following by substituting $var(\Lambda_i)$ and $cov(\lambda_i, \lambda_j)$ in Eq. 11.

$$var(\Lambda_g) = \Lambda(p^2 - p^4) + 2\Psi(p^3 - p^4). \tag{12}$$

Next we derive $cov(\Delta_g, \Lambda_g)$. Let δ_i be the indicator for the i^{th} closed-wedge in graph G. The covariance between Δ_g and Λ_g is

$$cov(\Delta_g, \Lambda_g) = \sum_{i=1}^{\Delta}\sum_{j=1}^{\Lambda} cov(\delta_i, \lambda_j) = \sum_{i=1}^{\Delta}\sum_{j=1}^{\Lambda} \mathbb{E}(\delta_i\lambda_j) - \mathbb{E}(\delta_i)\mathbb{E}(\lambda_j).$$

When δ_i and λ_j are independent, they share no edges, the covariance between them is $cov(\delta_i, \lambda_j) = 0$. There are two cases that δ_i and λ_j are dependent: the wedge has either one edge or two edges shared with the closed-wedge. In the first case, $\mathbb{E}(\delta_i\lambda_j) - \mathbb{E}(\delta_i)\mathbb{E}(\lambda_j) = p^3 - p^4$. Note that it is not $p^4 - p^5$ because our sampling method checks the closeness of a wedge whenever a wedge is encountered. Hence the probability of seeing a closed-wedge in a sample is p^2 instead of p^3. In the second case, the wedge is contained in the closed-wedge, and $\mathbb{E}(\delta_i\lambda_j) - \mathbb{E}(\delta_i)\mathbb{E}(\lambda_j) = p^2 - p^4$. Since there are Ω number of one-edge sharing, and Δ number of two-edge sharing, the covariance is

$$cov(\Delta_g, \Lambda_g) = \Delta(p^2 - p^4) + \Omega(p^3 - p^4). \tag{13}$$

Substitute Eq. 12 and Eq. 13 into 10, and assume that $1 - p \approx 1 - p^2 \approx 1$ because the sampling probability is very small for large graphs, we obtain

$$\mathbb{E}(\widehat{\mathcal{C}}) \approx \mathcal{C}\left(1+\frac{2\mathbb{E}(\Psi_g)}{\mathbb{E}(\Lambda_g)^2}-\frac{\mathbb{E}(\Omega_g)}{\mathbb{E}(\Lambda_g)\mathbb{E}(\Delta_g)}\right). \tag{14}$$

Let relative bias $RB = \mathbb{E}(\widehat{\mathcal{C}})/\mathcal{C} - 1$. After rearranging the formula above, and remember that $\mathbb{E}(\Lambda_g) = \Lambda p^2$, $\mathbb{E}(\Delta_g) = \Delta p^2$, $\mathbb{E}(\Omega_g) = \Omega p^3$, and $\mathbb{E}(\Psi_g) = \Psi p^3$, we quantify the bias with:

$$RB \approx \frac{1}{p}\left(\frac{2\Psi}{\Lambda^2} - \frac{\Omega}{\Lambda\Delta}\right), \tag{15}$$

and it can be estimated by

$$\widehat{RB} = \frac{2\Psi_g}{\Lambda_g^2} - \frac{\Omega_g}{\Lambda_g\Delta_g}. \tag{16}$$

Correspondingly, we have the following bias-corrected estimators:

$$\widehat{\mathcal{C}}^{*} = \frac{\Delta_g}{\Lambda_g}\left[1+\frac{1}{p}\left(\frac{2\Psi}{\Lambda^2} - \frac{\Omega}{\Lambda\Delta}\right)\right]^{-1}, \tag{17}$$

where p is the sampling probability, Ψ and Ω are the counts for shared wedges and shared wedges and closed-wedges, respectively. The practical estimator based on subgraph only is

$$\widehat{\mathcal{C}}^{+} = \frac{\Delta_g}{\Lambda_g}\left[1+\frac{2\Psi_g}{\Lambda_g^2} - \frac{\Omega_g}{\Lambda_g\Delta_g}\right]^{-1}, \tag{18}$$

where Ψ_g and Ω_g are the corresponding counts for Ψ and Ω in subgraph g.

D. Counting Ψ and Ω

The estimator $\widehat{\mathcal{C}}^{+}$ relies on Ψ_g and Ω_g. Ψ and Ω can be computed efficiently, especially for sample graphs that are typically not very large. Ψ is counted by iterating through all the edges using the following equation:

$$\Psi = \sum_{e(j,k)\in\mathcal{E}}\left[\binom{d_j-1}{2}+\binom{d_k-1}{2}+(d_j-1)(d_k-1)\right], \tag{19}$$

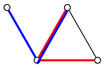
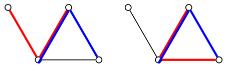
where j and k are end nodes of edge e and d_x is the degree of node x for $x = j, k$. This can be verified by looking at the three cases of shared wedges in Panel (a) of Fig. 2. For two first cases, there are $\binom{d_j-1}{2}+\binom{d_k-1}{2}$ shared wedges. For last case, there are $(d_j-1)(d_k-1)$ shared wedges.

Metric Ω is computed by summarizing the overlaps for each node in all the triangles using the following equation:

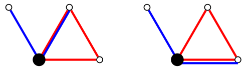
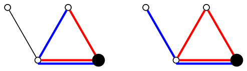
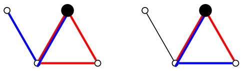
$$\Omega = \sum_{(i,j,k)\in\Delta} 4d_j - 6, \tag{20}$$

where i, j, k are nodes in graph G and (i, j, k) is a closed-wedge with center node j. Note that $d_j \geq 2$, hence Ω is always a positive value.

