Topics in Streaming

Lecture 18 and 19

October 27 and 29, 2020

Topics in Streaming

- F_p estimation for $p \in (0, 2]$ via p-stable distributions and pseudorandom generators
- Priority Sampling
- Precision Sampling and Applications to ℓ_2 sampling in streams
- ℓ_0 Sampling

Part I

F_p Estimation

F_2 Estimation and JL

For F_2 estimation and JL and Euclidean LSH we used important "stability" property of the Normal distribution.

Lemma

Let $Y_1, Y_2, \ldots, Y_d$ be independent random variables with distribution $\mathcal{N}(0,1)$. $Z = \sum_i x_i Y_i$ has distribution $\|x\|_2 \mathcal{N}(0,1)$

Standard Gaussian is 2-stable.

p-stable distributions

Definition

A real-valued distribution $\mathcal{D}$ is p-stable if $Z = \sum_{i=1}^{n} x_i Y_i$ has distribution $\|x\|_p \mathcal{D}$ when the Y_i are independent and each of them is distributed as $\mathcal{D}$.

$$\sum_{i=1}^{n} x_i Y_i \approx \|x\|_p Z$$

$$Z \approx \mathcal{D}$$

p-stable distributions

Definition

A real-valued distribution $\mathcal{D}$ is p-stable if $Z = \sum_{i=1}^{n} x_i Y_i$ has distribution $\|x\|_p \mathcal{D}$ when the Y_i are independent and each of them is distributed as $\mathcal{D}$.

Question: Do p-stable distributions exist for $p \neq 2$?

p-stable distributions

Fact: p-stable distributions exist for all $p \in (0, 2]$ and do not exist for $p > 2$.

$p = 1$ is the Cauchy distribution which is the distribution of the ratio of two independent Guassian random variables. Has a closed form density function $\frac{1}{\pi(1+x^2)}$. Mean and variance are *not* finite.

p-stable distributions

Fact: p-stable distributions exist for all $p \in (0, 2]$ and do not exist for $p > 2$.

$p = 1$ is the Cauchy distribution which is the distribution of the ratio of two independent Guassian random variables. Has a closed form density function $\frac{1}{\pi(1+x^2)}$. Mean and variance are *not* finite.

For general p no closed form formula for density but can sample from the distribution.

p-stable distributions

Fact: p-stable distributions exist for all $p \in (0, 2]$ and do not exist for $p > 2$.

$p = 1$ is the Cauchy distribution which is the distribution of the ratio of two independent Guassian random variables. Has a closed form density function $\frac{1}{\pi(1+x^2)}$. Mean and variance are *not* finite.

For general p no closed form formula for density but can sample from the distribution.

Streaming, sketching, LSH ideas for ℓ_2 generalize to ℓ_p for $p \in (0, 2]$ via p-stable distributions and additional technical work.

Sampling from p-stable distribution

For $p \in (0,2]$ let $\mathcal{D}_p$ denote p-stable distribution. Sampling from $\mathcal{D}_p$ via Chambers-Mallows-Stuck method

- Sample θ uniformly from $[-\pi/2, \pi/2]$.
- Sample r uniformly from $[0,1]$.
- Output

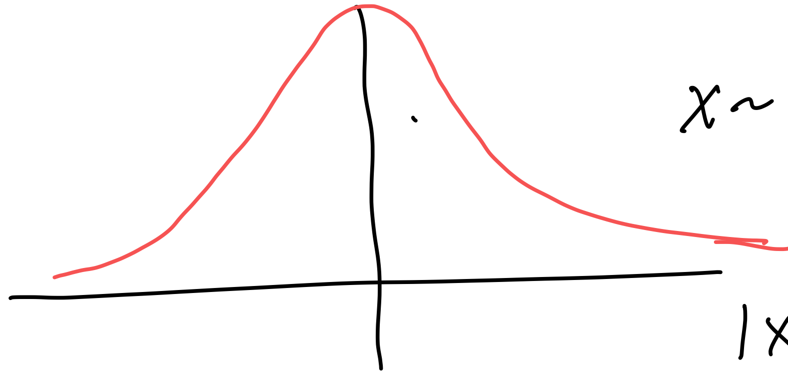
$$\frac{\sin(p\theta)}{(\cos\theta)^{1/p}}\left(\frac{\cos((1-p)\theta)}{\ln(1/r)}\right)^{(1-p)/p}.$$

p-stable distributions need not have finite mean/variance. Hence we need to work with *median* of distribution.

