# Efficient Estimation of Triangles in Very Large Graphs

Roohollah Etemadi, Jianguo Lu, Yung H. Tsin
School of Computer Science, University of Windsor, Canada
{etemadir, jlu, peter}@uwindsor.ca

## ABSTRACT

The number of triangles in a graph is an important metric for understanding the graph. It is also directly related to the clustering coefficient of a graph, which is one of the most important indicator for social networks. Counting the number of triangles is computationally expensive for very large graphs. Hence, estimation is necessary for large graphs, particularly for graphs that are hidden behind searchable interfaces where the graphs in their entirety are not available. For instance, user networks in Twitter and Facebook are not available for third parties to explore their properties directly.

This paper proposes a new method to estimate the number of triangles based on random edge sampling. It improves the traditional random edge sampling by probing the edges that have a higher probability of forming triangles. The method outperforms the traditional method consistently, and can be better by orders of magnitude when the graph is very large. The result is demonstrated on 20 graphs, including the largest graphs we can find. More importantly, we proved the improvement ratio, and verified our result on all the datasets. The analytical results are achieved by simplifying the variances of the estimators based on the assumption that the graph is very large. We believe that such big data assumption can lead to interesting results not only in triangle estimation, but also in other sampling problems.

## Keywords

Graph Sampling; Estimation; Triangles; Graph Algorithms; Clustering Coefficient

## 1. INTRODUCTION

Graphs are used to model interactions in many applications in online social networks, biology, biochemistry, and many other domains. The count of triangles in such graphs is an important structural property. For example, in online social networks, it is used to measure with what probability friends of friends are also friends (clustering coefficient [21, 28]). Counting triangles has also various applications such as spam detection [13] in computer networks, community detection and blog analysis [7, 20, 29] in social networks, protein identification [27], DNA sequence analysis [6] in biology, study of systemic risk [24], tracking the evolution of international trade [11] in economy, and more.

Enumerating triangles in massive graphs is not practical because the best-known algorithm has a time complexity of $O(M^{3/2})$, where $M$ is the number of edges [8, 14]. Thus, approximate algorithms are indispensable. Substantial work has been done on the streaming model where data items arrive sequentially and there is a limited memory window [1, 5, 9, 10, 16, 23]. Many streaming algorithms are designed specially to tackle such sampling restrictions. This paper focuses on a more generic sampling scheme without the streaming restriction. In addition for estimating triangles in large graphs, the method can also be applied in the scenario when a graph in its entirety is not available. For instance, many large networks, such as Twitter and Facebook user networks, are hidden behind searchable interfaces. Their properties can only be estimated by taking a sample from them.

When estimating the number of triangles, the most natural, and a naive one, is to take triples (three nodes) uniformly at random, then check whether they form triangles [2]. Unfortunately, this method is too costly to be of practical use. Most graphs, especially the large ones, are sparse. Hence, the vast majority of the triples have zero to two edges. It means that the cost of observing even one triangle in this method will be exorbitantly high. Buriol et al. ameliorate this problem by skipping the cases for zero edges [5]. They proposed to start with one random edge, then check whether there are triangles surrounding this edge. This method can be interpreted as starting with three random nodes, with the pre-condition that there needs to be at least one edge already in the triple.

When a random edge is given, there are numerous variations to check whether there is a containing triangle. Buriol et al. take a random node from the remaining set [5]; Tsourakakis et al. continue to select more random edges, in the hope to obtain a triangle [25]. The method proposed in [25] can be regarded as a random edge method: it selects random edges, forms a subgraph from the random edges. Then the count of the triangles in the subgraph is used to estimate the number of triangles in the original graph.

Both methods in [5] and [25] still suffer from the scarcity of triangles in the subgraph. In [5], although it skips the triples with zero edges, it could be better to skip triples

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

*CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA*

© 2016 ACM. ISBN 978-1-4503-4073-1/16/10…$15.00

DOI: http://dx.doi.org/10.1145/2983323.2983849

with one edge only, by starting with the triples that have at least two edges. For [25], triangle count in the subgraph can increase if we check their edges not only in the subgraph, but also in the original graph.

Motivated by these observations, this paper presents a *new sampling method* that combines the ideas from both [25] and [5]. The first step is the random edge sampling that is the same as that in [25]. Then, for every path of length two in the subgraph, we check the existence of the third edge in the original graph.

In this paper, we give the *unbiased estimator* and its variance for our sampling method. The variance is a long formula that involves several parameters, thereby it does not provide useful insight into the estimator, nor can it be compared with other sampling methods. Hence, we *simplify* the formula based on the assumption that the graph is very large. The simplified RSE (relative standard error) is $1/\sqrt{3\Delta|_g}$, where $\Delta|_g$ is the number of triangles restricted to the subgraph $g$. Intuitively, from the formula we can infer the 95% confidence interval by looking at the triangles in the subgraph.

After doing the similar treatment for the random edge method, we can *compare* the performance of these two estimators *analytically*. The analytical study demonstrates that our method is always better than the other method. This is confirmed by empirical experiments on 20 graphs, including the largest networks we can find.

Our contribution is twofold, in both the result and the method. For the result, we present a new estimator that outperforms the random edge method by orders of magnitude; For the method, we use the big data assumption to simplify the variances of various estimators. Thereby, performances of different triangle estimators can be compared analytically for the first time.

In presenting our theorems, we do not use the $\epsilon - \delta$ approximation notation as most other papers do, as it is self-evident from Chebyshev's inequality. What is more, Chebyshev's inequality is valid for any data distribution, hence it gives a loose range that has little practical implication. Estimates produced by multiple runs follow a normal distribution. This is implied by the central limit theorem and is verified by our experiments. The central limit theorem can be applied in this case because each estimation involves the summation (mean) of probabilities for all the triangles being sampled. With such normal distribution, we have a much tighter confidence interval, i.e., 95% confidence interval is within two standard deviations. Hence, in the remaining part of the paper only RSE and variance are discussed. A list of notations used in this paper is shown in Table 1.

Table 1: Summary of the notations.

| Notation | Meaning |
|---|---|
| $G(\mathcal{V}, \mathcal{E})$ | Original graph |
| $N, M$ | Number of nodes and edges in $G$ |
| $n$ | Sample size |
| $\langle d \rangle$ | Average degree |
| $\Delta$ | Number of triangles in $G$ |
| $K$ | Number of triangle pairs that share an edge |
| $g$ | A subgraph of $G$ |
| $\Delta_g$ | Number of triangles in $g$ |
| $\Delta\|_g$ | Number of triangles restricted in $g$ |
| $E_g$ | Random edge sampling method |
| $E_G$ | Our method that checks wedge closure in $G$ |
| $\widehat{\Delta}^{E_g}$ | The unbiased estimator for $E_g$ |
| $\widehat{\Delta}^{E_G}$ | Our unbiased estimator |

# 2. METHODS

## 2.1 Motivation

Given an undirected graph $G(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes, and $\mathcal{E}$ the set of edges. The graph is not a multi-graph and does not have self-loops. Let $N = |\mathcal{V}|$, $M = |\mathcal{E}|$, and $\Delta$ denote the number of triangles in $G$. A *wedge* $\mathcal{W}$ is a path $u-v-w$ of length two, where $u, v, w \in \mathcal{V}$, $u \neq w$, $(u, v) \in \mathcal{E}$, and $(v, w) \in \mathcal{E}$. A wedge $\mathcal{W}$ is closed if $(u, w) \in \mathcal{E}$. Otherwise it is open. Note that each triangle has three (closed) wedges.

