Lecture 6: Counting triangles Dynamic graphs & sampling

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

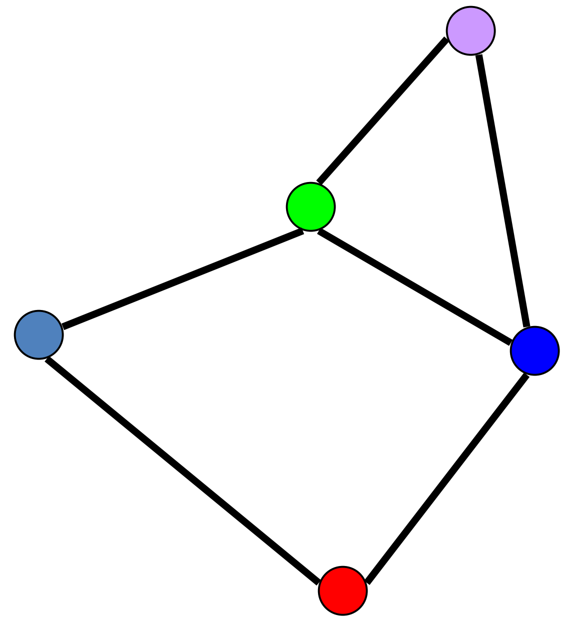
Plan

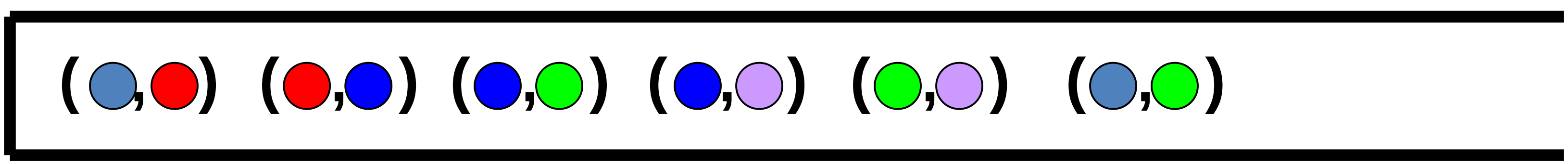
- Problem 3: Counting triangles
- Streaming for dynamic graphs

- Scriber?

Streaming for Graphs

- Graph G
  - n vertices
  - m edges
- Stream:
  list of edges
  (e.g., on a hard drive)

