COMS E6998-15: Algorithms for Massive Data (Fall'23) Sep 13, 2023

# Lecture 3: Streaming Number of Triangles

Instructor: *Alex Andoni* Scribes: *Jan Olivetti*

## 1 Triangle counting

$G$ graph with $n$ nodes and $m$ edges, $T$ = number of triangles

**Algorithm 1** Algorithm for counting number of triangles in a streamed graph

Pick 3-size sets $S_1, \ldots, S_k$
Compute $X_{S_i} = \#\text{edges in } S_i, i \in [k]$
Estimate $\hat{T} = \frac{M}{k}\#\{i : X_{S_i} = 3\}$ where $M = \#\text{sets of size 3 in } [n] = \binom{n}{3}$

## 2 Probability tools

**Theorem 1** (Markov's inequality)**.** *$X$ random variable, $X \geq 0, \lambda > 0 \Rightarrow P[X \geq \lambda] \leq \frac{E[X]}{\lambda}$,*

**Corollary 2.** *$P[X < 10E[X]] \geq .9$*

**Theorem 3** (Chebyshev's inequality)**.** *$X$ random variable, $\lambda > 0 \Rightarrow P[|X - E[X] \geq \lambda] \leq \frac{Var[X]}{\lambda^2}$,*

**Corollary 4.** *$\lambda^2 = 10Var[X] \Rightarrow P[X \in [E[X] - \lambda, E[X] + \lambda]] \geq .9$*

**Remarks:**

1. $X_1, X_2, ..., X_k$ i.i.d. r.v $\Rightarrow Var[X_1 + \ldots + X_k] = kVar[X_1]$
2. $X = \begin{cases} 1, & \text{with probability } p \in (0,1) \\ 0, & \text{otherwise} \end{cases}$

   $E[X] = p$

   $Var[X] = p(1-p)$

**Theorem 5** (Chebyshev/Hoeffding inequality)**.** *$X = X_1 + \ldots + X_k$, $X_1, \ldots X_k \in [0,1]$ independent random variables, $\mu = E[X] \Rightarrow P[|X - \mu| \geq \epsilon\mu] \leq 2e^{\frac{-\epsilon^2\mu}{3}}$,*

## 3 Analysis of triangle counting

**Claim 6.** *$E[\hat{T}] = T$*

*Proof.* $E[\hat{T}] = \frac{M}{k}\sum_{i=1}^{k} E[\mathbb{1}[X_{S_i} = 3]] = \frac{M}{k}\sum_{i=1}^{k} P_{S_i}[X_{S_i} = 3] = \frac{M}{k}\sum_{i=1}^{k}\frac{T}{M} = T$ □

**Claim 7.** *$Var[\hat{T}] \leq \frac{MT}{k}$*

*Proof.* $Var[\hat{T}] = \frac{M^2}{k^2}\sum_{i=1}^k Var[\mathbb{1}[X_{S_i} = 3]] \leq \frac{M^2}{k^2}P[X_{S_i} = 3] = \frac{M^2}{k^2}k\frac{T}{M} = \frac{MT}{k}$
where the inequality comes from remark 2 and the fact that $(1-p) \leq 1$. □

We want $\hat{T} \in T \pm \epsilon T = (1 \pm \epsilon)T$, so we set $\lambda = \epsilon T$.
To use the corollary of Chebyshev/Hoeffding, $\lambda^2 = \frac{10MT}{k} \Leftrightarrow \epsilon^2 T^2 = \frac{10MT}{k} \Leftrightarrow k = \frac{10M}{\epsilon^2 T}$
We need a lower bound on $T$. Suppose we are given $t$ such that $T \geq t$. Then, the space bound is $O(\frac{M}{\epsilon^2 t}) \leq O(\frac{n^3}{\epsilon^2 t})$.
Motivation for Algorithm 2: if $m = O(n)$, since $T \leq nm = O(n^2)$ and there are $O(n^3)$ possible sets, picking tuples at random is unlikely to give any triangles.

**Algorithm 2** Algorithm for counting number of triangles in a streamed graph with $m << n^2$

Pick $k$ random sets $S_1, \ldots, S_k$ from the family $\mathcal{F}$ of sets $S$ such that $X_S \geq 1$
(Problem: we need the edges to pick sets from $\mathcal{F}$ but the edges are in the stream. This problem will be dealt with in another lecture)
Estimate $\hat{T}' = \frac{M'}{k}\#\{i : X_{S_i} = 3\}$ where $M' = \#$sets such that $X_S \geq 1$

This algorithm has space $k' = O(\frac{M'}{\epsilon^2 t}) = O(\frac{nm}{\epsilon^2 t})$.

# 4 Dynamic sampling

**Setting:** in streaming, keep a vector $\mathbf{x} \in \mathbb{Z}^n$.
**General Turnstile Model:** stream is composed of updates to $\mathbf{x}$ at position $i$: $(i_t, \delta_t), \delta_t \in \mathbb{Z}$ sets $x_{i_t} := x_{i_{t-1}} + \delta_t$.
$\ell_0$ **dynamic sampling:** at the end of the stream, output a random $(i, x_i)$ with $P[i] = \frac{(1\pm\epsilon)\mathbb{1}[x_i \neq 0]}{||\mathbf{x}||_0} \pm \frac{1}{n^3}$, where $||\mathbf{x}||_0 = \text{supp}(x) = \#$nonzeros in $\mathbf{x}$.
Essentially, we want to pick a uniformly random $i$ where $x_i \neq 0$, with the probability having some small multiplicative and additive error.
**Goal:** solve using $\log n^{O(1)}$ space.
For triangle counting, $\mathbf{x} \in \mathbb{Z}^M, x_S = \#$edges in $S, \text{supp}(x) = M'$.

- **Case 1:** $\text{supp}(x) = 1$

  $\alpha = \sum_{i=1}^n x_i i, \beta = \sum_{i=1}^n x_i$

  $\alpha_t = \alpha_{t-1} + \delta_t x_{i_t}, \beta_t = \beta_{t-1} + \delta_t$

  This is possible because this sketch is linear with respect to $\mathbf{x}$.

$$A = \begin{bmatrix} 1 & 2 & \ldots & n \\ 1 & 1 & \ldots & 1 \end{bmatrix}, \begin{bmatrix} \alpha_t \\ \beta_t \end{bmatrix} = A\mathbf{x}^t = \mathbf{A}(\mathbf{x}^{t-1} + \mathbf{e}_{i_t} + \delta_t) = \mathbf{A}\mathbf{x}^{t-1} + \mathbf{A}(\mathbf{e}_{i_t} + \delta_t)$$

  Output $(\frac{\alpha}{\beta}, \beta)$.

- **Case 2:** either output when $\text{supp}(x) \leq 1$ or output when $\text{supp}(x) > 1$

  - **Case 2.1:** if $\mathbf{x} \geq 0, \gamma = \sum_{i=1}^n x_i i^2$

    Test: check that $\alpha^2 = \beta\gamma$, if not, say "supp > 1"

- **Case 2.2:** general $\mathbf{x} \in \mathbb{Z}^n$

  $p = \text{prime} > n^3, \gamma = (\sum_i x_i z^i) \bmod p, z \in \mathbb{Z}_p$

  Test: $\gamma = \beta z^{\frac{\alpha}{\beta}} \bmod p$.

  **Claim 8.** *if* $supp(x) > 1, P_z[\text{test passes}] \leq \frac{n}{p}$