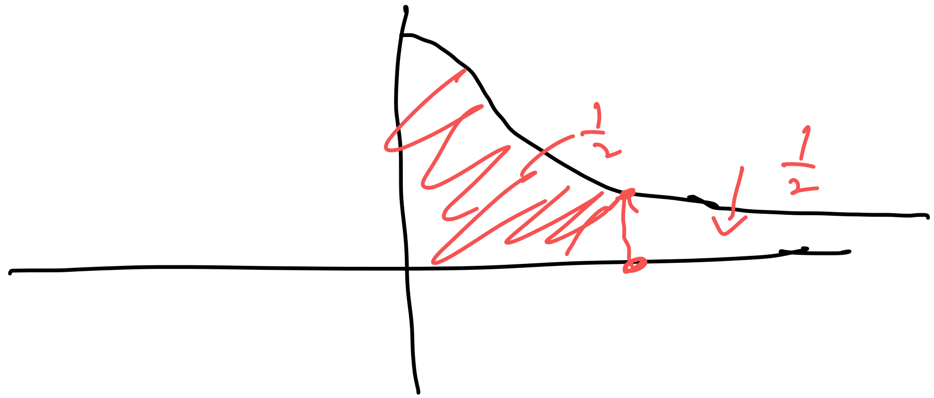
Definition

The median of a distribution $\mathcal{D}$ is θ if for $Y \sim \mathcal{D}$, $\Pr[Y \leq \mu] = 1/2$. If $\phi(x)$ is the probability density function of $\mathcal{D}$ then we have $\int_{-\infty}^{\mu} \phi(x)dx = 1/2$.

$X \sim N(0,1)$



$|X|$.



F_p estimation via p-stable distribution

For $p \in (0, 2]$ due to [Indyk]

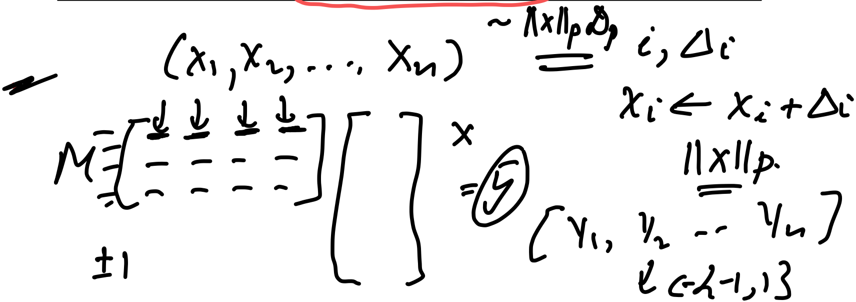
F_p-Estimate:

$k \leftarrow \Theta(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$

Let M be a $k \times n$ matrix where each $M_{ij} \sim \mathcal{D}_p$

$\mathbf{y} \leftarrow M\mathbf{x}$

Output $Y \leftarrow \frac{\text{median}(|y_1|, |y_2|, \ldots, |y_k|)}{\text{median}(|\mathcal{D}_p|)}$



$M\bar{x} = \bar{y}$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix}$$

$y_i \approx \|x\|_p \mathcal{D}_p$

$E[y_i] = 0$

$|y_i|^p$

$$\frac{\text{median}(|y_1|, |y_2|, \ldots, |y_k|)}{\text{median}(|\mathcal{D}_p|)}$$

$\approx \|x\|_p .$

F_p estimation via p-stable distribution

For $p \in (0, 2]$ due to [Indyk]

F_p-Estimate:

$k \leftarrow \Theta(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$

`Let` M `be a` $k \times n$ `matrix where each` $M_{ij} \sim \mathcal{D}_p$

$y \leftarrow Mx$

`Output` $Y \leftarrow \frac{\text{median}(|y_1|, |y_2|, \ldots, |y_k|)}{\text{median}(|\mathcal{D}_p|)}$

- Each y_j is distributed according to $\|x\|_p \mathcal{D}_p$
- Cannot take average of $|y_j|^p$ values since mean of distribution is not finite
- Take median of absolute values for k independent copies and normalize by median of distribution

Concentration Lemma

Lemma

Let $\epsilon > 0$ *and let* $\mathcal{D}$ *be a distribution with density function* ϕ *and a unique median* $\mu > 0$*. Suppose* ϕ *is absolutely continuous on* $[(1-\epsilon)\mu, (1+\epsilon)\mu]$ *and let* $\alpha = \min\{\phi(x) \mid x \in [(1-\epsilon)\mu, (1+\epsilon)\mu]$*. Let* $Y = \textit{median}(Y_1, Y_2, \ldots, Y_k)$ *where* $Y_1, \ldots, Y_k$ *are independent samples from the distribution* $\mathcal{D}$*. Then*

$$\Pr[|Y - \mu| \geq \epsilon\mu] \leq 2e^{-\frac{2}{3}\epsilon^2\mu^2\alpha^2 k}.$$

See notes for proof idea.

