TRIÈST: Counting Local and Global Triangles in Fully-Dynamic Streams with Fixed Memory Size

Lorenzo De Stefani (Brown University), Alessandro Epasto (Google Research NY), Matteo Riondato (Two Sigma Investments, LP), Eli Upfal (Brown University)

lorenzo@cs.brown.edu, aepasto@google.com, matteo@twosigma.com, eli@cs.brown.edu

MOTIVATION

Social networks are constantly evolving

- 1500 Facebook friend requests / sec.
- 2000 FB messages/sec.
- 1000 new tweets per second on Twitter

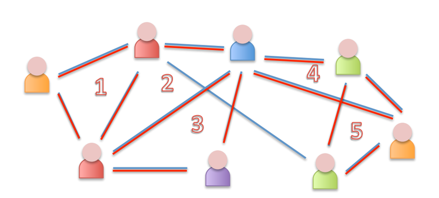
Properties of real graphs are **inherently volatile** and we need efficient algorithms that **keep track** of **fast changing properties** over **massive graphs** with billions of edges.

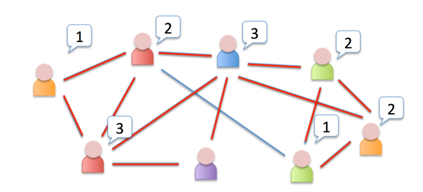
Key implications:

- Re-running the algorithm from scratch at every update is infeasible;
- Approximations provide sufficient information;
- No knowledge of the size of the stream…
- …there is no end of the stream, so no post-processing *at the end of the stream* is possible.

TRIANGLE COUNTING

Fundamental primitive in data mining.

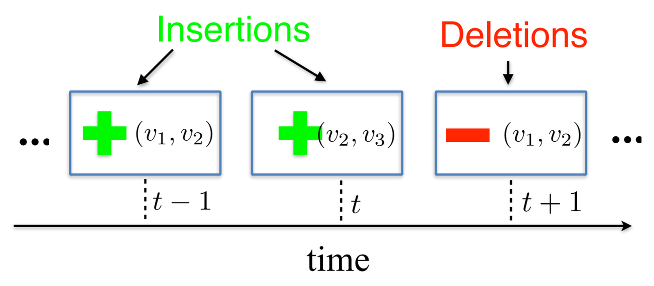

Global triangles


Local triangles for each vertex

Many applications: anomaly detection, event detection, link prediction, community detection.

MODEL

- Start from an empty graph;
- Time t discretized according to edge updates;
- Arbitrarily long sequence of edge updates…
- … in adversarial order.



OUR CONTRIBUTION

TRIÈST (TRIangle ESTimation)

A suite of three algorithms for triangles counting using reservoir sampling:

- TRIÈST-BASE: baseline algorithm for insertion only streams;
- TRIÈST-IMPR: improved algorithm for insertion only streams;
- TRIÈST-FD: algorithm for fully-dynamic graph streams.

For all algorithms:

- Single pass over the data;
- Pre-specified memory space;
- Full utilization of available memory;
- Unbiased estimators;
- Complete analysis of the variance;
- Concentration bounds.

SAMPLING STRATEGY

- TRIÈST builds on reservoir sampling to utilize all the available memory of fixed size *M*;
- At any time, the $K \leq M$ edges in memory are chosen uniformly at random from all subsets of *K* edges in the graph seen so far;
- The edges maintained in memory constitute a sample subgraph of the entire network;
- TRIÈST keeps counters *T* and *T(u)* for the number of global local triangles in the sample subgraph;
- The counters are used to obtain unbiased estimators for the number of global Δ and local $\Delta(v)$ triangles in the entire network.

TRIÈST FOR INSERTION ONLY STREAMS

TRIÈST-BASE

Using Reservoir Sampling TRIÈST-BASE maintains a sample of size *M* of the edges in the stream:

1. For $t \leq M$: add the *t*-th edge to the sample *S;*
2. For $t > M$, with prob. M/t add the *t*-th edge to *S* and discard an edge selected uniformly at random from *S;*

- *S* is a uniform sample of fixed size = *M;*
- Each time the sample *S* is updated, we update the global counter *T* (local *T(u)*) of the number of triangles in the sample subgraph;
- In order to count a triangle, all its edges need to be in the sample subgraph. The probability of this event is:

$$p_\Delta = \binom{M}{3} \Big/ \binom{t}{3}$$

- Global estimator: $\tau = T/p_\Delta$
- Local estimators: $\tau(u) = T(u)/p_\Delta$

TRIÈST-IMPR

Can we do better?