The equation can be explained using the example depicted in Panel (b) of Fig. 2. For node j, there are $d_j - 2$ number of wedges that share edge (j,k) for the the closed-wedge (i,j,k). Similarly, there are $d_j - 2$ number of shared pairs with common edge (j, i). Now, we need to look at other two closed-wedges (j,i,k) and (j,k,i). For closed-wedge (j,i,k), there are $d_j - 1$ number of wedges emanating from node j that share one edge with the closed wedge. Similarly, for closed-wedge (j, k, l), there are also $d_j - 1$ shares. Hence, overall for each node j, there are $4d_j - 6$ shared pairs.

Cases	**Overlap cases**	Ψ
For node j		$\binom{d_j-1}{2}=1$
For node k	Empty	$\binom{d_k-1}{2}=0$
Between j and k		$(d_j-1)(d_k-1)=2$

(a) Example for counting Ψ by checking edge (j, k). Need to repeat the process for every edge in the graph. For edge (j, k), $\Psi = \binom{d_j-1}{2}+\binom{d_k-1}{2}+(d_j-1)(d_k-1) = 3$.

Closed-wedge	**Overlap cases**	Φ
(i, j, k)		$d_j - 2$ for (j, k) $d_j - 2$ for (j, i)
(j, i, k)		$d_j - 1$
(j, k, i)		$d_j - 1$

(b) An example of counting Ω by checking node j. Need to repeat the process for every node in triangles in the graph. Large nodes indicate the center node of a closed-wedge. For node j, there are $2 \times (d_j - 2) + d_j - 1 + d_j - 1 = 4d_j - 6 = 6$ cases of overlaps between a wedge and a closed-wedge.

Figure 2: An example for computing Ψ and Ω in the sample graph in Fig. 1 Panel (a).

III. Experiments

A. Datasets

The bias phenomenon varies greatly from graph to graph. To find out the patterns behind, we need to experiment extensively with many different kinds of graphs. In total, we use 56 real graphs from a variety of areas such as online social networks, web graphs, Co-authorship, and citation networks. The graph size also varies from about 4×10^3 (very small) to 65×10^6 (very large). The directionality of

Table II: Properties of the networks in our experiments, sorted by graph size N.