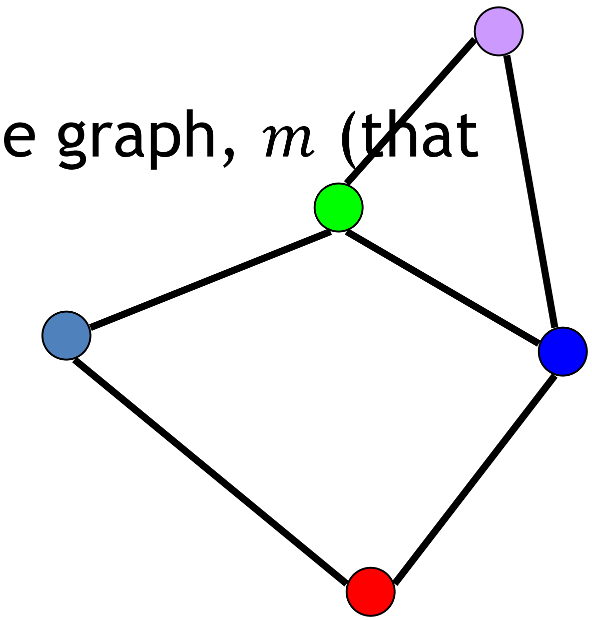


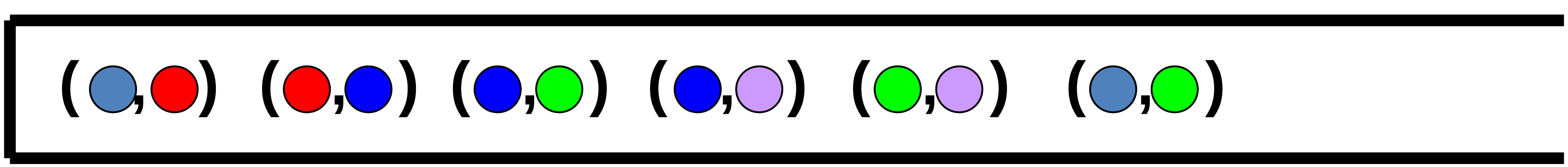


Streaming for Graphs

- Small (work)space:
  - Aim: to use $\sim n$ space
    - or $O(n \cdot \log n)$
    - Still much less than the size of the graph, m (that could be up to n^2)
  - $\ll n$ is usually *not* achievable

E.g., for web can have
$n = 1 \cdot 10^9$ nodes
$m = 100 \cdot 10^9$ edges

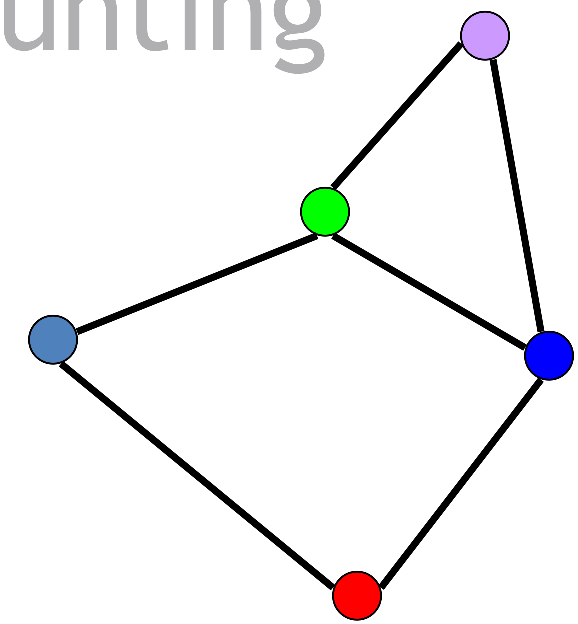




Problems

- 1. connectivity
  - Exact in $O(n)$ space
- 2. distances
  - α (odd) approximation in $O\left(n^{1+\frac{2}{\alpha+1}}\right)$
- 3. Count # triangles…

Problem 3: triangle counting



$$F = \frac{1}{1+1+3+1+3}$$

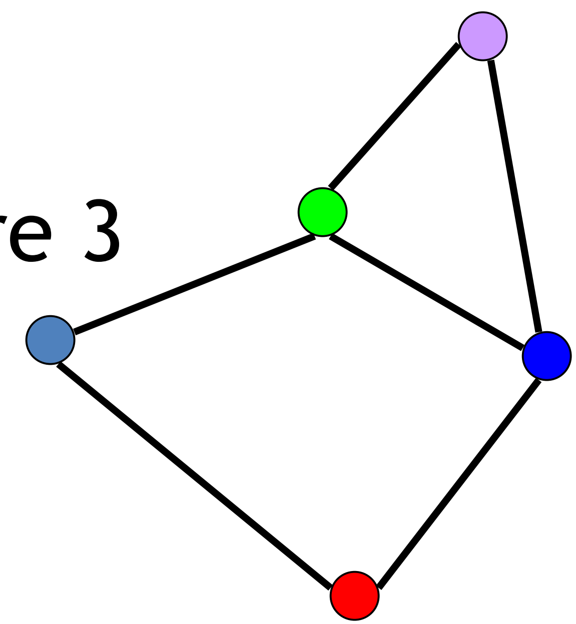
- T = number of triangles
- Motivation:
  - How often 2 friends of a person know each other?