- We do not need to sample all the edges of a triangle to count it!
- If two edges of a triangle are in the sample and the third one is observed on the stream, TRIÈST-IMPR counts the triangle independently of whether the third edge is actually sampled.
- Probability of counting a triangle increases to:

$$p_\Delta^{IMPR} = \binom{M}{2} \Big/ \binom{t}{2}$$

\+ Reduced variance of estimators

− Complexity of the analysis increases order of edges becomes relevant

TRIÈST provides unbiased estimators

ALGORITHM 2 TRIÈST-IMPR

Input: Insertion-only edge stream Σ, integer $M \geq 6$

$\mathcal{S} \leftarrow \emptyset$, $t \leftarrow 0$, $\tau \leftarrow 0$
for each element $(+,(u,v))$ from Σ **do**
  $t \leftarrow t+1$
  UpdateCounters(u,v)
  if SampleEdge$((u,v),t)$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u,v)\}$

function SampleEdge$((u,v),t)$
  if $t \leq M$ **then**
    return True
  else if FlipBiasedCoin$(\frac{M}{t})$ = heads **then**
    $(u',v') \leftarrow$ random edge from $\mathcal{S}$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(u',v')\}$
    return True
  return False

function UpdateCounters(u,v)
  $\mathcal{N}^{\mathcal{S}}_{u,v} \leftarrow \mathcal{N}^{\mathcal{S}}_u \cap \mathcal{N}^{\mathcal{S}}_v$
  $\eta = \max\{1, (t-1)(t-2)/(M(M-1))\}$
  for all $c \in \mathcal{N}^{\mathcal{S}}_{u,v}$ **do**
    $\tau \leftarrow \tau + \eta$
    $\tau_c \leftarrow \tau_c + \eta$
    $\tau_u \leftarrow \tau_u + \eta$
    $\tau_v \leftarrow \tau_v + \eta$

THEORETICAL GUARANTEES

Theorem 4.5. *Let $t \geq 0$ and assume $|\Delta^{(t)}| > 0$.*[3] *For any $\varepsilon, \delta \in (0,1)$, let*

$$\Phi = \sqrt[3]{8\varepsilon^{-2}\frac{3h^{(t)}+1}{|\Delta^{(t)}|}\ln\left(\frac{(3h^{(t)}+1)e}{\delta}\right)} .$$

If

$$M \geq \max\left\{t\Phi\left(1+\frac{1}{2}\ln^{2/3}(t\Phi)\right), 12\varepsilon^{-1}+e^2, 25\right\},$$

then $|\xi^{(t)}\tau^{(t)} - |\Delta^{(t)}|| < \varepsilon|\Delta^{(t)}|$ *with probability* $> 1-\delta$.

(ε, δ) multiplicative approximation

TRIÈST FOR FULLY DYNAMIC STREAMS

TRIÈST-FD

ALGORITHM 3 TRIÈST-FD

Input: Fully-dynamic edge stream Σ, integer $M \geq 6$

$\mathcal{S} \leftarrow \emptyset$, $d_i \leftarrow 0$, $d_o \leftarrow 0$, $t \leftarrow 0$, $s \leftarrow 0$
for each element $(\bullet,(u,v))$ from Σ **do**
  $t \leftarrow t+1$
  $s \leftarrow s \bullet 1$
  if $\bullet = +$ **then**
    if SampleEdge(u,v) **then**
      UpdateCounters$(+,(u,v))$
  else if $(u,v) \in \mathcal{S}$ **then**
    UpdateCounters$(-,(u,v))$
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{(u,v)\}$
    $d_i \leftarrow d_i + 1$
  else $d_o \leftarrow d_o + 1$

function SampleEdge(u,v)
  if $d_o + d_i = 0$ **then**
    if $|\mathcal{S}| < M$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u,v)\}$
      return True
    else if FlipBiasedCoin$(\frac{M}{t})$ = heads **then**
      Select (z,w) uniformly at random from $\mathcal{S}$
      UpdateCounters$(-,(z,w))$
      $\mathcal{S} \leftarrow (\mathcal{S} \setminus \{(z,w)\}) \cup \{(u,v)\}$
      return True
  else if FlipBiasedCoin$\left(\frac{d_i}{d_i+d_o}\right)$ = heads **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{(u,v)\}$
    $d_i \leftarrow d_i - 1$
    return True
  else
    $d_o \leftarrow d_o - 1$
    return False

- Handles edges insertions and deletions
- Single pass algorithm;
- Deletions can hit the sample subgraph;
- We build on Random Pairing sampling by *Gemulla et al. [2008]:*
  - Allows to pair deletions with future insertions;
  - We only need counters for the number of unpaired deletions.
- Variance of the estimators grows with respect to the effective size of the graph (total insertions - total deletions)

ANALYTICAL CHALLENGES

The analysis is complicated because in a fixed size sample, the inclusion of edges in the sample are not independent events:

- Proving analytical bounds for the sample variance of our algorithms is significantly harder compared to algorithms that use fixed probability sampling and have variable sample size.

