Lecture 5:
Precision Sampling (cont), Streaming for Graphs

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

Plan

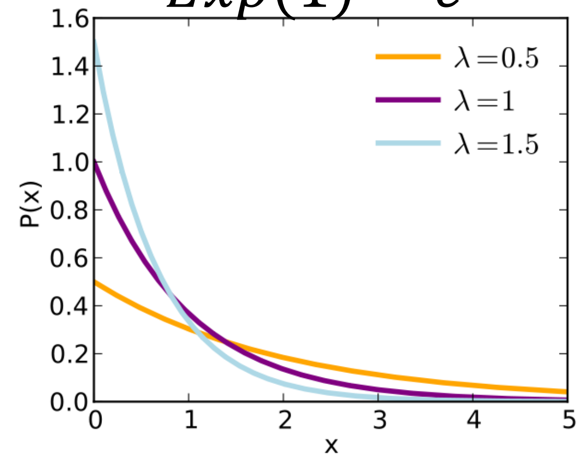
- Precision Sampling (continuation)
- Streaming for graphs

- Scriber?

Precision Sampling: Algorithm

- **Precision Sampling Lemma:** can get with 90% success:
  - O(1) additive error and 1.5 multiplicative error: $S/1.5 - O(1) < \tilde{S} < 1.5 \cdot S + O(1)$
  - with average cost equal to $O(\log n)$
- Algorithm:
  - Choose each $u_i \in Exp(1)$ i.i.d.
  - Estimator: $\tilde{S} = \max_i \tilde{a}_i / u_i$.
- Proof of correctness:
  - Claim 1: $\max a_i/u_i \sim \sum a_i / Exp(1)$
    - Hence, $\max \tilde{a}_i/u_i = \frac{\sum a_i}{Exp(1)} \pm 1$
  - Claim 2: Avg cost $= O(\log n)$

$Exp(1) \sim e^{-x}$

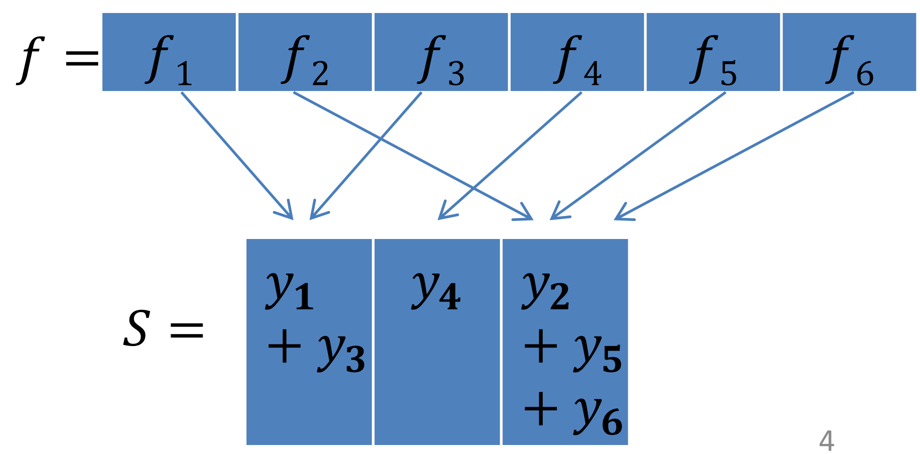


p-moments via Prec. Sampling

- **Theorem:** linear sketch for p-moment with $O(1)$ approximation, and $O(n^{1-2/p} \log^{O(1)} n)$ space (with 90% success probability).
- **Sketch:**
  - Pick random $r_i \in \{\pm 1\}$, and $u_i \sim Exp(1)$
  - let $y_i = f_i \cdot r_i / u_i^{1/p}$
  - Hash into a hash table S,

    $w = O(n^{1-\frac{2}{p}} \log^{O(1)} n)$ cells
- **Estimator:**
  - $\max_j |S[j]|^p$
- Sketch S is linear