Given a subgraph $g$ of $G$, we use $\Delta_g$ to denote the number of triangles in $g$, and $\Delta|_g$ the number of triangles restricted to the wedges in $g$, i.e., for every wedge $u - v - w$ in $g$, we check whether $(u, w) \in \mathcal{E}$. More formally,

$$\Delta|_g = \frac{1}{3}|\{(u, v, w)|\ (u, v), (v, w) \in g, (u, w) \in G\}|.$$

To estimate $\Delta$, a straightforward algorithm is the random edge sampling proposed by Tsourakakis et al. [26], which is called DOULION in [25], and called $E_g$ in this paper because it depends on the triangles in the sample graph $g$. The process is as follows: it selects random edges with an equal probability $p$ to generate a subgraph $g$. Then, the count of triangles in $g$ is used to approximate $\Delta$ in $G$ with the estimator

$$\widehat{\Delta}^{E_g} = \frac{\Delta_g}{p^3}. \tag{1}$$

A major drawback of the method is the scarcity of triangles in the sample graph. We can verify this by looking at the formula for the expected number of triangles in the sample graph $g$, which is

$$\mathbb{E}(\Delta_g) = \Delta p^3. \tag{2}$$

Because of the cubic function for a small $p$, we can barely see triangles in a sample graph. This problem is more acute when the graph is very large, henceforth the sampling probability is very small. In our subsequent experiments, $\Delta$ can be in the order of $10^{10}$, and $p$ is in the order of $10^{-5}$. In this scenario, it is obvious that it is far from observing any triangles in $g$, let alone enough number of triangles to guarantee the accuracy of estimation. It is necessary to devise a new sampling method that can increase the expected number of triangles in the sample.

## 2.2 Our method

The main idea of our method is to sample edges that have a higher probability of forming triangles. In social networks and other information networks, it is established that a friend of a friend has a higher probability of being friends as well [7, 28]. Thus, it would be beneficial to sample the edges for open wedges in a partially sampled graph. Following this rationale, our method divides the sampling into two steps. The first step is the same as a normal random edge sampling [25]: we take random edges with equal probability $p$. In the second step, in addition to counting the triangles in $g$, we also look at the open wedges in $g$, and check the closeness of these open wedges in the original graph. Since

the sampling method is changed, the estimator is no longer the one in the $E_g$ method. Instead, we give the estimator for $E_G$ as

$$\widehat{\Delta}^{E_G} = \frac{\Delta|_g}{p^2}, \tag{3}$$

which will be proved in the next section. Intuitively, we count the number of triangles that are restricted to $g$, then multiply it by a factor of $1/p^2$. Compared with the $E_g$ method, the number of observed triangles can be larger by a factor of $1/p$ under similar sampling cost.

EXAMPLE 1. *Figure 1 illustrates our sampling method. In this graph $G$, the number of triangles $\Delta = 3$. Suppose that the sampling probability $p = 0.5$, and six distinct edges are selected, resulting in a subgraph $g$ depicted in Panel (B). There is one triangle in $g$. Hence the estimate using the random edge method $E_g$ is*

$$\widehat{\Delta}^{E_g} = \frac{\Delta_g}{p^3} = \frac{1}{0.5^3} = 8. \tag{4}$$

*In our $E_G$ sampling, the first step is the same as $E_g$, i.e., six edges are selected with an equal probability $p = 0.5$. Then, there is an additional step to check the closeness of every open wedge. In the example, two wedges $3-2-1$ and $4-1-2$ are checked, and it is found that wedge $3-2-1$ is closed. Recall that there is already one triangle in the subgraph, which is equivalent to three closed wedges. Hence, all together there are four closed wedges, or $\Delta|_g = 4/3$. Note that in our sampling method, $\Delta|_g$ does not have to be an integer because it is $1/3$ of the closed wedges observed. The sampling cost is 8 because it checked 8 edges in total. The estimate is*

$$\widehat{\Delta}^{E_G} = \frac{\Delta|_g}{p^2} = \frac{4/3}{0.5^2} = \frac{16}{3}. \tag{5}$$

Our method applies extra checks in return for more triangles. One question is whether these additional triangles are worth the checking cost. Intuitively, the checking cost is proportional to $C$ (clustering coefficient), which measures the probability of seeing a triangle for an open wedge. If $w$ is the number of open wedges in $g$, we need to conduct closeness check $w$ times. There will be on average $w \times C$ number of additional triangles. In other words, $1 - C$ fraction of the checks are wasted. Note that for most networks, C is well above 0.01. On the other hand, the vast majority of the edges do not form triangles, especially when the graph is very large and the sample size is small. In those large graphs in our experiments, we need sample edges in the order of $10^5$ to form one triangle. Compared with this small success ratio, the cost of extra closure check is negligible. This argument is corroborated by our experiments depicted in Figure 5.

In the following, we derive the variance of this estimator, and compare it with that of $E_g$.

## 2.3 Variance of $E_G$

Let $w_i$ be an indicator for the $i^{th}$ closed wedge in the input graph $G$. The indicator $w_i$ is 1 when two edges in the $i^{th}$ closed wedge are sampled, otherwise it is 0. Since each triangle has three wedges, there are $3\Delta$ closed wedges. We label them from 1 to $3\Delta$. The number of triangles restricted to $g$ is

$$\Delta|_g = \frac{1}{3}\sum_{i=1}^{3\Delta} w_i. \tag{6}$$

For each wedge, the probability it is being sampled is $p^2$. The expected number of closed wedges in $G$ that are sampled in $g$ is:

$$3\mathbb{E}(\Delta|_g) = \mathbb{E}(\sum_{i=1}^{3\Delta} w_i) = \sum_{i=1}^{3\Delta}\mathbb{E}(w_i) = \sum_{i=1}^{3\Delta} p^2 = 3p^2\Delta.$$

