Tri-Fly: Distributed Estimation of Global and Local Triangle Counts in Graph Streams (Supplementary Document)

Kijung Shin[1], Mohammad Hammoud[2], Euiwoong Lee[1],
Jinoh Oh[3], Christos Faloutsos[1]

[1]Carnegie Mellon University, USA, {kijungs,euiwoonl,christos}@cs.cmu.edu
[2]Carnegie Mellon University in Qatar, Qatar, mhhamoud@cmu.edu
[3]Adobe Systems, USA, joh@adobe.com

Abstract. In this supplementary document, we provide accuracy analyses, complexity analyses, descriptions of datasets, and additional experimental results, all of which supplement the main paper [6].

A Detailed Bias and Variance Analyses

A.1 Bias Analysis (Proof of Theorem 1)

We provide a proof of Theorem 1 of the main paper.

Proof of Theorem 1. Consider a triangle $(u, v, w) \in \mathcal{T}^{(t)}$ and assume without loss of generality that $t_{vw} < t_{wu} < t_{uv} \leq t$. Let $d_i[uvw]$ be the contribution of (u, v, w) to each of $\bar{c}^{(t)}$, $c^{(t)}[u]$, $c^{(t)}[v]$, and $c^{(t)}[w]$ by each worker $i \in \mathcal{W}$. If we let $\mathcal{E}_i^{(t_{uv})}$ be the set of edges stored in worker i when edge (u, v) arrives, then by lines 7-12 and lines 21-22 of Algorithm 1,

$$d_i[uvw] = \begin{cases} 1/(|\mathcal{W}| \cdot p_i[uvw]) & \text{if } (v,w) \in \mathcal{E}_i^{(t_{uv})} \text{ and } (w,u) \in \mathcal{E}_i^{(t_{uv})} \\ 0 & \text{otherwise.} \end{cases}$$

By definition, $p_i[uvw]$ is the probability that both (v, w) and (w, u) are in $\mathcal{E}_i^{(t_{uv})}$. Therefore, $\mathbb{E}[d_i[uvw]] = 1/|\mathcal{W}|$. By linearity of expectation, the following equations hold:

$$\begin{aligned} \mathbb{E}[\bar{c}^{(t)}] &= \mathbb{E}\left[\sum_{i \in \mathcal{W}} \sum_{(u,v,w) \in \mathcal{T}^{(t)}} d_i[uvw]\right] = \sum_{i \in \mathcal{W}} \sum_{(u,v,w) \in \mathcal{T}^{(t)}} \mathbb{E}[d_i[uvw]] \\ &= \sum_{i \in \mathcal{W}} \sum_{(u,v,w) \in \mathcal{T}^{(t)}} \frac{1}{|\mathcal{W}|} = |\mathcal{T}^{(t)}|, \ \forall t \geq 1. \\ \mathbb{E}[c^{(t)}[u]] &= \mathbb{E}\left[\sum_{i \in \mathcal{W}} \sum_{(u,v,w) \in \mathcal{T}^{(t)}[u]} d_i[uvw]\right] = \sum_{i \in \mathcal{W}} \sum_{(u,v,w) \in \mathcal{T}^{(t)}[u]} \mathbb{E}[d_i[uvw]] \\ &= \sum_{i \in \mathcal{W}} \sum_{(u,v,w) \in \mathcal{T}^{(t)}[u]} \frac{1}{|\mathcal{W}|} = |\mathcal{T}^{(t)}[u]|, \ \forall v \in \mathcal{V}^{(t)}, \forall t \geq 1. \end{aligned}$$

Hence, the estimates given by Tri-Fly are unbiased. ■

(a) Type 1 Triangle Pair (b) Type 2 Triangle Pair

Fig. 4: Type 1 and Type 2 triangle pairs. In each triangle, the edge with double lines is the last edge to arrive.

A.2 Variance Analysis (Proof of Theorem 2)

We give a detailed variance analysis including a proof of Theorem 2 of the main paper. We first define the two types of triangle pairs illustrated in Figure 4.