Pseudorandom generator for F_p Estimation

For F_p estimation we need $M_{i,j}$ to be independent randomly distributed according to $\mathcal{D}_p$. Can use sampling from distribution even though it is not explicit.

How do we store M in small space?

Recall that for F_2 estimation and sketching we used matrix M where each row of M had 4-wise independent random variables. Needed separate proof to argue correctness.

Is there an equivalent limited independence hashing based algorithm for F_p estimation?

Pseudorandom generator for F_p Estimation

For F_p estimation we need $M_{i,j}$ to be independent randomly distributed according to $\mathcal{D}_p$. Can use sampling from distribution even though it is not explicit.

How do we store M in small space?

Recall that for F_2 estimation and sketching we used matrix M where each row of M had 4-wise independent random variables. Needed separate proof to argue correctness.

Is there an equivalent limited independence hashing based algorithm for F_p estimation? No but can use a powerful pseudorandomness tool from TCS.

Pseudorandom generator

- *P* class of decision problems decided in poly time.
- *RP* class of decision problems decided in randomized poly time with one-sided error
- *BPP* class of decision problems decided in randomized poly time with two-sided error allowed

Pseudorandom generator

- P class of decision problems decided in poly time.
- RP class of decision problems decided in randomized poly time with one-sided error
- BPP class of decision problems decided in randomized poly time with two-sided error allowed

Big Open Problem: Is $BPP = P$? Equivalently can every randomized polynomial time algorithm be derandomized with only polynomial-factor slow down?

Pseudorandom generator

- *P* class of decision problems decided in poly time.
- *RP* class of decision problems decided in randomized poly time with one-sided error
- *BPP* class of decision problems decided in randomized poly time with two-sided error allowed

Big Open Problem: Is $BPP = P$? Equivalently can every randomized polynomial time algorithm be derandomized with only polynomial-factor slow down?

Equivalently: Is there a pseudo-random generator that fools every poly-sized algorithm?

Nisan's pseudorandom generator

Nisan constructed explicit pseudo-random generator that fools space-bounded algorithms.

Theorem

Let $\mathcal{A}$ be an randomized algorithm that uses space at most $S(n)$ on an input of length n. Then there is a pseudo-random generator G that fools $\mathcal{A}$ and has seed length $\ell = O(S(n) \log n)$ and which is computable in $O(\ell)$ space and $poly(\ell)$ time.

Corollary

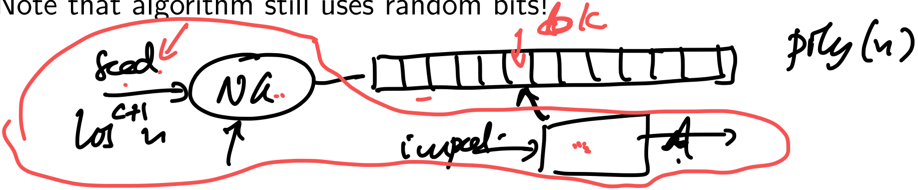
For $S(n) = O(\log^c n)$ the generator uses space $S(n) = O(\log^{c+1} n)$ and can generate any of the desired random pseudo-random bits for algorithm in $poly(\log n)$ time.

Applying Nisan's generator as a hammer

At a high-level if a streaming algorithm uses small space (polylogarithmic in input size) assuming access to *true* random bits then one can use Nisan's generator to reduce space.

- Nisan's generator requires small random seed. Store it.
- Generate required (pseudo)random bits "on the fly". Note that Nisan's generator itself runs in small space so total space is small.

Note that algorithm still uses random bits!



Applying Nisan's generator as a hammer

At a high-level if a streaming algorithm uses small space (polylogarithmic in input size) assuming access to *true* random bits then one can use Nisan's generator to reduce space.