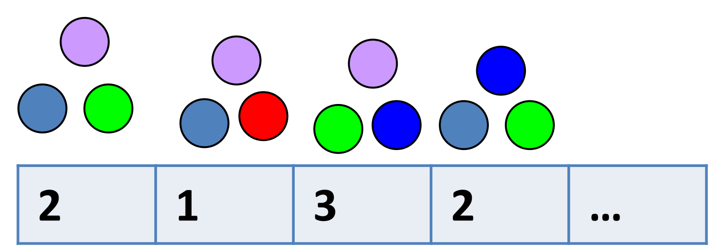
$$F = \frac{T}{3\sum_v \binom{\deg(v)}{2}}$$

  - $F \in [0,1]$
- Can we compute $\sum_v \binom{\deg(v)}{2}$ in $O(n)$ space?
  - Yes: just count the degrees, and compute
- T?
  - Hard to distinguish $T = 0$ vs $T = 1$ in $\ll m$ space
  - Suppose we have a lower bound $t \leq T$

Triangle counting: Approach

- Define a vector x with coordinate for each subset S of 3 nodes
  - x_S = how many edges among S
- $T = \#$ of coordinates which are 3
- Remember frequencies:
  - $F_p = \sum_S x_S^p$
- Claim: $T = F_0 - 1.5F_1 + 0.5F_2$





Triangle counting: Approach

- Claim: $T = F_0 - 1.5F_1 + 0.5F_2$
  - if $x_S = 0$, contributes 0 on the right
  - if $x_S = 1$, contributes 0
  - If $x_S = 2$, contributes 0
  - if $x_S = 3$, contributes 1 !
- Why such a formula exist?
  - Want a polynomial $f(x_S)$ which is =0 on {0,1,2} and =1 on {3}
  - Polynomial interpolation!
  - Need a polynomial of degree 3, but F_0 provides a degree of freedom, hence degree=2 is sufficient

Triangle Counting: Algorithm

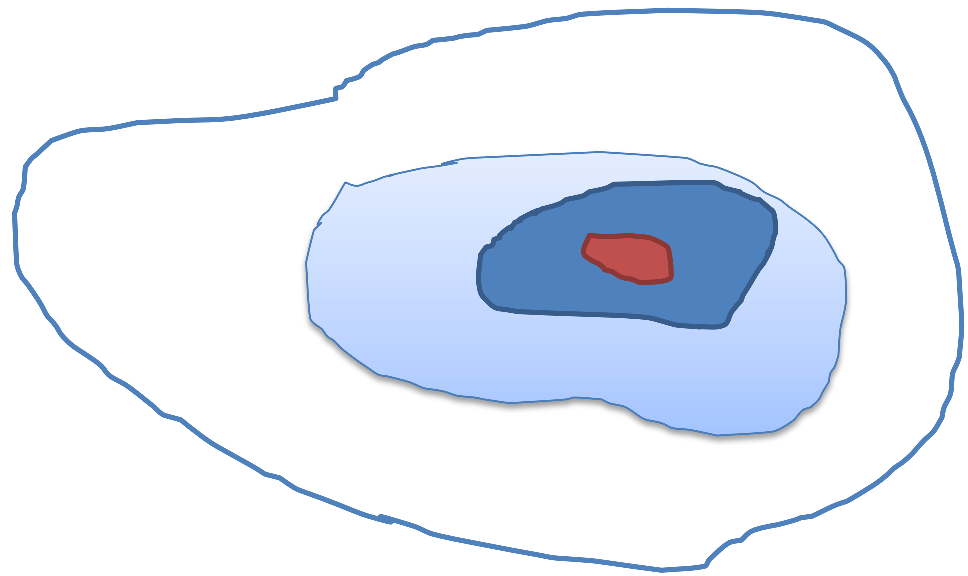
- $T = F_0 - 1.5F_1 + 0.5F_2$
- Algorithm:
  - Let $\widehat{F_0}, \widehat{F_1}, \widehat{F_2}$ be $1+\gamma$ estimates
  - General streaming!
    - Edge (i,j) increases count for all x_S s.t. $\{i,j\} \subset S$
  - $\hat{T} = \widehat{F_0} - 1.5\widehat{F_1} + 0.5\widehat{F_2}$
- How do we set γ?
  - Not a $1+\gamma$ multiplicative!
  - Additive error: $O(\gamma F_0) = O(\gamma mn)$
  - Set $\gamma = \frac{O(t)}{\epsilon mn}$ for $\pm\epsilon t$ additive error
- Total space: $O(\gamma^{-2} \log n) = O\left(\left(\frac{mn}{\epsilon t}\right)^2 \log n\right)$

