Lp sampling

$p>0$, $\varepsilon>0$, constant c, return $l \in [n]$

$$\Pr[l=i] = (1\pm\varepsilon)\frac{|x_i|^p}{\|x\|_p^p} \pm n^{-c}$$

L0 sampling

$$\Pr[l=i] = (1\pm\varepsilon)\frac{|x_i|^0}{\|x\|_0} \pm n^{-c}$$

$$0^0 = 0$$

$\varepsilon = 0$, $O(\log^4 n)$ bits

Assume x_i are integers in $\{-n^{O(1)} \ldots n^{O(1)}\}$

L0 sampling sketch

Idea: subsample elements in $[n]$ at different sampling rates

Algorithm:

For $j = [\log n]$ we will maintain hash f^n $h_j : [n] \to \{0, \ldots 2^j - 1\}$

For each j maintain

- $D_j = (1\pm 0.1)\|x_{S_j}\|_0$ $\quad S_j = \{i : h_j(i) = 0\}$ (failure prob. $\frac{1}{n^{O(1)}}$)
- $C_j = \sum_{i\in S_j} x_i$
- $T_j = \sum_{i\in S_j} i\, x_i$

Sample: Select smallest j^* such that $D_{j^*} = 1 \pm 0.1$ (if such j^* exists)

Output $\frac{T_{j^*}}{C_{j^*}} = i$

Claim: Algorithm succeeds with constant probability

Proof: $T = \mathrm{supp}(x)$, $|T| = \|x\|_0$

$$j = \lceil \log |T| \rceil$$

$$2^j = 2^{\lceil \log|T| \rceil} = [|T|, 2|T|)$$

$$Pr[\,|T\cap S_j| = 1\,] = \sum_{i\in T} Pr[\,i\in S_j \text{ and } i'\notin S_j \ \forall\, i'\in T,\ i'\neq i\,]$$

This calculation goes through (approx.) with k-wise independence $\longrightarrow$

$$= \sum_{i\in T} \frac{1}{|T|}\left(1-\frac{1}{|T|}\right)^{|T|-1} \qquad \frac{1}{2^j} \in \left(\frac{1}{2|T|}, \frac{1}{|T|}\right]$$

$$= \left(1-\frac{1}{|T|}\right)^{|T|-1} \approx \frac{1}{e}$$

Assuming no failures, algo. samples i uniformly at random from support of x. Repeat $O(\log n)$ times to get failure prob. $\frac{1}{n^{O(1)}}$

Instead of fully random hash f^n, can use k-wise independent hash f^n for $k = O(\log n)$

$T_j, C_j : O(\log n)$ $\quad D_j : O(\log^2 n)$

$\log n$ levels, repeat : $O(\log n)$ $\qquad \log\left(\frac{1}{\delta}\right)$

Storage : $O(\log^4 n)$ $\qquad O(\log^3 n \log(\frac{1}{\delta}))$

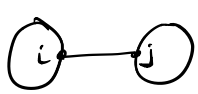
Best known : $O(\log^2 n \log(\frac{1}{\delta}))$ failure prob. δ

Graph Sketching

Connectivity :

Space : $O(n \log^{O(1)} n)$

Warmup : Only insertions



$c(i) \quad c(j)$

$\downarrow$

$\min\{c(i), c(j)\}$

Offline spanning forest algo:

Initially each node is its own component

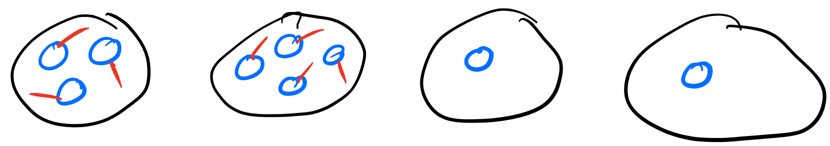
Repeat $O(\log n)$ times

- Each connected component picks an incident edge if one exist
- All connected components that are connected by newly picked edges are merged.

__Claim:__ CC : correct # connected components.

CC_i : # connected components in i^{th} iteration

$$(CC_{i+1} - CC) \leq (CC_i - CC)/2$$



Converges in $\log n$ iterations

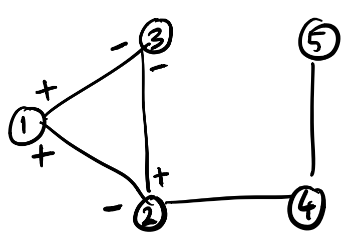
[Ahn, Guha, McGregor '12]

Node-Neighbor representation

For node i, X_i vector indexed by pairs of nodes

$$X_i \in \{-1, 0, 1\}^{\binom{n}{2}}$$

$$X_i[i,j] = \begin{cases} 1 & \text{if } (i,j) \in E \text{ and } j > i \\ -1 & \text{if } (i,j) \in E \text{ and } j < i \\ 0 & \text{if } (i,j) \notin E \end{cases}$$



$$\text{Supp}\left(\sum_{i \in S} X_i\right) = E(S, V \setminus S)$$

Maintain L_0 sampling sketches $A^s X_i$ for each X_i, $s \in \{1, \dots \log n\}$

Initially each node is its own component

For $s = 1$ to $\log n$

For each component C, compute

$$A^s\left(\sum_{i \in C} X_i\right) = \sum_{i \in C} A^s X_i$$

Use this sketch to sample an edge in $E(C, V \setminus C)$ if one exists.

Merge connected components.

Q^n: Each phase uses a distinct sketch. Why do we need this?

Note that correctness guarantees for l_0-sampling sketch break down if we use the results of previous queries to operate on the sketch and query again.

Consider this example to illustrate the issue: Suppose we construct l_0-sampling sketch for edges incident on a node. Use the sketch to sample an edge, delete the edge from the sketch and repeat the sampling process. One should be able to recover all the edges incident on the node. But this is not possible, since the sketch takes polylogarithmic space while the number of edges could be much larger!