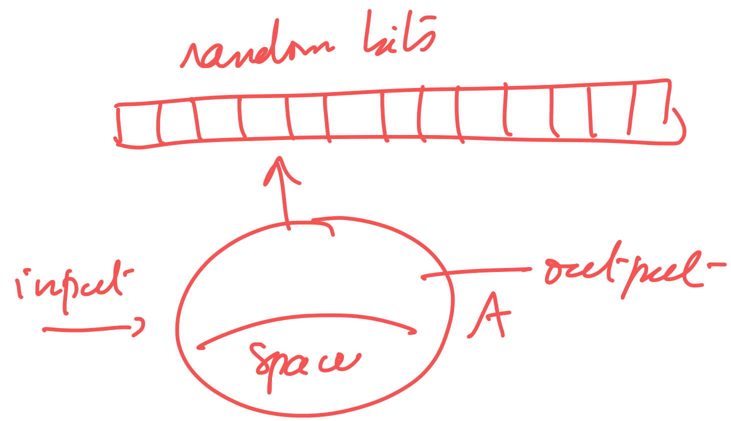
- Nisan's generator requires small random seed. Store it.
- Generate required (pseudo)random bits "on the fly". Note that Nisan's generator itself runs in small space so total space is small.

Note that algorithm still uses random bits!

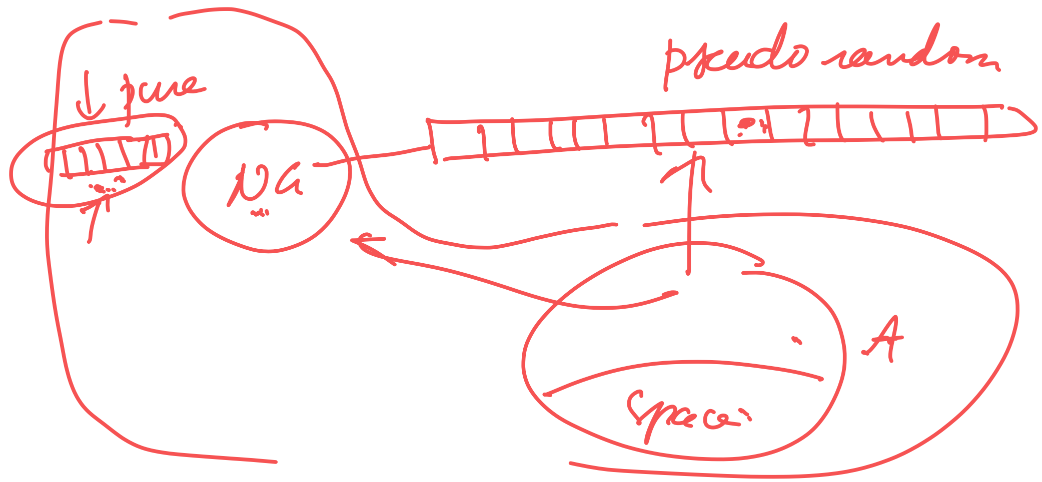
With additional discretization tricks one can convert Indyk's F_p estimation algorithm via Nisan's generator into a true small space algorithm.

[Kane-Nelson-Woodruff] show how to use limited independence hashing for F_p estimation instead of above hammer.