$\boxed{u \sim e^{-u}}$



Correctness of estimation

- **Theorem:** $\max_j |S[j]|^p$ is $O(1)$ approximation with 90% probability, with
  $w = O(n^{1-2/p} \log^{O(1)} n)$ cells
- **Proof:**
  - Use Precision Sampling Lem.
  - $a_i = |f_i|^p$
    - $\sum a_i = \sum |f_i|^p = F_p$
  - $\tilde{a}_i / u_i = |S[h(i)]|^p$
  - Need to show $|a_i - \tilde{a}_i|$ small
    - more precisely: $\left|\frac{\tilde{a}_i}{u_i} - \frac{a_i}{u_i}\right| \le \epsilon F_p$

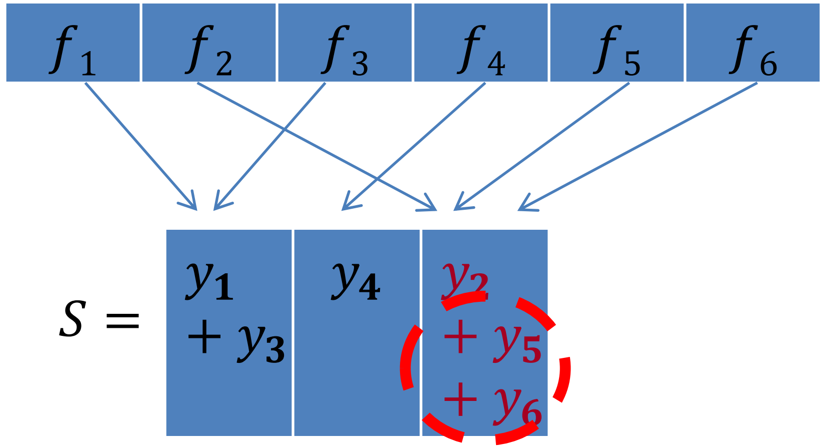
```
Algorithm PrecisionSamplingFp:

Initialize(w):
  array S[w]
  hash func h, into [w]
  hash func r, into {±1}
  reals u_i, from Exp distribution

Process(vector f ∈ ℜ^n):
  for(i=0; i<n; i++)
    S[h(i)] += f_i · r_i / u_i^{1/p};

Estimator:
  max_j |S[j]|^p
```

Correctness 2



- Claim: $\left|S[h(i)]^p - f_i^p/u_i\right| \le O(\epsilon F_p)$
- Consider cell $z = h(i)$
  - $S[z] = \frac{f_i r_i}{u_i^{1/p}} + C$
- How much chaff C is there?
  - $C = \sum_{j \ne i^*} y_j \cdot \chi[h(j) = z]$
  - $E[C^2] = \cdots \le ||y||^2/w$
  - What is $||y||^2$?
    - $E_u||y||^2 \le ||f||^2 \cdot E\left[\frac{1}{u^{2/p}}\right] = ||f||^2 \cdot O(\log n)$
  - $||f||^2 \le n^{1-2/p}||f||_p^2$
  - By Markov's: $C^2 \le ||f||_p^2 \boxed{\cdot n^{1-2/p} \cdot O(\log n)/w}$ with probability >90%
- Set $w = \frac{1}{\epsilon^{2/p}} n^{1-2/p} \cdot O(\log n)$, then
  - $|C|^p \le ||f||_p^p \cdot \epsilon = \epsilon F_p$