Definition 1 (Type 1 Triangle Pair). *A Type 1 triangle pair is two different triangles* (u, v, w) *and* (u, v, x) *sharing an edge* (u, v) *satisfying* $t_{wu} = \max(t_{uv}, t_{vw}, t_{wu})$ *and* $t_{xu} = \max(t_{uv}, t_{vx}, t_{xu})$.

Definition 2 (Type 2 Triangle Pair). *A Type 2 triangle pair is two different triangles* (u, v, w) *and* (u, v, x) *sharing an edge* (u, v) *satisfying* $t_{vw} = \max(t_{uv}, t_{vw}, t_{wu})$ *and* $t_{xu} = \max(t_{uv}, t_{vx}, t_{xu})$.

Let $p^{(t)}$ and $q^{(t)}$ be the counts of Type 1 pairs and Type 2 pairs, respectively, in $\mathcal{G}^{(t)}$, which is the graph composed of the edges arriving at time t or earlier. Then, as in the main paper, we define $z^{(t)}$ as

$$z^{(t)} := \max\left(0, |\mathcal{T}^{(t)}|\left(\frac{(t-1)(t-2)}{k(k-1)} - 1\right) + \left(p^{(t)} + q^{(t)}\right)\frac{t-1-k}{k}\right),$$

which upper bounds the variance of the estimate $\bar{c}^{(t)}$ when a single worker is used, as formalized in Lemma 1.

Lemma 1 (Variance with a Single Worker). *Assume that a single worker is used in Algorithm 1 (i.e.,* $|\mathcal{W}| = 1$*). At any time* t*, the variance of the estimate* $\bar{c}^{(t)}$ *of the global triangle count* $|\mathcal{T}^{(t)}|$ *is upper bounded by* $z^{(t)}$*. That is,*

$$Var[\bar{c}^{(t)}] \leq z^{(t)}, \ \forall t \geq 1 \tag{3}$$

Proof Sketch. Tri-Fly with a single worker is equivalent to $\text{Triest}_{\text{impr}}$ [1]. Eq. (3) follows from Theorem 4.13 in [1], where the variance of the estimate of the global triangle count in $\text{Triest}_{\text{impr}}$ is upper bounded. ■

Intuition behind $z^{(t)}$ and Lemma 1. The estimate $\bar{c}^{(t)}$ increases whenever a triangle is discovered. That is, $\bar{c}^{(t)}$ is the sum of the contributions of the triangles in $\mathcal{T}^{(t)}$, where the contribution of a triangle is zero if it is not discovered. However, the variance of $\bar{c}^{(t)}$ is not simply the sum of the variances of the contributions since events that certain triangles are discovered are not independent.

Especially, the variance of $\bar{c}^{(t)}$ increases additionally for each Type 1 or Type 2 pair, where the discoveries of the two triangles are positively dependent. In $z^{(t)}$, the term $\left(\frac{(t-1)(t-2)}{k(k-1)} - 1\right)$ upper bounds the variance of the contribution of each triangle, and the term $\frac{t-1-k}{k}$ upper bounds the increase in the variance by each Type 1 or Type 2 pair. Thus, $z^{(t)}$ itself upper bounds the variance of $\bar{c}^{(t)}$.

The upper bound of the variance of the estimate $\bar{c}^{(t)}$ in Tri-Fly decreases inversely proportional to the number of workers, as formalized in Theorem 2 of the main paper. We give a proof of Theorem 2.

Proof of Theorem 2. Let $\bar{c}_i^{(t)}$ be the global triangle count sent from each worker i by time t. Then, by line 21 of Algorithm 1, $\bar{c}^{(t)} = \sum_{i \in \mathcal{W}} \bar{c}_i^{(t)} / |\mathcal{W}|$. Since $\bar{c}_i^{(t)}$ of each worker $i \in \mathcal{W}$ is independent of those of the other workers,

$$\begin{aligned} Var[\bar{c}^{(t)}] &= \sum\nolimits_{i \in \mathcal{W}} Var[\bar{c}_i^{(t)} / |\mathcal{W}|] = \sum\nolimits_{i \in \mathcal{W}} Var[\bar{c}_i^{(t)}] / |\mathcal{W}|^2 \\ &\leq |\mathcal{W}| \cdot z^{(t)} / |\mathcal{W}|^2 = z^{(t)} / |\mathcal{W}|, \end{aligned}$$

where the inequality follows from Lemma 1 (i.e., for each worker $i \in \mathcal{W}$, $Var[\bar{c}_i^{(t)}] \leq z^{(t)}$). Hence, Eq. (2) of the main paper holds. ■