Original



New/Modifie



Part II

Priority Sampling

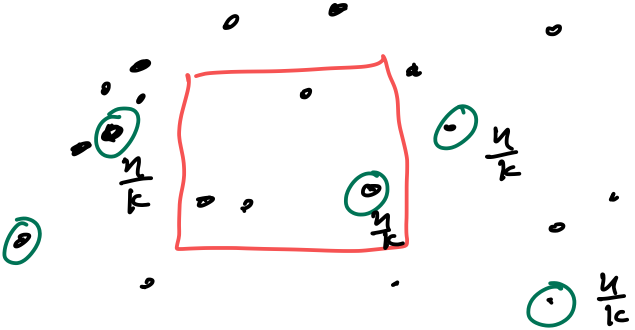
Sampling for data reduction

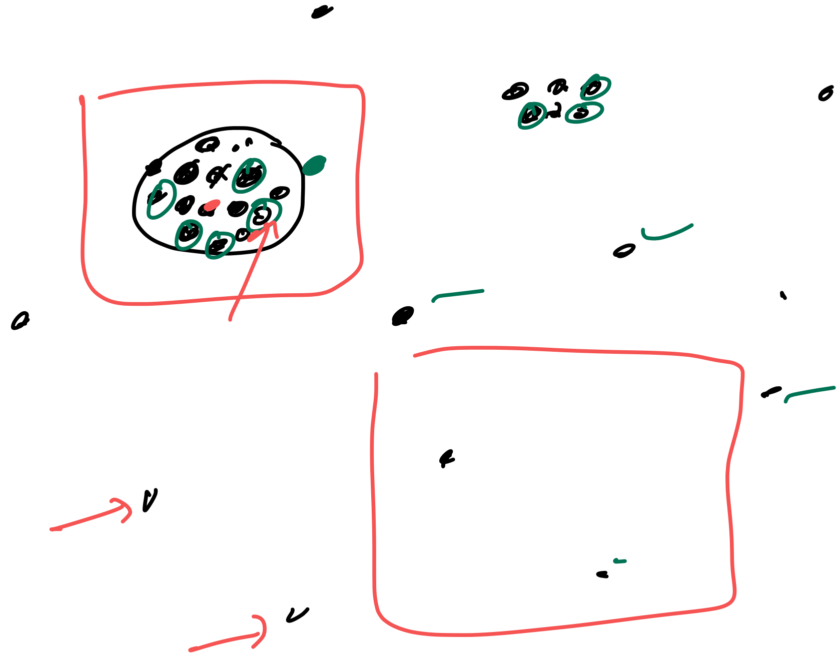
- X set of n points in the plane $a_1, a_2, \ldots, a_n$.
- Want to answer queries of the form: given some shape C (say circles), how many points inside C?
- standard data structures or brute force linear search say

n

picked k from n.

$\frac{n}{k} \times$





Sampling for data reduction

- X set of n points in the plane $a_1, a_2, \ldots, a_n$.
- Want to answer queries of the form: given some shape C (say circles), how many points inside C?
- standard data structures or brute force linear search say

Question: Suppose n is too large and we can only store k points for some $k < n$.

Sampling approach:

- S sample of size k (with replacement). Store only S
- Given query C, compute $|C \cap S|$. What should we report as an estimate for $|C \cap X|$?

Sampling for data reduction

- X set of n points in the plane $a_1, a_2, \ldots, a_n$.
- Want to answer queries of the form: given some shape C (say circles), how many points inside C?
- standard data structures or brute force linear search say

Question: Suppose n is too large and we can only store k points for some $k < n$.

Sampling approach:

- S sample of size k (with replacement). Store only S
- Given query C, compute $|C \cap S|$. What should we report as an estimate for $|C \cap X|$? $\frac{n}{k}|C \cap S|$ which is an unbiased estimator

Weighted case

- X set of n points in the plane $a_1, a_2, \ldots, a_n$. Each point a_i has a non-negative weight w_i
- Want to answer queries of the form: given some shape C (say circles), what is weight of point inside C?

Question: Suppose n is too large and we can only store k points for some $k < n$.

Sampling approach?

Weighted case

- X set of n points in the plane $a_1, a_2, \ldots, a_n$. Each point a_i has a non-negative weight w_i
- Want to answer queries of the form: given some shape C (say circles), what is weight of point inside C?

Question: Suppose n is too large and we can only store k points for some $k < n$.

Sampling approach?

- Easy to see that uniform sampling is not ideal
- Sample in proportion to weight? Say a_i sampled with $p_i = w_i/W$ where $W = \sum_i w_i$.
- What do we set the weight of the sampled points to? Can we control sample size? What is the variance?

Importance Sampling

- Decide sampling probabilities $p_1, p_2, \ldots, p_n$
- Choose a_i independently with probability p_i and if i is chosen set $\hat{w}_i = w_i/p_i$. If i is not chosen we implicitly set $\hat{w}_i = 0$.

$$E[\hat{w}_i] = p_i \cdot \frac{w_i}{p_i} = w_i$$

Importance Sampling

- Decide sampling probabilities $p_1, p_2, \ldots, p_n$
- Choose a_i independently with probability p_i and if i is chosen set $\hat{w}_i = w_i/p_i$. If i is not chosen we implicitly set $\hat{w}_i = \mathbf{0}$.
- For any i, $\mathbf{E}[\hat{w}_i] = w_i$.

$$\sum_{i=1}^{n} p_i$$

Importance Sampling

- Decide sampling probabilities $p_1, p_2, \ldots, p_n$
- Choose a_i independently with probability p_i and if i is chosen set $\hat{w}_i = w_i/p_i$. If i is not chosen we implicitly set $\hat{w}_i = 0$.
- For any i, $\mathbf{E}[\hat{w}_i] = w_i$. Hence for any C, $\mathbf{E}[\hat{w}(C \cap S)] = \mathbf{E}[w(C \cap S)]$.