Therefore, the unbiased estimator for $E_G$ sampling is

$$\widehat{\Delta}^{E_G} = \frac{\Delta|_g}{p^2}. \tag{7}$$

What is more important is the variance of the estimator. The variance is more complicated due to the covariance between wedges. Applying $var$ on the estimator, and expanding $\Delta|_g$ using Equation 6, we have

$$\begin{aligned} var(\hat{\Delta}^{E_G}) &= var\left(\frac{\Delta|_g}{p^2}\right) = var\left(\frac{1}{3}\sum_{i=1}^{3\Delta}\frac{1}{p^2}w_i\right) \\ &= \frac{1}{9p^4}\sum_{i=1}^{3\Delta}\sum_{j=1}^{3\Delta} cov(w_i, w_j) \\ &= \frac{1}{9p^4}\left(\sum_{i=1}^{3\Delta} var(w_i) + \sum_{i\neq j} cov(w_i, w_j)\right). \end{aligned} \tag{8}$$

Random variable $w_i$ follows a binomial distribution, whose variance is $p^2(1-p^2)$. The covariance of two independent variables $w_i$ and $w_j$ is zero. When $w_i$ and $w_j$ are dependent, they share one edge in common. When this happens, there are four cases as depicted in Figure 2. Their covariance is $cov(w_i, w_j) = \mathbb{E}(w_i w_j) - \mathbb{E}(w_i)\mathbb{E}(w_j) = p^3 - p^4$. Let $K$ denote the total number of pairs of triangles that share one edge in $G$. Considering that for each $cov(w_i, w_j)$ there is an equal $cov(w_j, w_i)$, $\sum_{i\neq j} cov(w_i, w_j) = 8K(p^3 - p^4)$. Therefore, we derive the following lemma:

LEMMA 1. *The variance of $\widehat{\Delta}^{E_G}$ is*

$$var(\widehat{\Delta}^{E_G}) = \frac{1}{9p^4}\left(3\Delta(p^2 - p^4) + 8K(p^3 - p^4)\right). \tag{9}$$

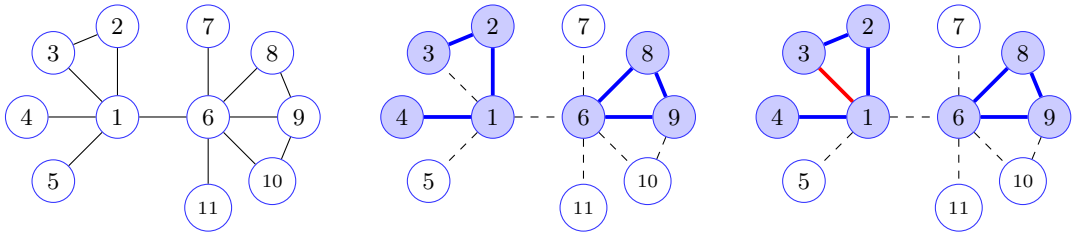
This result provides limited insight into the accuracy of estimator, because it is complex and depends on a few parameters including $p, K$, and $\Delta$. We can transform it into relative standard error $RSE = \sqrt{var}/\Delta$ as follows:

$$RSE(\widehat{\Delta}^{E_G}) = \left[\frac{1}{3\Delta|_g}\left(1 - p^2 + \frac{8K}{3\Delta}(p - p^2)\right)\right]^{\frac{1}{2}}. \tag{10}$$

When the sample size is small, i.e., when $p$ is small, we can see that the first term in Equation 10 plays a dominant role. Hence, RSE of the estimator can be approximated by the following

THEOREM 1. *When the sample size is small, RSE of the $E_G$ estimator can be approximated by*

$$RSE(\widehat{\Delta}^{E_G}) \approx \frac{1}{\sqrt{3\Delta|_g}}. \tag{11}$$




(A) Original graph (B) Step 1 (C) Step 2

Figure 1: Illustration of $E_g$ and $E_G$ sampling.

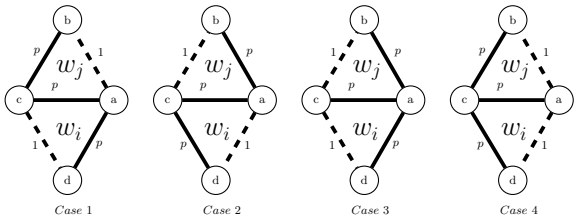



Figure 2: Dependent wedges of two shared triangles.

This result is useful for the comparison with the $E_g$ method that will be discussed in the next section. In addition to that, it gives us a practical guidance for conducting estimations. For example, if we want to have an estimation with 95% confidence interval of $\Delta \pm 0.1 \times \Delta$, then we need to have an RSE that is approximately $0.1/1.96 \approx 0.05$. According to Equation 11, the number of triangles we need to see is

$$\Delta|_g = \frac{1}{3 \times RSE^2} = \frac{1}{3 \times 0.05^2} = 133.$$

## 2.4 Variance of $E_g$

Although [25] gave the variance for the $E_g$ estimator, it is a long formula that buries intuitive interpretations. Similar to our previous treatment for the $E_G$ estimator, we transform the variance to RSE and simplified it into the following theorem:

THEOREM 2. *When the sample size is small, RSE of the $E_g$ estimator can be approximated by*

$$RSE(\widehat{\Delta}^{E_g}) \approx \frac{1}{\sqrt{\Delta_g}}. \tag{12}$$

PROOF. *See Appendix A.*

Next, we want to compare these two methods. One would be tempted to compare their RSE ratio given a fixed sampling percentage. This approach turns out not ideal because given a fixed sampling probability, $p$ is small for $E_g$ could be already a very large one for $E_G$. Therefore it violates our small sample assumption.

Hence, we compare their sample size to achieve the same RSE. Comparing Equations 11 and 12, we obtained the performance ratio between the two methods:

COROLLARY 1. *Let $n_{E_g}$ and $n_{E_G}$ be the number of sample edges of $E_g$ and $E_G$ respectively for achieving the same RSE in the two methods. A relation between $n_{E_g}$ and $n_{E_G}$ is:*

$$\frac{n_{E_g}}{n_{E_G}} \approx \left[\frac{3M}{n_{E_g}}\right]^{\frac{1}{2}}. \tag{13}$$

PROOF. *See Appendix B.*

Recall that $M$ is the number of edges in $G$, which is always larger than sample size $n_{E_G}$. Therefore, $E_g$ always needs more samples to achieve the same accuracy. When the sample size becomes bigger and approaches the total data size, the difference diminishes.

# 3. EXPERIMENTS

Our analytical results are derived with approximations based on the assumption on the data size and sample size. The experiments are designed to confirm the validity of the analytical results, and empirically demonstrate how much better our method is. In particular, analytical results do not include the cost of additional closeness check. These experiments confirm that such cost does not affect the overall performance of our method.

## 3.1 Datasets

We use 20 real world graphs to evaluate the algorithms, whose statistics are summarized in Table 2. We removed repeated edges and self-loops, and ignored the edge directionality in directed networks. Therefore, some statistics may be different from other papers working on the same datasets. For example, we found that the Twitter data contains 18% repeated edges. Such repeated edges have to be removed to guarantee the accuracy of the sampling.