B Time and Space Complexity Analyses

We discuss the time and space complexities of Tri-Fly.

B.1 Time Complexity Analysis

The time complexity of Tri-Fly for processing the first t edges in the input stream (i.e., $\{e^{(1)}, ...e^{(t)}\}$) is summarized in Table 3 of the main paper. The masters take $O(t \cdot |\mathcal{W}|)$ in total since each edge is broadcast to $|\mathcal{W}|$ workers. Each worker takes $O(t \cdot \min(t, k))$, and thus the workers take $O(|\mathcal{W}| \cdot t \cdot \min(t, k))$ in total, as formalized in Lemma 2 and Theorem 3.

Lemma 2. *Processing edge $e^{(s)}$ by each worker (lines 7-16 of Algorithm 1) takes $O(\min(s, k))$.*

Proof. The most expensive step of processing $e^{(s)} = (u, v)$ in Algorithm 1 is to find the common neighbors of nodes u and v (line 8 of Algorithm 1). Let $\mathcal{E}_i^{(s)}$ be the set of edges stored in worker i when edge $e^{(s)}$ arrives. Then, for each worker $i \in \mathcal{W}$, computing $\mathcal{N}_i[u] \cap \mathcal{N}_i[v]$ requires accessing $|\mathcal{N}_i[u]| + |\mathcal{N}_i[v]| = O(|\mathcal{E}_i^{(s)}|) = O(\min(s, k))$ edges. ■

Theorem 3 (Time Complexity of Workers in Tri-Fly). *In Algorithm 1, the time complexity of each worker for processing the first t edges in the input stream is $O(t \cdot \min(t, k))$.*

Proof. From Lemma 2, each worker takes $O(\min(s,k))$ to process each edge $e^{(s)}$. Thus, each worker takes $O\left(\sum_{s=1}^{t}\min(s,k)\right)$ to process t edges. From

$$\sum_{s=1}^{t}\min(s,k) \leq \sum_{s=1}^{t}\min(t,k) = t\cdot\min(t,k),$$

processing the first t edges by each worker takes $O(t\cdot\min(t,k))$. ■

Lastly, the aggregators take $O(|\mathcal{W}|\cdot t\cdot\min(t,k))$ in total, as formalized in Theorem 4.

Theorem 4 (Time Complexity of Aggregators in Tri-Fly). *In Algorithm 1, the time complexity of the aggregators for processing the first t edges in the input stream is $O(|\mathcal{W}|\cdot t\cdot\min(t,k))$ in total.*

Proof. While processing edge $e^{(s)} = (u,v)$ (lines 7-16 of Algorithm 1), each worker sends less than $\min(s,k)+3$ counts to the aggregators in total because $|\mathcal{N}_i[u]\cap\mathcal{N}_i[v]| + 3 < |\mathcal{E}_i^{(s)}| + 3 = \min(s,k)+3$. Thus, while processing the first t edges, the workers send less than $\sum_{i\in\mathcal{W}}\sum_{s=1}^{t}(\min(s,k)+3)$ counts to the aggregators in total. Since processing each count takes $O(1)$, the time complexity of the aggregators is $O(\sum_{i\in\mathcal{W}}\sum_{s=1}^{t}(\min(s,k)+3))$ in total. Since

$$\begin{aligned}\sum_{i\in\mathcal{W}}\sum_{s=1}^{t}(\min(s,k)+3) &< \sum_{i\in\mathcal{W}}(t\cdot\min(t,k)+3t)\\ &= |\mathcal{W}|\cdot(t\cdot\min(t,k)+3t) = O(|\mathcal{W}|\cdot t\cdot\min(t,k)),\end{aligned}$$

Theorem 4 holds. ■

Notice that, with a fixed storage budget k, the time complexity of Tri-Fly is linear in the number of edges in the input stream, as confirmed empirically in Section 5.2 of the main paper and Section D.