Importance Sampling

- Decide sampling probabilities $p_1, p_2, \ldots, p_n$
- Choose a_i independently with probability p_i and if i is chosen set $\hat{w}_i = w_i/p_i$. If i is not chosen we implicitly set $\hat{w}_i = 0$.
- For any i, $\mathbf{E}[\hat{w}_i] = w_i$. Hence for any C, $\mathbf{E}[\hat{w}(C \cap S)] = \mathbf{E}[w(C \cap S)]$.

Question: How should we choose p_i's?

Importance Sampling

- Decide sampling probabilities $p_1, p_2, \ldots, p_n$
- Choose a_i independently with probability p_i and if i is chosen set $\hat{w}_i = w_i/p_i$. If i is not chosen we implicitly set $\hat{w}_i = 0$.
- For any i, $\mathbf{E}[\hat{w}_i] = w_i$. Hence for any C, $\mathbf{E}[\hat{w}(C \cap S)] = \mathbf{E}[w(C \cap S)]$.

Question: How should we choose p_i's?

- Choose to reduce variance for queries of interest (depends on queries)
- Expected number of chosen points is $\sum_i p_i$ and hence choose p_i's to roughly meet the memory bound. If we have memory of size k then can scale p_i values (sampling rate) to achieve this.

Importance Sampling in Streaming Setting

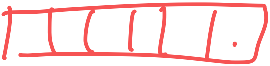
Setting:

- points $a_1, \ldots, a_n$ with weights arriving in stream
- have a memory size of k
- want to maintain a k-sample (to utilize memory as well as possible) such that we can estimate $w(C \cap X)$ accurately
- Stream length unknown! How can we adjust sampling rate?

$w_1 \quad w_2 \quad \ldots \quad w_n$

$a_1, a_2, \ldots, a_n$

Memory k.



Given query C. [illegible]

want to estimate $w(C \cap X)$

Priority Sampling

[Duffield,Lund,Thorup]

- Queries are arbitrary subset sums so no structure there to exploit
- Focus on streaming aspect and using memory

Priority Sampling

[Duffield,Lund,Thorup]

- Queries are arbitrary subset sums so no structure there to exploit
- Focus on streaming aspect and using memory

$q_i := \frac{w_i}{u_i}$ $u_i \in_R (0,1)$

Scheme:

1. For each $i \in [n]$ set priority $q_i = w_i / u_i$ where u_i is chosen uniformly (and independently from other items) at random from $[0, 1]$.
2. S is the set of items with the k highest priorities.
3. τ is the $(k+1)$'st highest priority. If $k \geq n$ we set $\tau = 0$.
4. If $i \in S$, set $\hat{w}_i = \max\{w_i, \tau\}$, else set $\hat{w}_i = 0$.

$a_1, \; a_2, \; a_{10}, \; a_n$

$u_1 \quad u_2 \quad \dots \quad u_n$ indep

$u_i \in (0,1)$

$0.1 \quad 0.5 \quad 0.4 \quad 0.6$

$q_1 = \frac{1}{0.1} \quad \frac{1}{0.5} \quad \frac{1}{0.4} \quad \frac{1}{0.6}$ $\quad k = 2$

$S = \{a_1, a_{10}\}$ $\quad \tau = \frac{1}{0.5}$

$\hat{w}_1 = \max\{w_1, \tau\}$

$\{1, \tau\}.$

τ

$\frac{1}{k+2}$

$k+2$

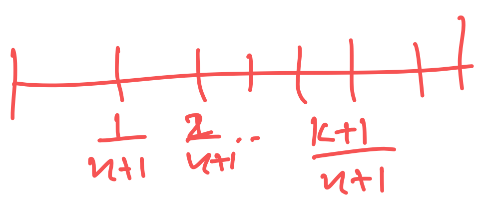
$\tau \approx ?$

throwing n random #

n (0,1) random #s.

want k+1 st smallest value?