We include almost all the largest graphs that we can find. Examples are the most recent academic citation graph released by Microsoft, which contains 46 million nodes, and the well-known Twitter user network that has 41 million nodes. In addition to these large graphs, we also include some smaller graphs of various scales for comparison purpose. The types of the graphs are also diversified, covering various areas. There are web graphs, online social networks, citation graphs, co-author and co-purchasing relations etc.

The experiments are conducted on two servers, each has 256 GB memory and 24 cores. The data and code are available on the website http://etemadir.myweb.cs.uwindsor.ca/cikm2016/triangles.php.

## 3.2 Experimental setup

We verify Theorems 1 and 2 by comparing observed RSEs obtained from running the estimators on the datasets and

Table 2: Properties of the networks in our experiments, sorted by graph size $N$.

| Data Set | N | $\langle d \rangle$ | $C$ | $\Delta(\times 10^6)$ | K $(\times 10^6)$ | R(K/$\Delta$) | Description |
|---|---|---|---|---|---|---|---|
| Ego-facebook [15] | 4,039 | 43.69 | 0.519 | 1.6 | 228 | 141.92 | Online social network (OSN) in Facebook |
| Enron-email [12] | 36,692 | 10.02 | 0.085 | 0.7 | 36 | 50.24 | Email communication network in Enron |
| Brightkite [15] | 58,228 | 7.35 | 0.110 | 0.49 | 29 | 59.14 | OSN in Brightkite |
| Dblp-Coau [15] | 317,080 | 6.62 | 0.306 | 2.2 | 105 | 47.22 | Co-authorship network in DBLP |
| Web-NotreDame [15] | 325,729 | 6.69 | 0.087 | 8.9 | 1,552 | 174.23 | Web graph of Notre Dame |
| Amazon [15] | 334,863 | 5.53 | 0.205 | 0.6 | 3 | 5.29 | Co-purchasing network from Amazon |
| Citeseer [12] | 384,413 | 9.03 | 0.049 | 1.3 | 15 | 11.63 | Citation network in Citeseer |
| Dogster [12] | 426,820 | 40.03 | 0.014 | 83 | 42,069 | 503.82 | OSN from dogster.com website |
| Web-Google [12] | 875,713 | 9.87 | 0.055 | 13 | 621 | 46.38 | Web graph from Google |
| Youtube [15] | 1,134,890 | 5.27 | 0.006 | 3 | 251 | 82.37 | OSN in Youtube |
| Dblp [12] | 1,314,050 | 8.16 | 0.170 | 12 | 436 | 35.84 | Co-authorship network in DBLP |
| As-skitter [15] | 1,696,415 | 13.08 | 0.005 | 28 | 20,522 | 713.34 | Internet connections from Skitter project |
| Flicker [12] | 2,302,925 | 19.83 | 0.107 | 837 | 613,838 | 732.85 | Online social network in Flicker |
| Orkut [12] | 3,072,441 | 76.28 | 0.041 | 627 | 67,098 | 106.91 | Online social network in Orkut |
| LiveJournal [15] | 3,997,962 | 17.35 | 0.125 | 177 | 39,492 | 222.09 | OSN in LiveJournal |
| Orkut2 [3, 4] | 11,514,053 | 56.80 | 0.0002 | 223 | 34,671 | 155.38 | OSN in Orkut |
| Web-Arabic [3, 4] | 22,743,881 | 48.70 | 0.031 | 36,895 | 112,260,907 | 3,042.68 | Web graph from Arabian countries |
| Twitter [12] | 41,652,230 | 57.74 | 0.0008 | 34,825 | 176,266,104 | 5,061.49 | OSN from Twitter |
| MicrosoftAc.G. [19] | 46,742,304 | 22.61 | 0.015 | 578 | 19,589 | 33.88 | Citation network from Microsoft Academic |
| Friendster [12] | 65,608,366 | 55.06 | 0.017 | 4,173 | 185,191 | 44.37 | OSN of website Friendster |