B.2 Space Complexity Analysis

The space complexity of Tri-Fly for processing the first t edges in the input stream (i.e., $\{e^{(1)},...e^{(t)}\}$) is summarized in Table 3 of the main paper. Each master requires $O(1)$ space since it simply broadcasts the received edges. Thus, the masters require $O(|\mathcal{M}|)$ space in total. Since each worker requires $O(\min(t,k))$ space for sampled edges, the workers require $O(|\mathcal{W}|\cdot\min(t,k))$ space in total. The aggregators require $O(|\mathcal{V}^{(t)}|)$ space in total to maintain 1 estimate for the global triangle count and $|\mathcal{V}^{(t)}|$ estimates for the local triangle counts.

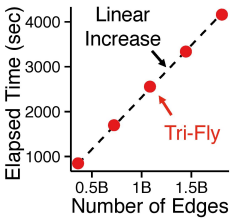

Fig. 5: **Tri-Fly scales linearly with the size of the input stream.** The graph streams were created by sampling different numbers of edges from the Friendster dataset.

C Descriptions of Datasets

We give the descriptions of the real-world datasets used in our experiments. The sizes of the datasets are summarized in Table 4 of the main paper.

- BerkStan [3]: A hyperlink network between the web pages from UC Berkeley and Stanford University.
- Patent [2]: A citation network between U.S. patents.
- Flickr [4]: A friendship network between the members of Flickr, an image and video hosting website.
- FriendSter [7]: A friendship network between the members of Friendster, a social gaming website.

D Scalability in Graph Streams with Realistic Structures

To show the linear scalability of Tri-Fly in graph streams with realistic structures, we measured its running times in input streams created by sampling different numbers of edges from the Friendster dataset. The experimental settings were the same with those in Section 5.2 of the main paper, and thus we fixed storage budget k to 10^7. As seen in Figure 5, Tri-Fly scaled linearly with the size of the input stream, as expected from Theorems 3-4 in Section B.1.

E Related Work on Triangle Counting in a Graph Stream with Multiple Sources

Using multiple machines was discussed theoretically by Pavan et al. [5] for triangle counting in a graph stream with multiple sources. The authors aimed to minimize communication cost while giving the same estimation as if edges are streamed from one source. In their algorithm, the number of machines is determined by the number of sources, and using more machines makes the algorithm neither faster nor more accurate (see Theorem 5.2 of [5]). Thus, their goal is clearly different from ours, which is to utilize multiple machines for fast and accurate estimation.

References

1. De Stefani, L., Epasto, A., Riondato, M., Upfal, E.: Triest: Counting local and global triangles in fully-dynamic streams with fixed memory size. In: KDD (2016)
2. Hall, B.H., Jaffe, A.B., Trajtenberg, M.: The nber patent citation data file: Lessons, insights and methodological tools. Tech. rep., National Bureau of Economic Research (2001)
3. Leskovec, J., Lang, K.J., Dasgupta, A., Mahoney, M.W.: Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. Internet Mathematics 6(1), 29–123 (2009)
4. Mislove, A., Marcon, M., Gummadi, K.P., Druschel, P., Bhattacharjee, B.: Measurement and analysis of online social networks. In: IMC (2007)
5. Pavan, A., Tangwongan, K., Tirthapura, S.: Parallel and distributed triangle counting on graph streams. Technical report, IBM, Tech. Rep. (2013)
6. Shin, K., Hammoud, M., Lee, E., Oh, J., Faloutsos, C.: Tri-fly: Distributed estimation of global and local triangle counts in graph streams. In: PAKDD (2018)
7. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. Knowl. Inf. Syst. 42(1), 181–213 (2015)