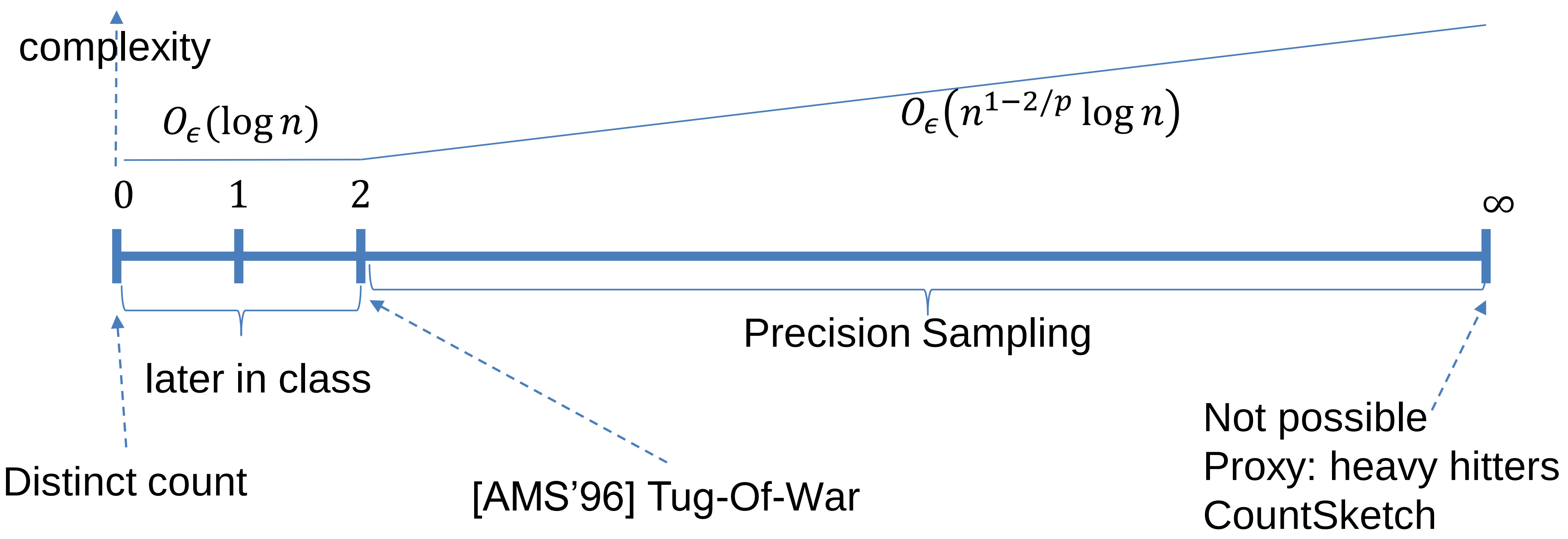
$y_i = f_i \cdot r_i / u_i^{1/p}$
where $r_i \in \{\pm 1\}$
u_i exponential r.v.

Correctness (final)

- **Claim:** $\left|S[h(i)]^p - f_i^p/u_i\right| \leq O(\epsilon F_p)$
- $S[h(i)]^p = \left(\frac{f_i}{u_i^{1/p}} + C\right)^p$
  - where $C = \sum_{j \neq i^*} y_j \cdot \chi[h(j) = h(i)]$
- Proved:
  - $E[C^2] \leq ||y||^2/w$
  - this implies $C^p \leq \epsilon F_p$ with 90% for fixed i
  - But need for all i !
- Want: $C^2 \leq \beta ||y||^2/w$ with high probability for some smallish β
  - Can indeed prove for $\beta = O(\log^2 n)$ with strong concentration inequality (Bernstein).

Recap: frequency moments

- p-moment for $p > 2$
  - Via Precision Sampling
    - Estimate of sum from poor estimates

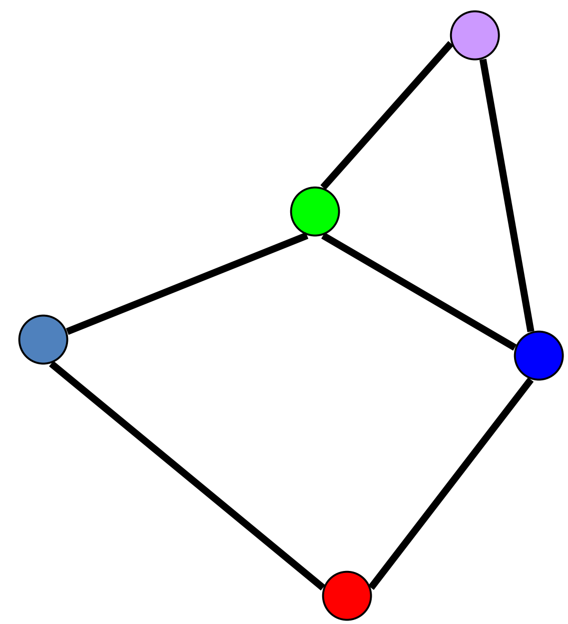


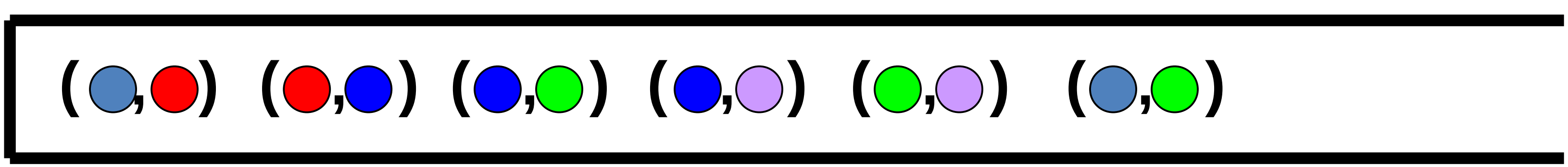
COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Streaming for Graphs