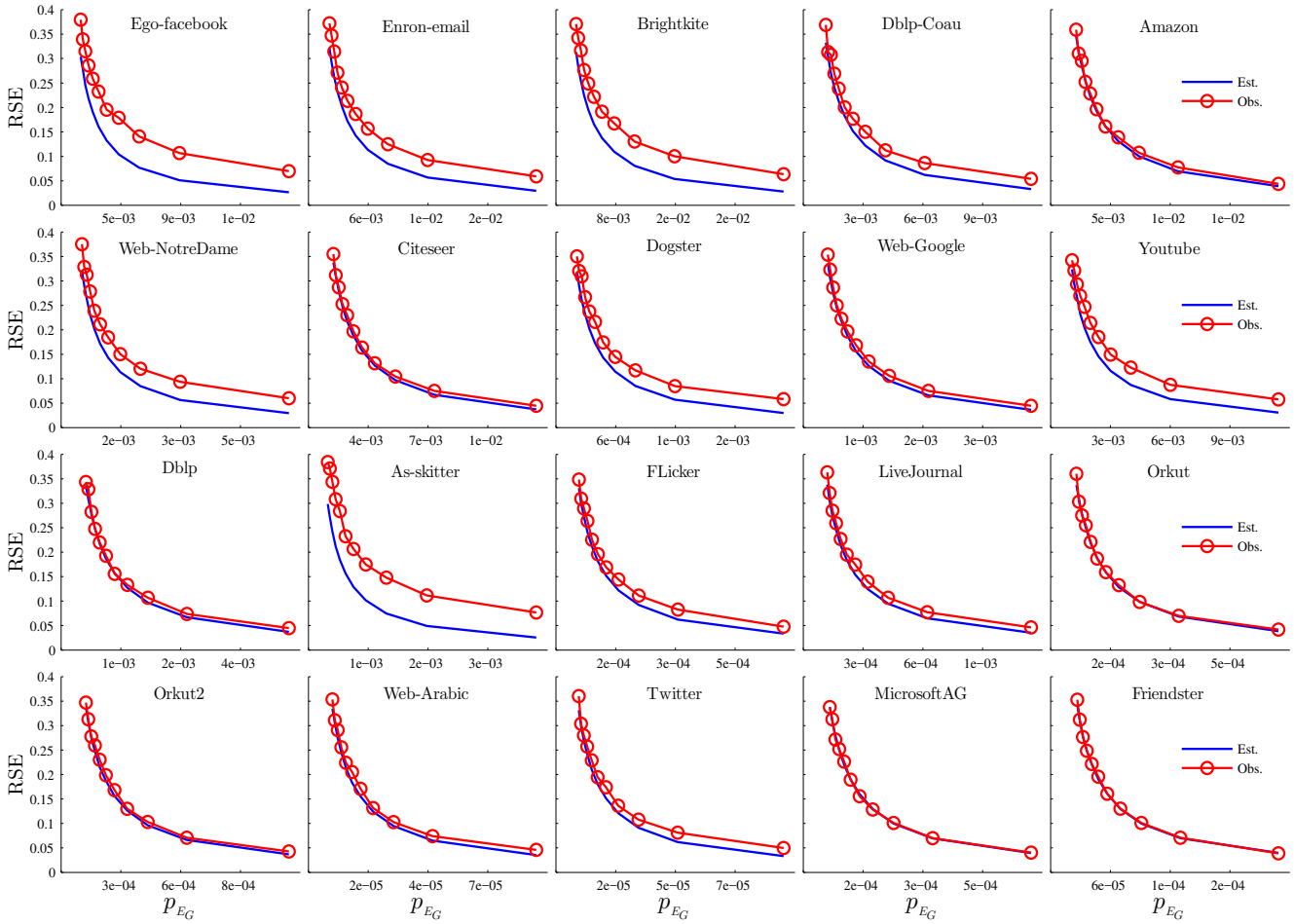



Figure 3: The observed and estimated RSEs of estimator $E_G$. Estimated values are obtained from Equation 11.

expected RSEs derived from Equations 11 and 12. To obtain observed RSEs, we repeated the estimation $k$ times using the same sample size, each time obtaining an estimate $\Delta_i$. Let $\mu = \frac{1}{k}\sum_{i=1}^{k}\Delta_i$. The observed RSE is calculated using

$$RSE = \frac{1}{\Delta}\sqrt{\frac{1}{k}\sum(\Delta_i - \mu)^2}. \quad (14)$$

In our experiments, $k = 1000$ for all graphs.

For the sample size parameter, most existing methods, such as DOULION [25], use a fixed range of sampling probability $p$ or percentages of the edges sampled for all graphs. Fixed percentage creates a wide variation for RSEs: one percent of sample data may not be enough to have an accurate estimate for small graphs, but can achieve very good (small) RSE for large graphs. Instead of fixed percentage, we target at a fixed range of RSEs, and choose the sample size that can create the RSEs at the desired range. RSE can reflect the confidence interval of the estimates, which is the main concern of any estimator.

All the experiments target RSEs between the range of 0.05 and 0.4. Note that this setting translates to 95% confidence intervals between $\Delta \pm 0.1\Delta$ and $\Delta \pm 0.8\Delta$. Next, we need to select sample size $n$ so that the observed RSE would be in that range. Recall that, from Equations 11 and 12, we can derive $\Delta_g$ and $\Delta|_g$ from desired RSEs. However, we still do not know what is the sample size $n$ to obtain that number of triangles. To solve this problem, we derived the following theorem to decide the sample size for $E_G$.

THEOREM 3. *In $E_G$ sampling, the relationship between $n_{E_G}$ and $\Delta|_g$ is*

$$n_{E_G} \approx \left\lceil \frac{3N\Delta|_g}{2C\Gamma} \right\rceil^{\frac{1}{2}}, \quad (15)$$

*where $C$ is the clustering coefficient, and $\Gamma = \delta^2 + 1$, where $\delta$ is the coefficient of degree variation.*

PROOF. *See Appendix C.*

We want to emphasize that we do not need to know those parameters such as $C$, $\Gamma$, $N$ to estimate the number of triangles. Equation 15 is used only in our experiment to select the sample size so that we know the results are within a certain RSE range. The triangle estimation itself only needs to know the sampling percentage and $\Delta|_g$.

Under certain circumstances, such as in sampling hidden data sources, the sampling percentage $p$ is not known since it cannot be derived from $n_{E_G}$. We do know $n_{E_G}$, but $p = n_{E_G}/M$ depends on $M$, the number of edges in the graph. Note that $M = N \times \langle d \rangle$. Both number of nodes $N$ and average degree $\langle d \rangle$ can be estimated effectively using random edge sampling. We refer to [17] for the estimation of $N$, and [18] for the estimation of average degree.

## 3.3 Verification of Theorems

Due to the approximations made in our derivations, we need to investigate the impacts of those approximations for different datasets.

**Theorem 1**. The observed RSEs and the projected RSEs are plotted in Figure 3 for the $E_G$ method. We can make several observations:

- For all the graphs, Equation 11 is a good approximation for the real RSE observed. The approximation is lower than the actual value, because we omitted the remaining terms in Equation 10;
- the approximation is more accurate when the data is large. Recall that the datasets are sorted in increasing order of data size;
- among the large graphs, Twitter and Web-Arabic demonstrate large deviation than other large graphs. A closer check reveals that they both have very large $K$'s, and large ratios between K and $\Delta$. According to Equation 10, the ratio $K/\Delta$ in the third term plays a key role on the impact of the approximation. Note that $K$ can be even larger than $\Delta$, because it is the number of combinations of triangles that share a common edge.

**Theorem 2**. Figure 4 shows the observed RSEs vs the approximated values derived from Theorem 2. Overall the approximation fits better with the real data than the $E_G$ method. This is expected because the major term we omitted in Equation 12 is smaller. It is

$$\frac{2K}{\Delta}p^2, \quad (16)$$

while the term omitted in Equation 11 is

$$\frac{8K}{3\Delta}p. \quad (17)$$

**Corollary 1**. Figure 5 demonstrates the major result of this paper: to what extent our method improves the random edge sampling method. In the experiment, we include the cost for checking wedge closures. Overall, our projected improvement ratio fits well with the observed data. Again, in large graphs the projection fits better with the real data.

Two additional observations are:

- The advantage of our method grows with the RSE (or decreases with the desired accuracy level). Our method is good when the sample size is small. When a large portion of the data is sampled, our method may not be as good as the random edge method. Recall that in all the derivations, we assume that the sampling probability $p$ is small. Despite such assumption, our method is consistently better than $E_g$ in all datasets. RSE ranges from 0.05 to 0.4. 0.05 is a reasonable RSE from which we can obtain 95% confidence interval between $\Delta \pm 0.1 \times \Delta$.
- The improvement grows with data size. When RSE= 0.05, it improves by a factor of four for Facebook, and a factor of 30 for Twitter. Figure 6 plots the improvement as a function of data size. It can be seen that the sample size ratio is correlated positively with the data size. The Pearson correlation coefficient is 0.94 for edge size, and 0.99 for triangle size, when both the data size and the improvement ratio are in logarithmic scale. The unlogged correlation coefficient is 0.75 for edge size and 0.82 for triangle size. We can see that the improvement correlates with $\Delta$ more strongly than $M$. This demonstrates that our method is especially good for large graphs with a large number of triangles.