$\frac{k+2}{n}$ $\quad \tau = \frac{n}{k+2}$



$\frac{n+1}{k+1}$

Priority Sampling

[Duffield,Lund,Thorup]

- Queries are arbitrary subset sums so no structure there to exploit
- Focus on streaming aspect and using memory

Scheme:

1. For each $i \in [n]$ set priority $q_i = w_i/u_i$ where u_i is chosen uniformly (and independently from other items) at random from $[0, 1]$.
2. S is the set of items with the k highest priorities.
3. τ is the $(k+1)$'st highest priority. If $k \geq n$ we set $\tau = 0$.
4. If $i \in S$, set $\hat{w}_i = \max\{w_i, \tau\}$, else set $\hat{w}_i = 0$.

Claim: Can maintain S, τ in streaming setting

$a_1, a_2, \ldots, a_n$

w 10, 5, 3, 2, 11

u 0.3 0.2 0.5 . 0.6

$q_i = \frac{w_i}{u_i}$

sort and take k highest elements

$\tau = k+1$ highest priority

$\hat{w}_i = \max\{w_i, \tau\}$.

Priority Sampling

Intuition: from uniform weight case

- Suppose $w_i = 1$ for all i. Then sampling k without repetition can be done via adaptation of reservoir sampling.
- A different approach: pick a uniformly random $r_i \in [0, 1]$ for each i. And pick top k in terms of r_i values (simulates random permutation) but can be done in streaming fashion. Many other distributions would work too and picking top k according to $1/r_i$ works too.
- Why $1/r_i$? What is the expected value of τ?

Priority Sampling: Properties

Lemma

$\mathbf{E}[\hat{w}_i] = w_i$. ✓

Priority Sampling: Properties

Lemma

$\mathbf{E}[\hat{w}_i] = w_i$.

Lemma

$Var[\hat{w}_i] = \mathbf{E}[\hat{v}_i]$ *where* $\hat{v}_i = \begin{cases} \tau \max\{0, \tau - w_i\} & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$

Useful: storing τ and w_i gives $Var[\hat{w}_i]$.

Priority Sampling: Properties

Lemma

$\mathbf{E}[\hat{w}_i] = w_i$.

Lemma

$$Var[\hat{w}_i] = \mathbf{E}[\hat{v}_i] \text{ where } \hat{v}_i = \begin{cases} \tau \max\{0, \tau - w_i\} & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

Useful: storing τ and w_i gives $Var[\hat{w}_i]$.

Lemma

If $k \geq 2$ *for any* $i \neq j$, $\mathbf{E}[\hat{w}_i \hat{w}_j] = w_i w_j$.

Lemma

Fix any set $C \subset [n]$. $\mathbf{E}\left[\prod_{i \in C} \hat{w}_i\right] = \prod_{i \in C} w_i$ *if* $|C| \leq k$ *and is* 0 *if* $|C| > k$.

Variance of subset sum

Lemma

If $k \geq 2$ for any $i \neq j$, $\mathbf{E}[\hat{w}_i \hat{w}_j] = w_i w_j$.

Consequence:

- Fix C. Unbiased estimator of $w(C \cap X)$ is $\hat{w}(C \cap S)$.
- Can we know the variance of the estimate to know if we are doing ok?
- $Var[\hat{w}(C \cap S)] = \sum_{i \in C \cap S} Var[\hat{w}_i] = \sum_{i \in C \cap S} \mathbf{E}[\hat{v}_i]$. Hence, storing τ and $\hat{w}_i$ values suffices to estimate the variance of the estimate.

Priority Sampling: Properties

Lemma

$\mathbf{E}[\hat{w}_i] = w_i$.

Priority Sampling: Properties

Lemma

$\mathbf{E}[\hat{w}_i] = w_i$.

Fix i. Let $A(\tau')$ be the event that the k'th highest priority among items $j \neq i$ is τ'.

Note that u_i is independent of τ'. Hence $i \in S$ if $q_i = w_i/u_i \geq \tau'$ and if $i \in S$ then $\hat{w}_i = \max\{w_i, \tau'\}$, otherwise $\hat{w}_i = 0$. To evaluate $\Pr[i \in S \mid A(\tau')]$ we consider two cases.

Case 1: $w_i \geq \tau'$. Here we have $\Pr[i \in S \mid A(\tau')] = 1$ and $\hat{w}_i = w_i$.

Case 2: $w_i < \tau'$. Then $\Pr[i \in S \mid A(\tau')] = \frac{w_i}{\tau'}$ and $\hat{w}_i = \tau'$.