Triangle Counting: Algorithm 2

- Let's consider an even simpler algorithm:
  - Pick a few random $S_1, \dots S_k$ of 3 nodes (at start)
  - Compute x_{S_i}, for $i \in [k]$ while stream goes by
  - Let c be the number of i s.t. $x_{S_i} = 3$
  - Estimate: $R = \frac{M}{k} \cdot c$, where $M = \binom{n}{3}$
- We can see:
  - $E[R] = T$
  - $Var[R] \le \frac{M}{k} T$
  - Chebyshev: $|R - T| \le O\left(\sqrt{\frac{MT}{k}}\right)$

Triangle Counting: Algorithm 2+

- $|R - T| \leq O\left(\sqrt{\frac{MT}{k}}\right)$
- Need $k = \frac{O(1)}{\epsilon^2} \cdot \frac{M}{t}$
  where $M = O(n^3)$
- Better?
  - Sample S
    from set of smaller size
    $M' \ll M$!
  - for which $x_S \geq 1$
- Then obtain:
  - Chebyshev: $|R - T| \leq O\left(\sqrt{\frac{M'T}{k}}\right)$
- Need $k = \frac{O(1)}{\epsilon^2} \cdot \frac{M'}{t} = O\left(\frac{1}{\epsilon^2} \cdot \frac{mn}{t}\right)$
  since $M' = O(mn)$



Sampling in Graphs

- Setting 1:
  - Suppose we just have positive updates
  - Not linear

- Setting 2:
  - General steaming: also negative updates…
  - Why? Dynamic graphs

Dynamic Graphs

- Stream contains both insertions and deletions of edges
  - Use 1: a log of updates to the graph
    - [Ahn-Guha-McGregor'12]: "hyperlinks can be removed and tiresome friends can be de-friended"
  - Use 2: graph is distributed over a number of computers
    - Then want *linear* sketches
    - Generally: dynamic streams ⇔ linear sketches
  - Use 3: if time-efficient, then it's a data structure!

Revisit Problem 1: Connectivity

- Can we do connectivity in dynamic graphs?
  - Algorithm from the previous lecture?
    - No…
- Theorem [AGM12]: can check s-t connectivity in dynamic graphs with $O(n \cdot \log^5 n)$ space (with 90% success probability)
- Approach: sampling in (dynamic) graphs

Sub-Problem: dynamic sampling

- Stream: general updates to a vector $x \in \{-1,0,1\}^n$
  - (will work for general x too)
- Goal:
  - Output i with probability $\frac{|x_i|}{\sum_j |x_j|}$
- Does "standard" sampling work?
  - No:
    - After putting $x_i = 1$ for $n/2$ coordinates, add 1 more and delete the first $n/2$…

Dynamic Sampling

- Goal: output i with probability $\frac{|x_i|}{\sum_j |x_j|}$
- Let $A = \{i \text{ s.t.} x_i \neq 0\}$
- Intuition:
  - Suppose $|A| = 10$
    - How can we sample i with non-zero x_i?
    - Use CountSketch!
    - Each $x_i \neq 0$ ($i \in A$) is a $1/10$-heavy hitter
    - Can recover all of them
    - $O(\log n)$ space total
  - Suppose $A = n/10$
    - Downsample first: pick a random set $D \subset [n]$, of size $|D| \approx 100$
    - Focus on substream on $i \in D$ only (ignore the rest)
    - What's $|A \cap D|$?
      - In expectation =10
    - Use CountSketch on the downsampled stream…
  - In general: prepare for all levels