Dataset	N($\times 10^6$)	$\langle d \rangle$	$\mathcal{C}$	$\Delta(\times 10^6)$	$\Lambda(\times 10^9)$	$\langle d^3 \rangle$ $(\times 10^6)$	$\langle d^2 \rangle (\times 10^3)$	$\frac{2\Psi}{\Lambda^2}$-$\frac{\Omega}{\Delta\Lambda}$	$\frac{2\Psi}{\Lambda^2}$	$\frac{\Omega}{\Delta\Lambda}$	Description
Ego-facebook [26]	0.004	43.69	0.519	4.8	0.009	1.09	4.6	1.2e-5	7.4e-5	6.2e-5	OSN
CA-GrQc [26]	0.005	5.52	0.629	0.1	0.0002	0.003	0.09	-7.0e-5	5.2e-4	5.9e-4	Collaboration
Wiki-vote [26]	0.007	28.32	0.125	1.8	0.014	1.2	4.1	-3.7e-6	5.9e-5	6.3e-5	OSN
AstroPh [27]	0.01	21.10	0.31	4.0	0.012	0.17	1.3	4.8e-6	3.2e-5	2.7e-5	Citation
CA-CondMat [26]	0.02	8.08	0.264	0.5	2	0.01	0.1	2.5e-5	8.3e-5	5.8e-5	Coauthorship
HepPh [27]	0.02	224.14	0.279	587	2	184	149	1.0e-7	1.8e-6	1.7e-6	Coauthorship
Enron-email [27]	0.03	10.02	0.085	2	0.025	0.8	1.4	1.6e-5	5.2e-5	3.5e-5	E-communication
Brightkite [26]	0.05	7.35	0.110	1.4	0.013	0.15	0.4	1.8e-5	5.8e-5	3.9e-5	OSN
Facebook [27]	0.06	25.64	0.147	10.5	0.07	0.43	2.2	1.0e-6	8.3e-6	7.2e-6	OSN
Epinions [27]	0.07	10.69	0.065	4.8	0.07	1.3	1.9	3.5e-6	2.3e-5	1.9e-5	OSN
Slashdot-Zoo [27]	0.07	11.82	0.023	1.6	0.06	1.1	1.7	5.2e-6	2.2e-5	1.7e-5	OSN
Prosper [27]	0.08	74.60	0.003	3.4	1.1	30	24	-1.9e-7	2.6e-6	2.8e-6	Interaction
Livemocha [27]	0.1	42.13	0.014	10.0	0.716	12	13	-2.5e-7	3.2e-6	3.5e-6	OSN
Douban [27]	0.1	4.22	0.01	0.1	0.011	0.01	0.1	-4.7e-6	1.6e-5	2.1e-5	OSN
Gowalla [26]	0.1	9.66	0.023	6.8	0.290	24	2.9	4.9e-5	5.6e-5	7.1e-6	OSN
Libimseti [27]	0.2	155.97	0.007	207	28	2,203	255	1.3e-7	6.5e-7	5.1e-7	OSN
Digg [27]	0.2	11.07	0.061	42	0.69	15	4.9	5.1e-6	9.7e-6	4.6e-6	OSN
Web-Stanford [27]	0.2	14.13	0.008	33	3.94	536	27	7.1e-6	9.7e-6	2.5e-6	Web graph
Dblp-Coau [26]	0.3	6.62	0.306	6	0.021	0.008	0.1	-4.5e-7	8.3e-6	8.7e-6	Coauthorship
Web-NotreDame [26]	0.3	6.69	0.087	26	0.304	8.3	1.8	2.5e-5	2.9e-5	4.5e-6	Web graph
Amazon [26]	0.3	5.53	0.205	2	0.009	0.002	0.06	5.5e-6	1.0e-5	5.2e-6	Co-purchasing
Actor [27]	0.3	78.68	0.166	1,040	6.26	36	32	5.1e-8	5.3e-7	4.8e-7	Collaboration
Citeseer [27]	0.3	9.03	0.049	4	0.081	0.14	0.4	5.1e-6	8.8e-6	3.6e-6	Citation
Dogster [27]	0.4	40.03	0.014	250	17	1,742	82	1.5e-6	2.4e-6	8.7e-7	OSN
Catster [27]	0.6	50.32	0.028	1,969	69	11,637	222	1.2e-6	1.5e-6	2.2e-7	OSN
Web-Berkeley [27]	0.6	19.40	0.0069	194	27.9	3,348	81	2.2e-6	2.9e-6	6.5e-7	Web graph
Web-Google [27]	0.8	9.87	0.055	40	0.727	4.5	1.6	6.3e-6	7.5e-6	1.1e-6	Web graph
Youtube [26]	1.1	5.27	0.006	9	1	30	2.6	1.2e-5	1.5e-5	3.9e-6	OSN
Dblp [27]	1.3	8.16	0.170	36	0.214	0.067	0.3	1.2e-6	2.4e-6	1.1e-6	Coauthorship
Hyves [27]	1.4	3.96	0.001	2	1.4	45	2	2.9e-5	3.0e-5	9.5e-7	OSN
Wiki-Polish [27]	1.5	55.17	0.01	3,402	308	81,387	404	1.2e-6	1.3e-6	6.5e-8	Web graph
Trec-wt10g [27]	1.6	8.33	0.014	63	4.3	63	5.4	4.3e-6	5.3e-6	1.0e-6	Web graph
Wiki-Portuguese [27]	1.6	48.19	0.022	3,798	170	17,635	213	9.1e-7	9.7e-7	5.5e-8	Web graph
Wiki-Japanese [27]	1.6	69.82	0.021	3,863	180	15,595	223	6.9e-7	7.7e-7	8.0e-8	Web graph
Pokec [27]	1.6	27.31	0.046	97	2.08	3.8	2.5	1.2e-6	1.5e-6	2.3e-7	OSN
As-skitter [26]	1.6	13.08	0.005	86	16	341	18	1.5e-6	2.2e-6	6.9e-7	Internet topology
Wiki-Italian [27]	1.8	72.90	0.024	9,419	388	47,127	416	5.3e-7	5.8e-7	4.6e-8	Web graph
Wiki-En [27]	1.8	39.05	0.003	379	122.9	10,112	131	9.6e-7	1.2e-6	2.9e-7	Web graph
Hudong [27]	1.9	14.54	0.003	64	18.7	358	18	1.7e-6	2.0e-6	3.3e-7	Web graph
Hollywood [28], [29]	1.9	24.51	0.152	614	4	1.5	4	-4.4e-9	3.01e-7	3.06e-7	OSN
Baidu [27]	2.1	15.89	0.002	75	30.8	1,600	28	3.0e-6	3.6e-6	5.9e-7	Web graph
Flicker [27]	2.3	19.83	0.107	2,512	23	84	20	7.5e-8	4.4e-7	3.6e-7	OSN
Flixster [27]	2.5	6.27	0.013	23	1.7	0.88	1.3	6.0e-8	8.3e-7	7.7e-7	OSN
Wiki-Russian [27]	2.8	44.20	0.015	5,697	370	39,457	259	7.7e-7	8.2e-7	5.1e-8	Web graph
Wiki-Franch [27]	3.0	55.21	0.015	6,843	455	35,771	301	4.7e-7	5.2e-7	4.9e-8	Web graph
Orkut [27]	3.0	76.28	0.041	1,882	45	194	29	2.2e-7	3.0e-7	8.1e-8	OSN
Wiki-German [27]	3.2	40.77	0.0088	2,899	328	40,234	203	1.1e-6	1.2e-6	5.3e-8	Web graph
USpatent [27]	3.7	8.75	0.067	22	0.33	0.011	0.1	7.2e-8	5.2e-7	4.5e-7	Citation
LiveJournal [26]	3.9	17.35	0.125	533	4	3.1	2.1	5.0e-7	7.7e-7	2.7e-7	OSN
Orkut2 [28], [29]	11	56.80	0.0002	669	2,543	36,715	441	2.3e-8	6.6e-8	4.3e-8	OSN
DBpedia [27]	18	13.89	0.0016	986	583	19,199	63	9.7e-7	1.0e-6	5.6e-8	Web graph
Web-Arabic [28], [29]	22	48.70	0.031	110,686	3,531	86,644	310	1.5e-7	1.5e-7	4.7e-9	Web graph
Gsh-2015 [28], [29]	29	9.18	0.007	1,169	164	1,000	11	9.9e-7	1.1e-6	1.1e-7	Web graph
Twitter [27]	41	57.74	0.0008	104,474	123,435	5,659,930	5,927	1.8e-8	2.0e-8	1.4e-9	OSN
MicrosoftAc.G. [30]	46	22.61	0.015	1,734	115	203	4.9	7.1e-7	7.2e-7	1.07e-8	Citation
Friendster [27]	65	55.06	0.017	12,521	720	23	22	1.5e-9	4.3e-9	2.8e-9	OSN

directed graphs is ignored and self-edges are removed. The properties of the graphs are summarized in Table II. The codes along with the intermediate data are available on the website http://cs.uwindsor.ca/~etemadir/cbias. We used two servers each with 24 cores and 256 GB RAM to calculate ground truth for weeks for large graphs.