Streaming for Graphs

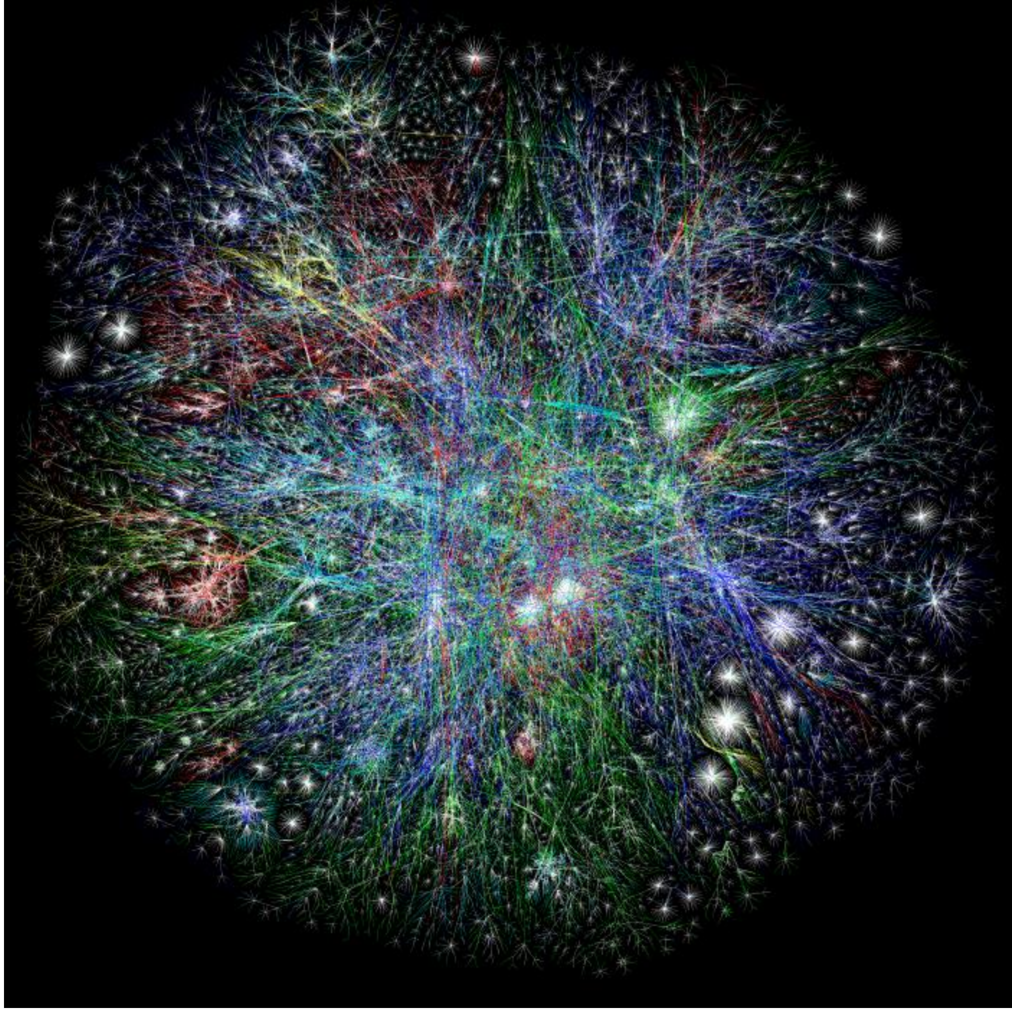
- Graph G
  - n vertices
  - m edges
- Stream:
  list of edges
  (e.g., on a hard drive)