Several benefits:

- full utilization of available fixed memory;
- reduced variance through the graph evolution;
- no need to fix a priori sampling probability;
- improved performance in experimental setting

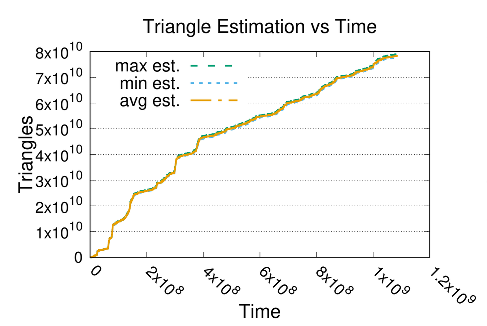
EXPERIMENTAL SETTING

Graph	$\|V\|$	$\|E\|$	$\|\Delta\|$
Patent (Co-Aut.)	1,162,227	3,660,945	3.53×10^6
Patent (Cit.)	2,745,762	13,965,410	6.91×10^6
LastFm	681,387	43,518,693	1.13×10^9
Yahoo! Answers	2,432,573	1.21×10^9	7.86×10^{10}
Twitter	41,652,230	1.47×10^9	3.46×10^{10}

Work	Single pass	Fixed space	Local counts	Global counts	Fully-dynamic streams
[Becchetti et al. 2010]	✗	✓/✗†	✓	✗	✗
[Kolountzakis et al. 2012]	✗	✗	✗	✓	✗
[Pavan et al. 2013]	✓	✓	✗	✓	✗
[Jha et al. 2015]	✓	✓	✗	✓	✗
[Ahmed et al. 2014]	✓	✗	✗	✓	✗
[Lim and Kang 2015]	✓	✗	✓	✗	✗
This work	✓	✓	✓	✓	✓

None of the previous works has all the following properties: Single pass, fully-dynamic, fixed memory space, small query time, unbiased estimate of global and local triangles.

EXPERIMENTS: TRIÈST-IMPR



Yahoo! graph with billions edges;

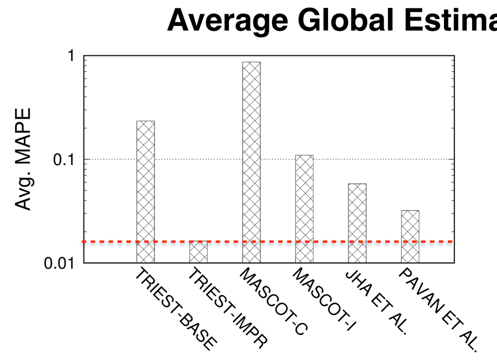
Max/Min estimate are almost coincident;

Very low variance:

$M = 10^6$ (<0.1% graph).

Highly concentrated estimates around the unbiased mean.

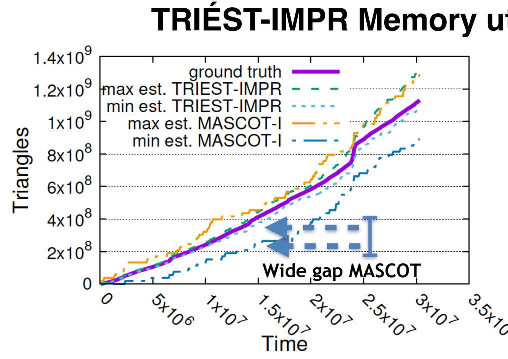
Average Global Estimation Error



Allowing the same *expected memory* use to all algorithms.

TRIÈST-IMPR has smaller error than all s.o.a methods.

TRIÈST-IMPR Memory utilization



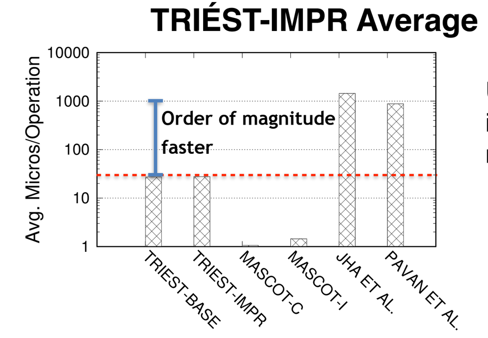
LastFM graph

- 4×10^7 + edges
- 10^9 + triangles
- $M = 10^5$

($<0.3\%$ size of the graph)

Better use of available memory space than fixed sampling probability approaches, through graph evolution