B. The Bias

First, we demonstrate the existence of bias using Fig. 3. In the plot, the observed bias is obtained by repeating the estimation for 5×10^4 times except for the very large datasets. X-axes are sampling probability p. We can see that the range of p varies from data to data. We do not set a fixed range of p because, for different data sizes, the required sampling probability is different to achieve the same accuracy. Larger data normally requires smaller sampling probability. Hence, we fix the RSE (Relative Standard Error) to be within the range of 0.1 to 0.4. Then, p is derived from RSE using the formula provided in [20].

We can use Eq. 15 to interpret our experiments. From the equation, we can tell that the bias depends on the sampling probability p. Thus, we can expect that the bias diminishes with the increase of sample size, as verified by all the datasets. When the graph is very large, p could be very small to achieve accurate estimation. For instance, in WebArabic, p is in the order of 10^{-5} to achieve 95% confidence interval

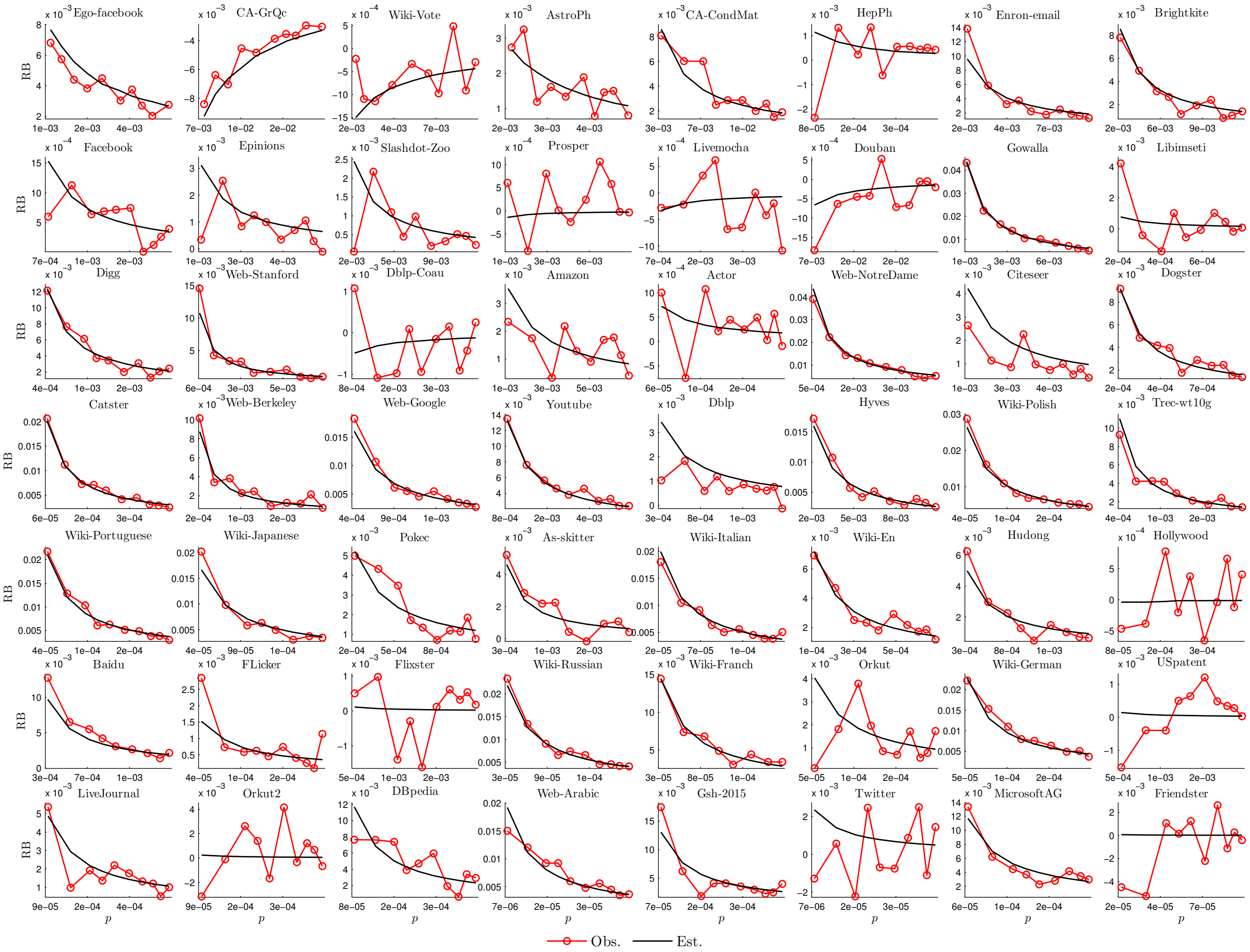


Figure 3: The observed vs. estimated RB of $\widehat{\mathcal{C}}$. The results were obtained over 5×10^4 independent runs for all graphs except for the large graphs in the last row with 10^4 independent repetitions. The estimated RBs are obtained using Eq. 16.

$\mathcal{C} \pm 0.1\mathcal{C}$. Secondly, the bias depends on the structure of the graph characterized by $r = \frac{2\Psi}{\Lambda^2} - \frac{\Omega}{\Lambda\Delta}$. Empirically, r is a small value that ranges from 10^{-5} to 10^{-9} among 56 graphs we explored.

To verify our estimated bias, we plot the estimated bias given in Eq. 16 side by side with the observed bias. Overall, the observed and estimated biases fit well. Observed bias fluctuates for some data sets, because of the low bias (hence high variation) of the estimations. For graphs with larger bias (e.g., RB$>1\%$), our equation fits the observed RB smoothly. This confirms that two approximations made during the derivation are valid, i.e. 1) it is good enough to take the quadratic expansion of the Taylor expansion; 2) It is valid to assume that $1 - p \approx 1$.