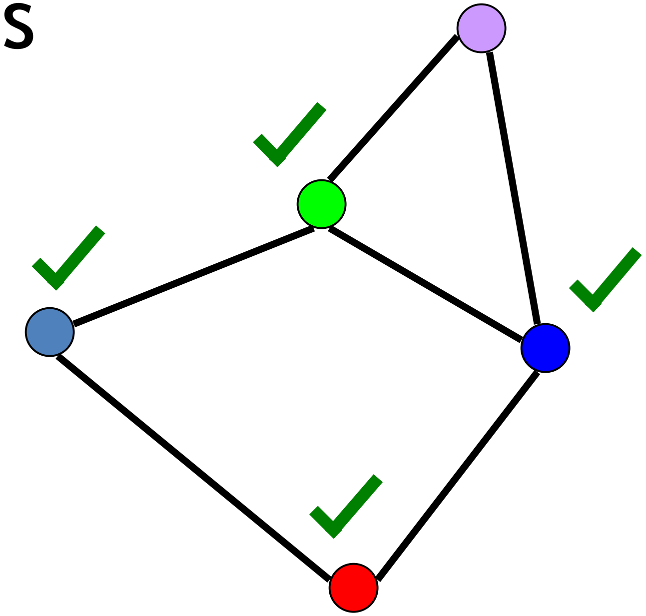
Graphs

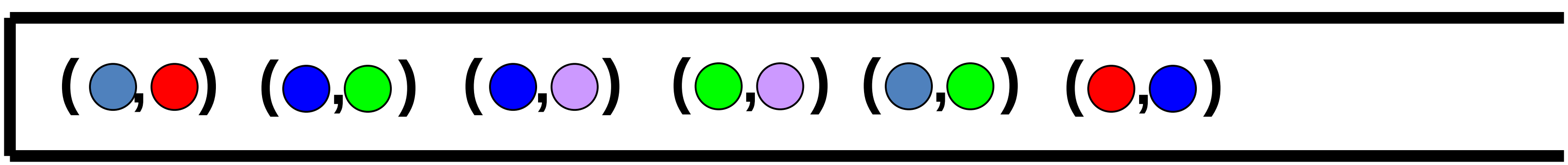
- Web
- Social graphs
- Phone calls
- Maps
- Geographical data
- …



Why streaming for graphs?

- Want to run graph algorithms
  - graph stored on hard drive
  - A linear scan on hard MUCH more efficient than random access
  - Usual algorithms are usually random-access
    - think Breadth-First-Search





For which problems?

- Most of usual-suspect algorithms use random-access

- Questions:
  - Connectivity
  - Distances (similarities) between nodes
  - PageRank (stationary distribution of random walk)
  - Counting # of triangles (measure of clusterability)
  - Various other statistics
  - Matchings
  - Graph partitioning
  - …

Parameters for graph algorithms

- Size of the workspace:
  - Aim: to use $\sim n$ space
    - or $O(n \cdot \log n)$
    - Still much less than 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

Problem 1: connectivity

- Check whether the graph is connected?
  - in $O(n)$ space
- Idea:
  - Store minimum spanning tree
- Algorithm:
  - Keep a subgraph H (starts empty)
  - when see an edge (i, j):
    - If (i, j) does not create a cycle in H, add it to H
  - Space: $\leq n - 1$ edges only
- Can use H for:
  - Connectivity between 2 nodes
  - # connected components

Problem 2: distance

- Given s, t, compute the distance between them
  - Up to approximation α, odd integer
- Modification of the previous algorithm:
  - Keep a subgraph H
  - On edge (i, j): if $d_H(i, j) > \alpha$, add (i, j) to H
- Space?
  - All cycles in H have length $\geq \alpha + 2$
  - Thm [Bollobas]: then $|H| \leq O\left(n^{1+\frac{2}{\alpha+1}}\right)$
- Few other results known!

Detour: Bollobas Theorem

- Thm [Bollobas]: If all cycles of length $\geq \alpha + 2$ then $|H| \leq O\left(n^{1+\frac{2}{\alpha+1}}\right)$
- Simplified case: all nodes of degree d
- Proof:
  - Suppose: $\alpha = 2k - 1$
  - Explore a vertex v
  - At depth k: all nodes differ!
  - Hence $d^k \leq n$
  - Or $d \leq n^{1/k}$
  - $m \leq n^{1+1/k} = n^{1+\frac{2}{\alpha+1}}$