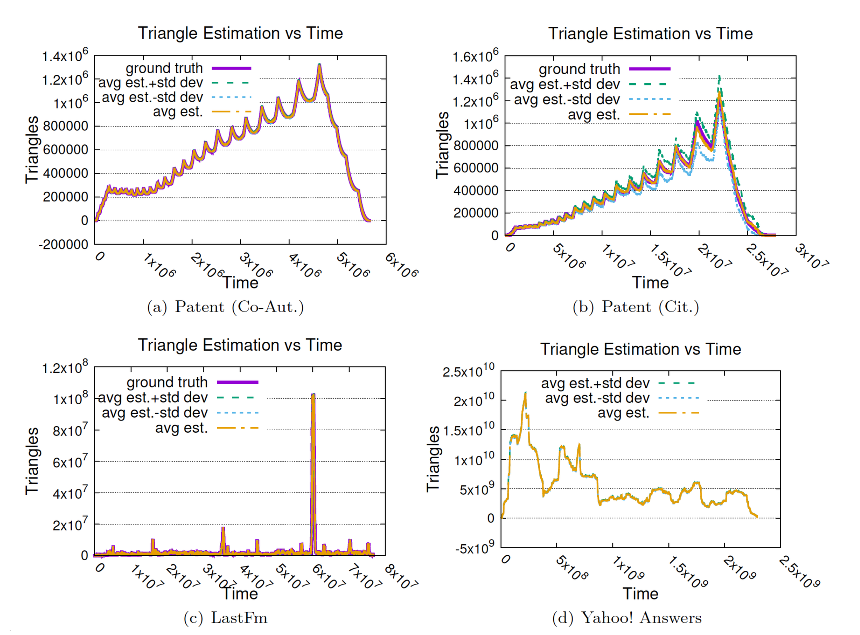
TRIÈST-IMPR Average Update Time



Update time larger only w.r.t independent sampling methods

Our algorithm is very fast ≈100 μs per update

EXPERIMENTS: TRIÈST-FD



Sliding window deletions, M = 2×10^5 for (a,b,c) and M = 10^6 for (d)

Estimation errors for sliding window deletions

Graph	M	Avg. Global MAPE	Avg. Local Pearson	Avg. Local ε Err.
LastFM	200000	0.040	0.620	0.53
	1000000	0.006	0.950	0.33
Pat. (Co-Aut.)	200000	0.060	0.278	0.50
	1000000	0.006	0.790	0.21
Pat. (Cit.)	200000	0.280	0.068	0.06
	1000000	0.026	0.510	0.04

Estimation errors for massive deletions

Graph	M	Avg. Global MAPE	Avg. Local Pearson	Avg. Local ε Err.
LastFM	200000	0.005	0.980	0.020
	1000000	0.002	0.999	0.001
Pat. (Co-Aut.)	200000	0.010	0.660	0.300
	1000000	0.001	0.990	0.006
Pat. (Cit.)	200000	0.170	0.090	0.160
	1000000	0.040	0.600	0.130

TRIÈST-FD archives high accuracy and scalability even with limited sample space and high number of deletions

REFERENCES

1. L. De Stefani, A. Epasto, M. Riondato, and E. Upfal. TRIÈST: Counting local and global triangles. n fully-dynamic streams with fixed memory size, KDD '16. ACM, 2016.
2. L. Becchetti, P. Boldi, C. Castillo, and A. Gionis. Efficient algorithms for large-scale local triangle counting. ACM TKDD , 4(3):13:1–13:28, 2010.
3. R. Gemulla, W. Lehner, and P. J. Haas. Maintaining bounded-size sample synopses of evolving datasets. The VLDB Journal , 17(2):173–201, 2008.
4. M. Jha, C. Seshadhri, and A. Pinar. A space-efficient streaming algorithm for estimating transitivity and triangle counts using the birthday paradox. ACM TKDD , 9(3):15:1–15:21, 2015.
5. M. N. Kolountzakis, G. L. Miller, R. Peng, and C. E. Tsourakakis. Efficient triangle counting in large graphs via degree-based vertex partitioning. Internet Mathematics , 8(1–2):161–185, 2012.
6. Y. Lim and U. Kang. MASCOT: Memory-efficient and accurate sampling for counting local triangles in graph streams., KDD '15, pages 685–694. ACM, 2015.
7. A. Pavan, K. Tangwongsan, S. Tirthapura, and K.-L. Wu. Counting and sampling triangles from a graph stream. Proceedings of the VLDB Endowment , 6(14):1870–1881, 2013.
8. J. S. Vitter. Random sampling with a reservoir. ACM Transactions on Mathematical Software , 11(1):37–57, 1985

Full paper and software available at: http://bigdata.cs.brown.edu/triangles.html