The most important result of this paper is the bias-corrected estimator $\widehat{\mathcal{C}}^+$. Fig. 4 compares the RB of $\widehat{\mathcal{C}}^+$ and $\widehat{\mathcal{C}}$. We can see that $\widehat{\mathcal{C}}^+$ corrects the bias consistently for all the datasets. For the same reason explained above, RBs fluctuates because the bias is very small, hence we see the large variation. For data sets where bias is large (above 1%), such as Gowalla, Web-Stanford, Web-NotreDame, Web-Google, and all the graphs from Wiki, RBs are more smooth.

Fig. 5 gives another perspective for understanding Eq. 15. This time we put 56 data sets in one plot, and demonstrate how good Eq. 15 is to quantify the bias. Panel (A) plots observed RB against the estimated ones. The observed RB is taken for anticipated RSE=0.2. We can see that observed RBs fit Eq. 15 well. It is not a perfectly straight line because the estimation varies for each run. There are a few data sets that have their relative biases larger than 1%. In most cases, the RB is very small value that is close to zero. In some cases, the bias is negative.

C. Positive and Negative Bias

We can observe that there are both positive and negative biases, although most of the datasets demonstrate positive bias. We should note that by Jensen's inequality, $\mathbb{E}(1/X) \geq 1/\mathbb{E}(X)$, thus we may have the wrong impression that there

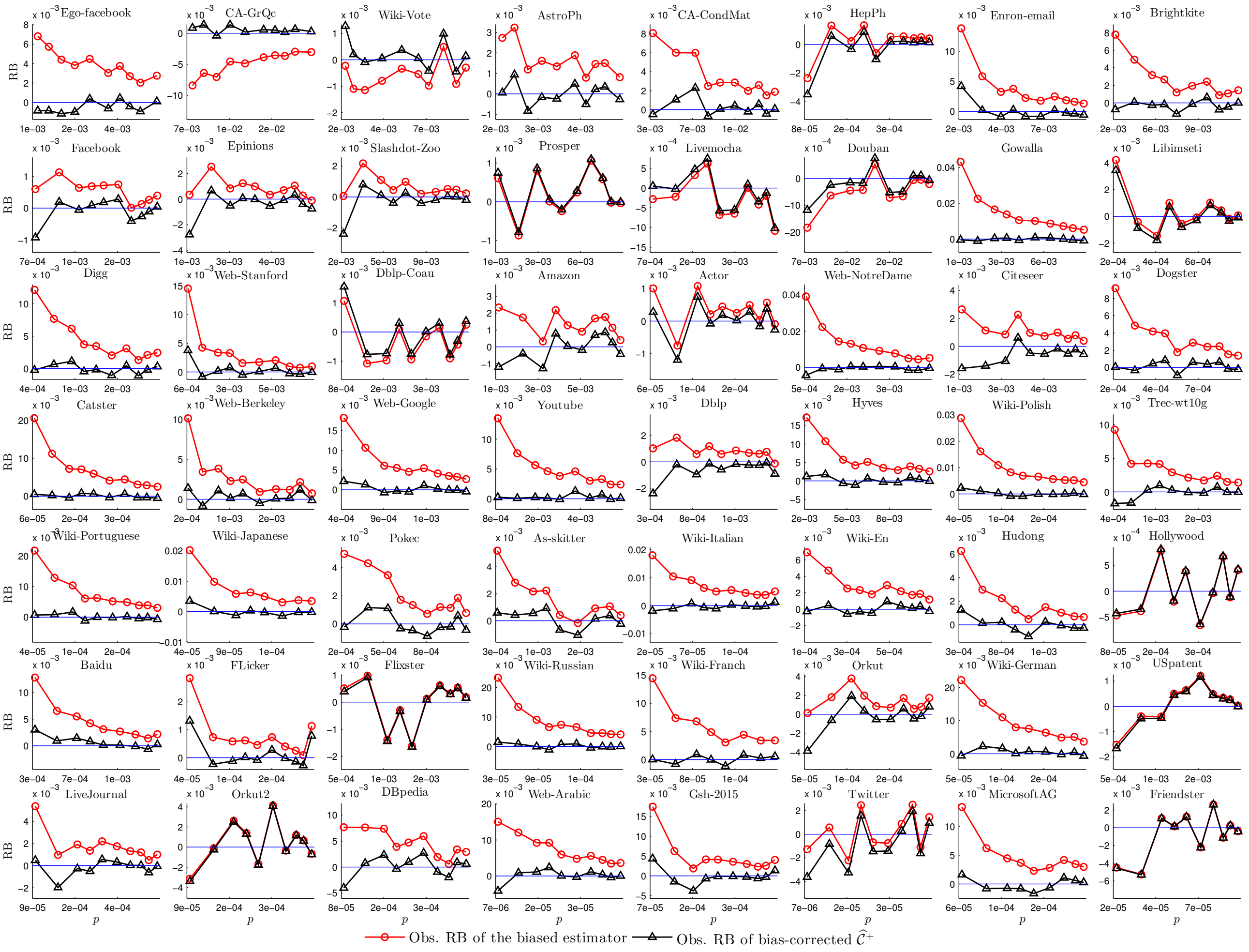


Figure 4: The bias-corrected estimator $\widehat{\mathcal{C}}^{+}$ vs. biased estimator. The observed RBs were obtained over 5×10^4 independent runs for all graphs except for the large graphs in the last row with 10^4 independent repetitions.

is positive bias only. However, Jensen's inequality can not be applied to our result because we are looking at the bias of Y/X, not $1/X$. Therefore, in our case, we can have both positive and negative biases.

Next, let's check when negative bias can occur. According to Eq. 15, the negative bias occurs when $\frac{2\Psi}{\Lambda^2} < \frac{\Omega}{\Lambda\Delta}$. In other words, it happens when Ω is large compared with other metrics. Empirically, it happens only for seven graphs among 56. These graphs are CA-GrQc, Wiki-vote, Prosper, Livemocha, Douban, DBLP-coau, Hollywood. All these graphs are (online) social networks with relatively high clustering coefficients.