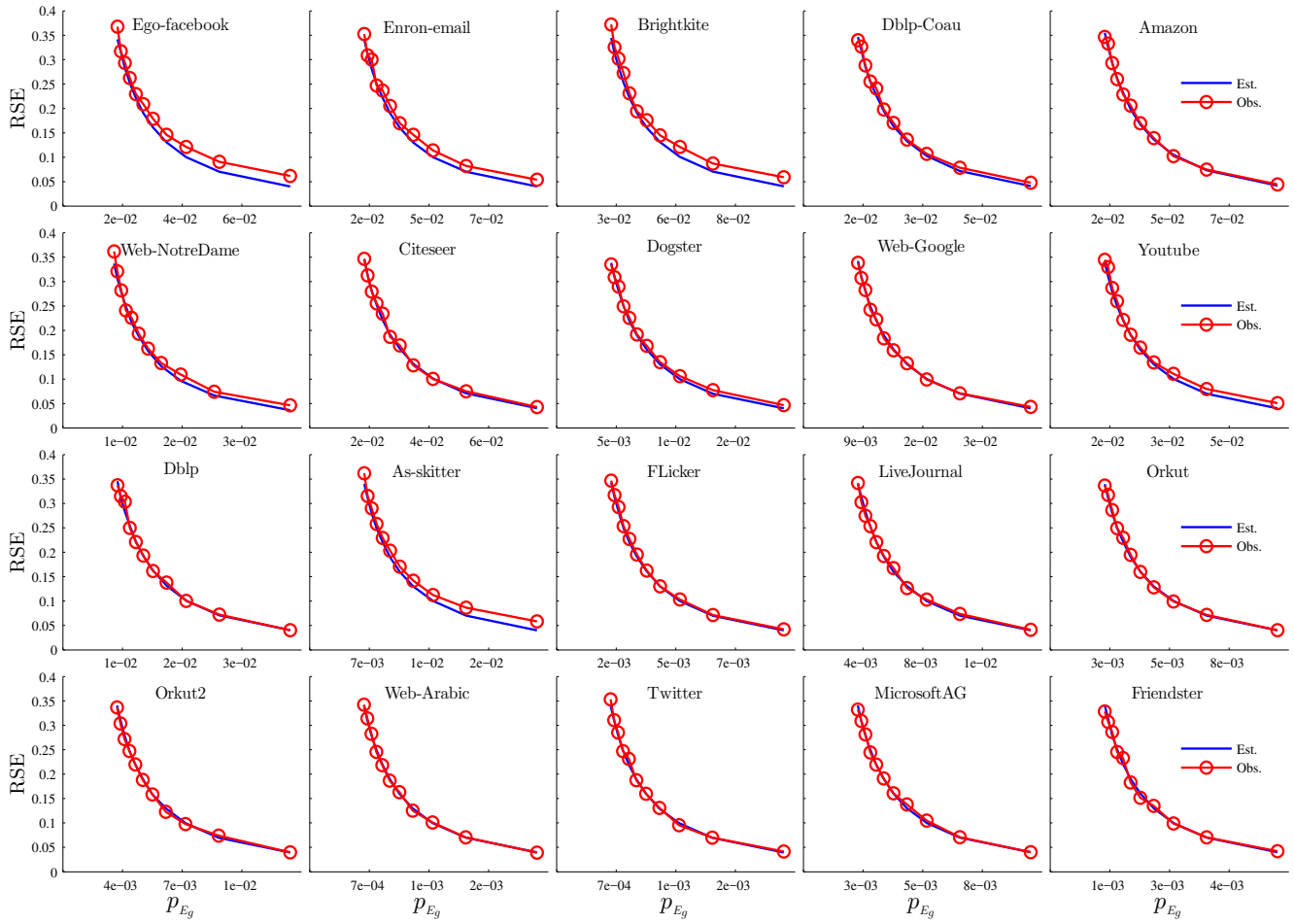



Figure 4: The observed and estimated RSEs of estimator $E_g$. Estimated values are based on Equation 12.

## 4. RELATED WORK

The first sampling based algorithm is proposed by Bar-Yossef et al in 2002 [2]. In one of their methods called the naive method, three random nodes are selected to form a triplet, and it is checked whether it is a triangle or not. The fraction of sampled triangles among selected triplets is used to estimate the counts of triangles in an input graph. The drawback of this method is that it is not easy to obtain a random triangle in sparse graphs. This method has been improved by decreasing the sample space from $\binom{N}{3}$ to $M(N-2)$ [5]. The idea is constructing a sampled triplet with a random edge $e$ and nodes from remaining ones.

In [23], Pavan et al introduced neighborhood sampling. This method first selects an edge uniformly at random. Then, one of its neighboring edges is sampled proportional to the degrees of its end nodes. A random wedge is created by two sampled edges.Then the method checks whether the wedge is closed or not.

Edge sampling is proposed by Tsourakakis et al in [25]. In this approach, edges are sampled uniformly at random, and the sampled edges form a subgraph. The count of triangles in the subgraph is used to estimate $\Delta$. In [25], the authors proved that the estimator is unbiased, and derived its variance.

To generate subgraph $g$ a different sampling method called triangle coloring is proposed in [22]. First, it colors nodes of an input graph uniformly at random with $N$ number of colors where $N = \frac{1}{p}$ and $p$ is a sampling probability. Then, each edge with the same color for its end-nodes is selected to generate $g$. This approach needs to record a label for each node of an input graph, and to scan all edges of the input graph. In [1], $g$ is generated by selecting random edges with different probabilities. This means that if an edge closes a wedge in $g$ it is selected unconditionally, and if it is adjacent to sampled edges so far, its sampling probability is $q$, otherwise it is $p$. In this method finding an appropriate value for $q$ is challenging since it depends on the input graph.

## 5. DISCUSSIONS AND CONCLUSIONS

This paper proposes a triangle estimation method that outperforms the previous one by a factor of up to 30 when RSE is 0.05. The improvement can be higher when RSE is bigger, or the required accuracy is reduced. We proved that the estimator is unbiased, and derived its variance. The variance in the original form lacks intuitive interpretation due to the long formula and multiple variables involved. Based on the big graph (henceforth small sample) assumption, we simplified the RSE to $(3\Delta|g)^{-1/2}$. Such simplified result

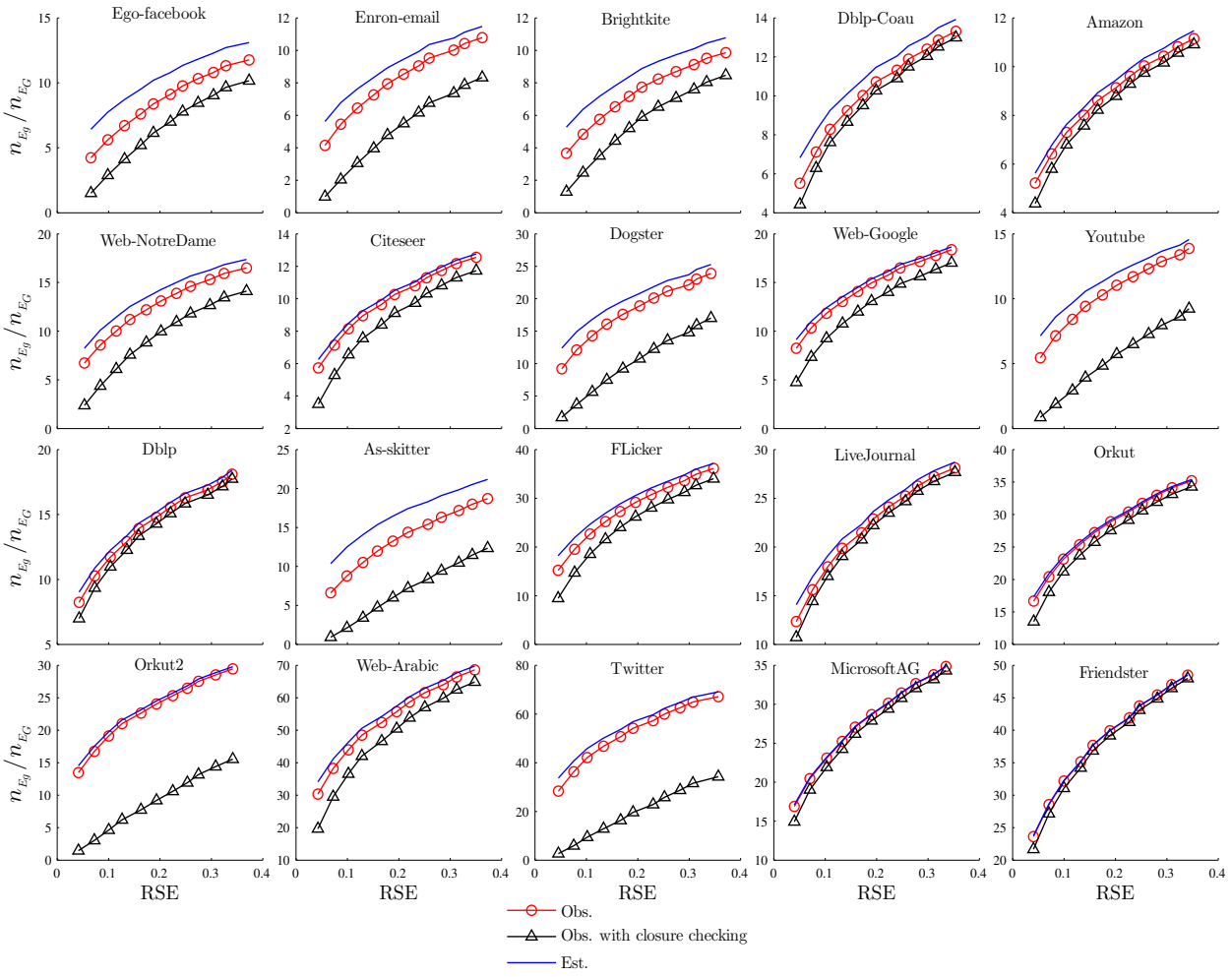



Figure 5: The observed and estimated ratio between the sample sizes of estimators $E_g$ and $E_G$ with the same RSEs, along with the ratio when the cost for closure check is included. Estimated values are obtained based on Equation 13.

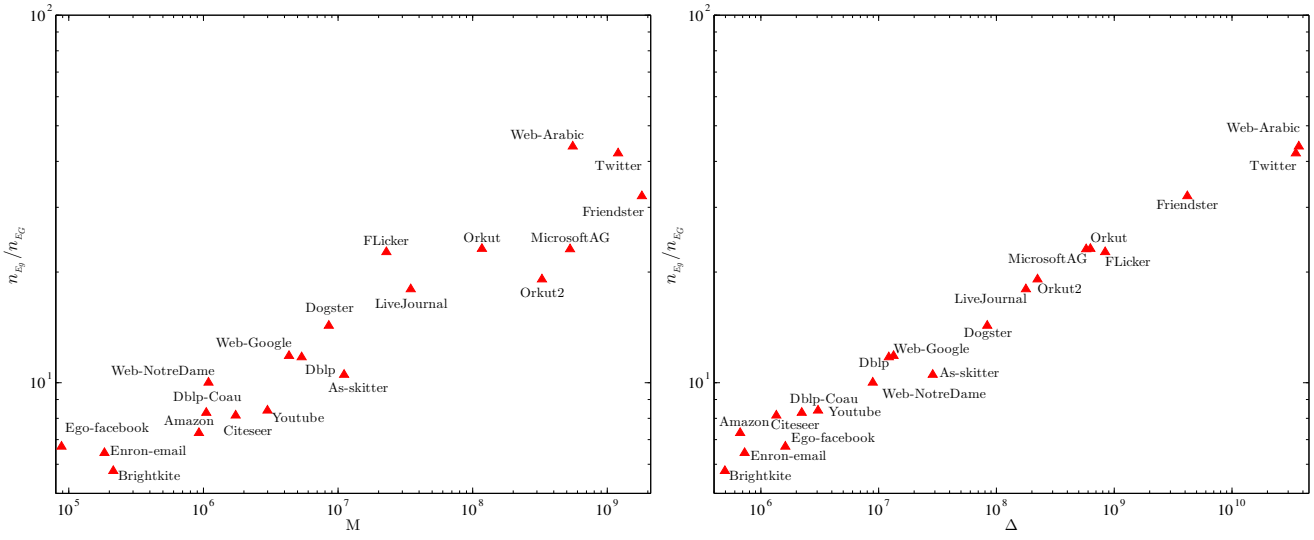



Figure 6: Improvement as a function of data size $M$ and $\Delta$, where RSE=0.1. The maximum and minimum ratios are 43.93 and 5.75 respectively.

gives us practical guidance in sampling. We can derive the confidence interval by looking at the triangles observed in the sample, hence we can decide when to stop the sampling. Although several assumptions are made for our conclusions, we empirically show that our approximation is still very close to the real RSEs observed.

The simplified formula also allows us to compare our method with other methods analytically. In the past, performances for different methods, including various streaming algorithms for triangle counting, are compared empirically. Those results may vary from data to data. Our work is the first to study the performance ratio analytically.

Our method is particularly suitable for very large graphs. It reduces the sample size by orders of magnitude for large graphs. We show that the performance improvement positively correlates with data size. When $\Delta$ and improvement ratio are logged, their Pearson correlation coefficient is as high as 0.99, almost a linear function for all 20 datasets.

The method is motivated by the scarcity of triangles in sampled graph when the original graph is very large. We can increase the probability of observing more triangles by checking the wedges in the sample graph. This strategy works very well for several reasons: First, most networks tend to cluster together, as friend's of friend's have a tendency to be friends as well. Thus, when checking the closeness of a wedge, it has a high probability that the wedge is closed; Secondly, checking the closeness of a wedge is more efficient than throwing a random edge in identifying a triangle. Throwing a random edge in a very large graph may well end up with an isolated edge, not even connecting with any other edge, let alone forming a triangle. On the other hand, checking the closeness of a wedge works at least in the vicinity of two connected edges. Its chance is higher to form a triangle when the sample size is small.