Recall that terms $\frac{2\Psi}{\Lambda^2}$ and $\frac{\Omega}{\Lambda\Delta}$ are resulted from the first two terms of Taylor expansion. The relationship between these two terms decides not only the positive and negative bias, but also the size of the bias. When the two terms are close, the overall bias would be small. Hence, we use Fig. 5 Panel (B) to show the role of the first term. The figure plots observed bias against the first term. We can see that the first term can describe the RB for most data sets. However, there are some outliers. We plotted the labels for the first 12 graphs describe in Table II. Recall that we sort the graphs by their (node) size. Overall, we can see that the smaller the graph is, the farther away it deviates from the linear fit. In other words, for large graphs, we can use the first term $\frac{2\Psi}{\Lambda^2}$ to approximate the bias.

The next question is, if we focus on large graphs only, can we simplify $\frac{2\Psi}{\Lambda^2}$ further? Estimation is needed only for very large graphs. Hence the assumption on large graphs is valid. The values of Ψ and Λ lack intuitively interpretation. Even though it is easy to estimate them from a sample graph, it would be helpful to give a more intuitive understanding of the values as described in the next subsection.

D. *Characterizing Bias using Second and Third Moments*

When the graph is large, interestingly Λ and Ψ can be approximated by the second and third moments of the degrees of the graph. Recall that

$$\Lambda = \sum_{i=1}^{N} \Lambda_i = \sum_{i=1}^{N} \binom{d_i}{2}.$$

When the graph is large, d_i^2 dominates, and the above can be simplified as

$$\Lambda \approx 0.5N\langle d^2\rangle. \tag{21}$$

Similarly,

$$\begin{aligned}\Psi = &\sum_{(i,j)\in\mathcal{E}} \left[\binom{d_i-1}{2} + \binom{d_j-1}{2} + (d_i-1)(d_j-1) \right] \\ =&\sum_{i=1}^{N} d_i \binom{d_i-1}{2} + \sum_{(i,j)\in\mathcal{E}} (d_i-1)(d_j-1) \approx 0.5N\langle d^3\rangle.\end{aligned}$$

Therefore, we can approximate the bias by ignoring the second term:

$$RB \approx \frac{4\langle d^3\rangle}{pN\langle d^2\rangle^2}, \tag{22}$$

where $\langle d^2\rangle = \sum_{i=1}^{N} d_i^2/N$, and $\langle d^3\rangle = \sum_{i=1}^{N} d_i^3/N$.

Although this result is not rigorous, we can demonstrate that $\frac{2\langle d^3\rangle}{N\langle d^2\rangle^2}$ can approximate $\frac{\Psi}{\Lambda^2}$ well using Fig. 6. A visual inspection reveals that, among all 56 graphs including those small ones, all the data points are aligned well along the line. The Pearson correlation coefficient between those two is 0.99 for both logged and unlogged data points. We show the loglog plot only here. The unlogged version will have most data points cramped on the left lower corner due to the uneven distribution of those values.

This result tells us that the bias can be mostly determined by the third and second moments of degrees of the graph when the graph is large. We want to emphasize that when estimating clustering coefficient using $\widehat{\mathcal{C}}^+$, we only need to know the Ψ and Ω in the sample graph. There is no need to calculate the second and third moments for the entire graph. Eq. 22 is used only to understand the nature of the bias–the bias is large only when the third moment of the degree is large.

E. When the Bias Is Large

Practitioners need to know what kind of graphs may have a large bias. Our results point out that when the bias is large, Ψ is relatively large while Ω is relatively small. This happens for web graphs, such as a Stanford Web depicted in Fig. 7. The figure shows only a sampled graph obtained from a random walk. Yet it can reveal the overall structure of the graph: It contains a ball (s) that has a very high degree. Therefore, Ψ (or, equivalently, $\langle d^3\rangle$) is large. At the same, there are no triangles in the ball structure, Ω will not increase for this node. This explains why the web graphs often have higher bias, as indicated in Fig. 3.

IV. Related Work

Edge sampling has been used to estimate $\mathcal{C}$ in [12], [14] and [21]. Such methods approximate $\mathcal{C}$ using unbiased estimators for Δ and Λ, i.e., $\widehat{\mathcal{C}} = \widehat{\Delta}/\widehat{\Lambda}$. [12] samples edges with different probabilities, i.e., neighbors of sampled edges are chosen with higher probabilities, to increase the chance of identifying a (closed) wedge in subgraph g. Authors in [14], [21] use random edge technique to implement *wedge sampling*. Obviously, such estimators for $\mathcal{C}$ are biased, also mentioned in [12], [14], [21]. However, the authors have not provided an analytical and even experimental results for the bias of such estimators.

Random walk is another technique in this context [10], [11]. First, nodes are sampled uniformly at random proportional to their degrees, and metrics Δ and Λ are estimated based on the properties of sampled nodes. Then, $\mathcal{C}$ is estimated using $\widehat{\Delta}/\widehat{\Lambda}$. Unfortunately, such method results in a biased estimator for $\mathcal{C}$. Moreover, the performance of random walk based methods depends on the structure of the input graph and varies very from data to data.

A straightforward technique is *wedge sampling* [9]. It selects wedges uniformly at random. The fraction of closed wedges is used as an approximation for $\mathcal{C}$. Unfortunately, sampling a random wedge from large graphs is computationally expensive. Therefore, [31] used MapReduce to implement this method on large graphs. The extension of [9] with more experimental results can be found in [25], [32].