## 6. ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. We would also like to thank the support from NSERC (Natural Sciences and Engineering Research Council of Canada), the Cross Border Institute of the University of Windosr, and the University of Windsor for an GRF grant.

## 7. REFERENCES

[1] 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.

[2] Z. Bar-Yossef, R. Kumar, and D. Sivakumar. Reductions in streaming algorithms, with an application to counting triangles in graphs. In *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms,* 623-632. Society for Industrial and Applied Mathematics, 2002.

[3] 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 World Wide Web,* 587-596. ACM, 2011.

[4] 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.

[5] 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.

[6] G. J. Gerhardt, N. Lemke, and G. Corso. Network clustering coefficient approach to DNA sequence analysis. *Chaos, Solitons & Fractals*, 28(4):1037–1045, 2006.

[7] W. Han, X. Zhu, Z. Zhu, W. Chen, W. Zheng, and J. Lu. Weibo, and a tale of two worlds. *The IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 121–128, 2015.

[8] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM Journal on Computing*, 7(4):413–423, 1978.

[9] 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*, 9(3):15:1–15:21, Feb. 2015.

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

[11] T. Kastelle, J. Steen, and P. Liesch. Measurig globalisation: an evolutionary economic approach to tracking the evolution of international trade. In *DRUID Summer Conference on Knowledge, Innovation and Competitiveness: Dynamycs of Firms, Networks, Regions and Institutions-Copenhagen, Denemark, June,* 18-20., 2006.

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

[13] 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.

[14] M. Latapy. Main-memory triangle computations for very large (sparse (power-law)) graphs. *Theoretical Computer Science*, 407(1):458–473, 2008.

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

[16] 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.

[17] J. Lu and D. Li. Bias correction in a small sample from big data. *IEEE Transactions on Knowledge and Data Engineering*, 25(11):2658–2663, 2013.

[18] J. Lu and H. Wang. Variance reduction in large graph sampling. *Information Processing & Management*, 50(3):476–491, 2014.

[19] http://research.microsoft.com/en-us/projects/mag/.

[20] M. C. Nascimento. Community detection in networks via a spectral heuristic based on the clustering

coefficient. *Discrete Applied Mathematics*, 176:89–99, 2014.

[21] M. E. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences*, 99(suppl 1):2566–2572, 2002.

[22] R. Pagh and C. E. Tsourakakis. Colorful triangle counting and a mapreduce implementation. *Information Processing Letters*, 112(7):277–281, 2012.

[23] A. Pavan, K. Tangwongsan, S. Tirthapura, and K.-L. Wu. Counting and sampling triangles from a graph stream. *Proceedings of the VLDB*, 6(14):1870–1881, 2013.

[24] 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*, 394:211–216, 2014.

[25] 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.

[26] C. E. Tsourakakis, M. N. Kolountzakis, and G. L. Miller. Triangle sparsifiers. *J. Graph Algorithms Appl.*, 15(6):703–726, 2011.

[27] J. Wang, M. Li, H. Wang, and Y. Pan. Identification of essential proteins based on edge clustering coefficient. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(4):1070–1080, 2012.

[28] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, 1998.

[29] Y. Zhang and J. Lu. Discover millions of fake followers in weibo. *Social Network Analysis and Mining*, 6(1):1–15, 2016.

# APPENDIX

## A. PROOF OF THEOREM 2

Based on the variance of $\mathrm{E}_g$ [25], its RSE is as follows:

$$RSE(\widehat{\Delta}^{E_g}) = \left[\frac{1}{\Delta p^3}(1-p^3+\frac{2K}{\Delta}(p^2-p^3))\right]^{\frac{1}{2}} = \left[\frac{1}{\Delta_g}(1-p^3+\frac{2K}{\Delta}(p^2-p^3))\right]^{\frac{1}{2}}. \quad (18)$$

When the sampling probability $p$ is small, the terms $-p^3+\frac{2K}{\Delta}(p^2-p^3)$ is ignorable. Thus, the RSE of $\mathrm{E}_g$ is estimated by $1/\sqrt{\Delta_g}$.

## B. PROOF OF COROLLARY 1

PROOF. Let $p_{E_g}$ and $p_{E_G}$ be sampling probabilities of $\mathrm{E}_g$ and $\mathrm{E}_G$, respectively. We aim at getting the same RSE for both methods. Therefore, for small sample sizes the following equation holds

$$RSE(\widehat{\Delta}^{E_g}) = RSE(\widehat{\Delta}^{E_G})$$
$$\frac{1}{\sqrt{\Delta_g}} = \frac{1}{\sqrt{3\Delta|_g}}$$
$$\Delta_g = 3\Delta|_g. \quad (19)$$

Since $\Delta_g = \Delta p_{E_g}^3$ and $\Delta|_g = \Delta p_{E_G}^2$, we get

$$\Delta p_{E_g}^3 = 3\Delta p_{E_G}^2$$
$$p_{E_g}^3 = 3p_{E_G}^2. \quad (20)$$

By substituting $p_{E_g} = \frac{n_{E_g}}{M}$ and $p_{E_G} = \frac{n_{E_G}}{M}$, in Equation 20, the Corollary is proved.

## C. PROOF OF THEOREM 3

PROOF. Based on the estimator $\widehat{N}$ for graph node size $N$ [17], we obtain the relation between $N$ and the sample subgraph $g$ as follows

$$N = \frac{1}{w_g}\binom{n}{2}\Gamma. \quad (21)$$

Here, $n = 2 \times n_{E_g}$ is the total number of times the nodes have been sampled. It is the sum of all the degrees of the sample graph $g$, or number of edges times two. Variable $w_g$ is the number of collisions in [17], which can be interpreted as the number of wedges in $g$. After rearranging the above formula, we have:

$$n^2 - n = \frac{2Nw_g}{\Gamma}. \quad (22)$$

Recall that $w_g = 3\Delta|_g/C$, and approximate $n-1$ with $n$, we get

$$n \approx \left[\frac{6N\Delta|_g}{C\Gamma}\right]^{\frac{1}{2}}. \quad (23)$$

The approximation $n \approx n-1$ can be applied because $n$ is in the order of $\sqrt{N}$ so that collisions (wedges) can be observed in the sample graph. When $N$ is large, say in the order of $10^6$, $n$ is in the order of $10^3$. Substituting $n$ with $2 \times n_{E_G}$, we have:

$$n_{E_G} \approx \left[\frac{3N\Delta|_g}{2C\Gamma}\right]^{\frac{1}{2}}. \quad (24)$$