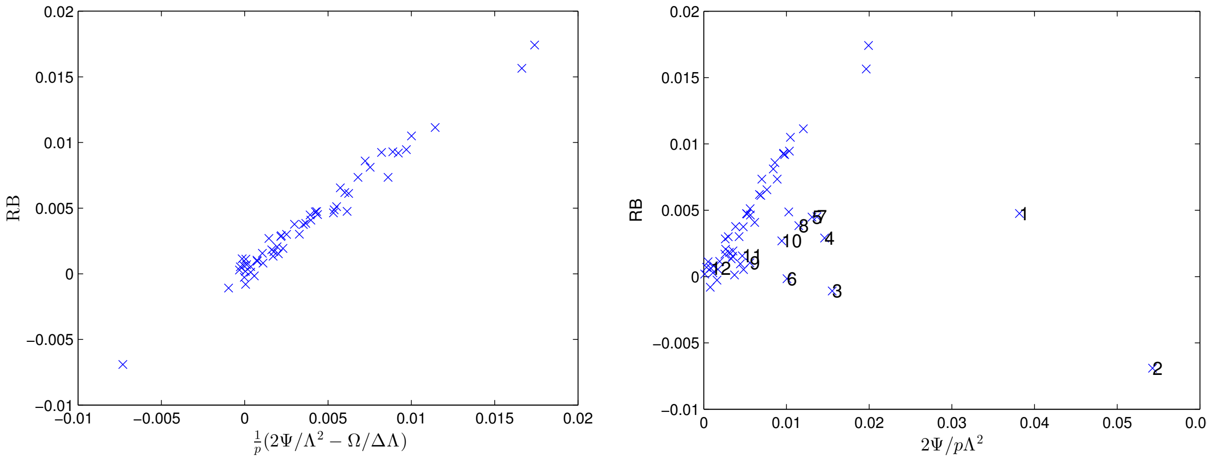
Another closely related direction is estimating the count of triangles using random edge [15], [16], [18]–[20], [33], random wedge [17], [25], and random walk [34] sampling. Obviously, to estimate $\mathcal{C}$, one also needs to approximate the number of wedges in the original graph.

V. Discussions and Conclusions

Bias is a perplexing problem in estimating graph properties in general [22] and clustering coefficient in particular [14]. It is difficult to observe because, for many graphs, especially small ones, the bias is almost negligible. Therefore, it has been taken for granted to use biased estimators by practitioners as well as researchers [10], [11]. It only became a more prominent problem recently, when people started to estimate very large graphs.

Bias for clustering coefficient estimation is difficult to quantify and correct [12], [21]. It involves two variables, i.e., Λ_g and Δ_g. We use Taylor expansion to approximate the bias, and show that quadratic expansion is good enough for the approximation. The quadratic expansion involves the variance and covariance of wedges and closed-wedges in the sample graph. We show that they can be quantified by Ψ and Ω and estimated by Ψ_g and Ω_g. Based on this result, we propose a bias-corrected estimator $\widehat{\mathcal{C}}^+$.

Bias for clustering coefficient estimation is difficult to understand. We observe positive and negative biases for



(A) RB vs. Two terms of Taylor expansion

(B) RB vs. First term of Taylor expansion

Figure 5: RB depends on the first term and second term of the Taylor expansion. The outliers are the 10 smallest graphs. Observed RBs are taken when RSE=0.2 over 10^5 independent runs.

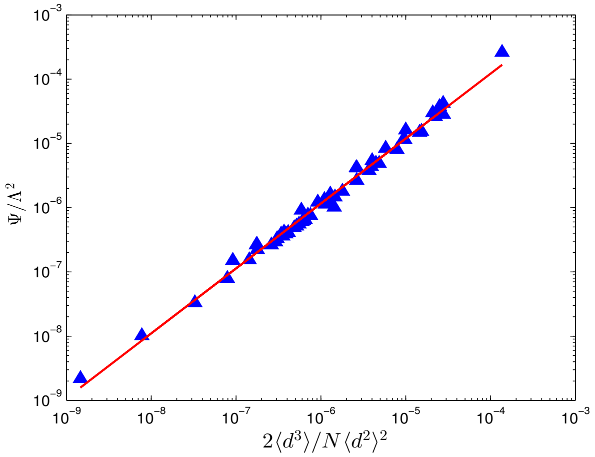


Figure 6: $\frac{\Psi}{\Lambda^2}$ against $\frac{2\langle d^3\rangle}{N\langle d^2\rangle^2}$ for 56 graphs.

Figure 7: A sample of the Stanford Web network.

different graphs. It can be as large as 4% for some graphs. On the other hand, it can be small even if the graph is very large. To understand negative bias, we find that the second term dominates the equation only when the network is relatively small, and they are (online) social networks. In other words, their clustering coefficient is high.

Our result is simple and elegant: first we quantify the bias as $RB \approx \frac{1}{p}\left(\frac{2\Psi}{\Lambda^2} - \frac{\Omega}{\Lambda\Delta}\right)$. This is much simpler than the original Taylor expansion because of our assumption that $1 - p \approx 1$. The assumption is valid when the graph is large. In most of the data sets we experimented with, p is typically in the order of 10^{-4} to achieve reasonable accuracy. Furthermore, we demonstrate that RB can be simplified further by ignoring the second term when the graph is large, i.e., $RB \approx \frac{2\Psi}{p\Lambda^2}$. Based on this, we can simplify the result further by approximate the bias using the second and third moments of the degrees of the graph, i.e., $RB \approx \frac{4\langle d^3\rangle}{pN\langle d^2\rangle^2}$. This is instrumental in helping us identify the type of graphs that have high bias.

Acknowledgements

The research is supported by CZI (Chan-Zuckerberg Initiative) and NSERC Discovery Grant.

References

[1] J. Leskovec and C. Faloutsos, “Sampling from large graphs,” in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’06. New York, NY, USA: ACM, 2006, pp. 631–636.

[2] J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica, “Graphx: Graph processing in a distributed dataflow framework,” in *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, 2014, pp. 599–613.

[3] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[4] H.-Y. Lam and D.-Y. Yeung, "A learning approach to spam detection based on social networks," in *4th Conference on Email and Anti-Spam (CEAS)*, 2007.

[5] M. C. Nascimento, "Community detection in networks via a spectral heuristic based on the clustering coefficient," *Discrete Applied Mathematics*, vol. 176, pp. 89–99, 2014.

[6] G. J. Gerhardt, N. Lemke, and G. Corso, "Network clustering coefficient approach to DNA sequence analysis," *Chaos, Solitons & Fractals*, vol. 28, no. 4, pp. 1037–1045, 2006.

[7] B. M. Tabak, M. Takami, J. M. Rocha, D. O. Cajueiro, and S. R. Souza, "Directed clustering coefficient as a measure of systemic risk in complex banking networks," *Physica A: Statistical Mechanics and its Applications*, vol. 394, pp. 211–216, 2014.

[8] J. Lu and H. Wang, "Variance reduction in large graph sampling," *Information Processing and Management*, vol. 50, no. 3, pp. 476–491, 2014.

[9] T. Schank and D. Wagner, "Approximating clustering coefficient and transitivity," *Journal of Graph Algorithms and Applications*, vol. 9, no. 2, pp. 265–275, 2005.

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

[11] L. Katzir and S. J. Hardiman, "Estimating clustering coefficients and size of social networks via random walk," *ACM Transactions on the Web (TWEB)*, vol. 9, no. 4, p. 19, 2015.

[12] N. K. Ahmed, N. Duffield, J. Neville, and R. Kompella, "Graph sample and hold: A framework for big-graph analytics," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining,* 1446-1455. ACM, 2014.

[13] X. Chen, Y. Li, P. Wang, and J. Lui, "A general framework for estimating graphlet statistics via random walk," in *Proceedings of the VLDB Endowment*, 2016, pp. 253–264.

[14] M. Jha, C. Seshadhri, and A. Pinar, "A space efficient streaming algorithm for triangle counting using the birthday paradox," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '13. New York, USA: ACM, 2013, pp. 589–597.

[15] H. Jowhari and M. Ghodsi, "New streaming algorithms for counting triangles in graphs," in *International Computing and Combinatorics Conference,* 710-716. Springer, 2005.

[16] L. S. Buriol, G. Frahling, S. Leonardi, A. Marchetti-Spaccamela, and C. Sohler, "Counting triangles in data streams," in *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems,* 253-262. ACM, 2006.

[17] A. McGregor, S. Vorotnikova, and H. T. Vu, "Better algorithms for counting triangles in data streams," in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, ser. PODS '16. New York, NY, USA: ACM, 2016, pp. 401–411.

[18] C. E. Tsourakakis, U. Kang, G. L. Miller, and C. Faloutsos, "Doulion: counting triangles in massive graphs with a coin," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining,* 837-846. ACM, 2009.

[19] Y. Lim and U. Kang, "Mascot: Memory-efficient and accurate sampling for counting local triangles in graph streams," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 685-694. ACM, 2015.

[20] R. Etemadi, J. Lu, and Y. H. Tsin, "Efficient estimation of triangles in very large graphs," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, ser. CIKM '16. ACM, 2016, pp. 1251–1260.

[21] M. Jha, C. Seshadhri, and A. Pinar, "A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox," *ACM Trans. Knowl. Discov. Data*, vol. 9, no. 3, pp. 15:1–15:21, Feb. 2015.

[22] J. Lu and D. Li, "Bias correction in a small sample from big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2658–2663, 2013.

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

[24] R. P. Stanley, "Differentiably finite power series," *European journal of combinatorics*, vol. 1, no. 2, pp. 175–188, 1980.

[25] C. Seshadhri, A. Pinar, and T. G. Kolda, "Triadic measures on graphs: The power of wedge sampling," in *SIAM International Conference on Data Mining (SDM)*. SIAM, 2013, pp. 10–18.

[26] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, Jun. 2014.

[27] J. Kunegis, "Konect - the koblenz network collection," http://konect.uni-koblenz.de/networks, May 2016.

[28] P. Boldi, M. Rosa, M. Santini, and S. Vigna, "Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks," in *Proceedings of the 20th international conference on WWW,* 587-596. ACM, 2011.

[29] P. Boldi and S. Vigna, "The WebGraph framework I: Compression techniques," in *Proceeding of the Thirteenth International World Wide Web Conference,* 595-601. ACM, 2004.

[30] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-j. P. Hsu, and K. Wang, "An overview of microsoft academic service (mas) and applications," in *Proceedings of the 24th International Conference on World Wide Web*. ACM, 2015, pp. 243–246.

[31] T. G. Kolda, A. Pinar, T. Plantenga, C. Seshadhri, and C. Task, "Counting triangles in massive graphs with mapreduce," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. S48–S77, 2014.

[32] C. Seshadhri, A. Pinar, and T. G. Kolda, "Wedge sampling for computing clustering coefficients and triangle counts on large graphs," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 7, no. 4, pp. 294–307, 2014.

[33] L. De Stefani, A. Epasto, M. Riondato, and E. Upfal, "*triest*: Counting local and global triangles in fully-dynamic streams with fixed memory size," in *Proceedings of the 22th ACM KDD international conference on Knowledge discovery and data mining*. ACM, 2016.

[34] M. Rahman and M. A. Hasan, "Sampling triples from restricted networks using mcmc strategy," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, ser. CIKM '14. New York, NY, USA: ACM, 2014, pp. 1519–1528.