In both cases we see that $E[\hat{w}_i] = w_i$.

Variance

Lemma

$$Var[\hat{w}_i] = \mathbf{E}[\hat{v}_i] \text{ where } \hat{v}_i = \begin{cases} \tau \max\{0, \tau - w_i\} & \text{if } i \in S \\ 0 & \text{if } i \notin S \end{cases}$$

Variance

Lemma

$$Var[\hat{w}_i] = \mathbf{E}[\hat{v}_i] \textit{ where } \hat{v}_i = \begin{cases} \tau \max\{0, \tau - w_i\} & \textit{if } i \in S \\ 0 & \textit{if } i \notin S \end{cases}$$

Fix i. We define $A(\tau')$ to be the event that τ' is the k'th highest priority among elements $j \neq i$.

Show that

$$E[\hat{v}_i \mid A(\tau')] = E[\hat{w}_i^2 \mid A(\tau')] - w_i^2.$$

Since u_i is independent of τ' we can remove conditioning

Variance

$$E[\hat{v}_i \mid A(\tau')] = E[\hat{w}_i^2 \mid A(\tau')] - w_i^2.$$

$$\begin{aligned}
\mathbf{E}[\hat{v}_i \mid A(\tau')] &= \Pr[i \in S \mid A(\tau')] \times \mathbf{E}[\hat{v}_i \mid i \in S \wedge A(\tau')] \\
&= \min\{1, w_i/\tau'\} \times \tau' \max\{0, \tau' - w_i\} \\
&= \max\{0, w_i\tau' - w_i^2\}.
\end{aligned}$$

Variance

$$E[\hat{v}_i \mid A(\tau')] = E[\hat{w}_i^2 \mid A(\tau')] - w_i^2.$$

$$\begin{aligned}
\mathbf{E}[\hat{v}_i \mid A(\tau')] &= \Pr[i \in S \mid A(\tau')] \times \mathbf{E}[\hat{v}_i \mid i \in S \wedge A(\tau')] \\
&= \min\{1, w_i/\tau'\} \times \tau' \max\{0, \tau' - w_i\} \\
&= \max\{0, w_i\tau' - w_i^2\}.
\end{aligned}$$

$$\begin{aligned}
\mathbf{E}\left[\hat{w}_i^2 \mid A(\tau')\right] &= \Pr[i \in S \mid A(\tau')] \times \mathbf{E}\left[\hat{w}_i^2 \mid i \in S \wedge A(\tau')\right] \\
&= \min\{1, w_i/\tau'\} \times (\max\{w_i, \tau'\})^2 \\
&= \max\{w_i^2, w_i\tau'\}.
\end{aligned}$$

Variance of subset sum

Lemma

If $k \geq 2$ *for any* $i \neq j$, $\mathbf{E}[\hat{w}_i \hat{w}_j] = w_i w_j$.

More generally

Lemma

Fix any set $C \subset [n]$. $\mathbf{E}\left[\prod_{i \in C} \hat{w}_i\right] = \prod_{i \in C} w_i$ *if* $|C| \leq k$ *and is* $\mathbf{0}$ *if* $|C| > k$.

Variance of subset sum

Lemma

If $k \geq 2$ *for any* $i \neq j$, $\mathbf{E}[\hat{w}_i \hat{w}_j] = w_i w_j$.

More generally

Lemma

Fix any set $C \subset [n]$. $\mathbf{E}\left[\prod_{i \in C} \hat{w}_i\right] = \prod_{i \in C} w_i$ *if* $|C| \leq k$ *and is* $\mathbf{0}$ *if* $|C| > k$.

Requires a proof by induction. See notes

Variance of subset sum

Lemma

If $k \geq 2$ for any $i \neq j$, $\mathbf{E}[\hat{w}_i \hat{w}_j] = w_i w_j$.

More generally

Lemma

Fix any set $C \subset [n]$. $\mathbf{E}\left[\prod_{i \in C} \hat{w}_i\right] = \prod_{i \in C} w_i$ if $|C| \leq k$ and is $\mathbf{0}$ if $|C| > k$.

Requires a proof by induction. See notes

Why is this interesting/non-obvious? In vanilla importance sampling the variables $\hat{w}_i$ are independent. However, here the variables are correlated because we choose exactly k. Nevertheless, they exhibit properties similar to independence.

Application of ℓ_2 sampling to F_p estimation

For $p > 2$ AMS-Sampling gives algorithm to estimate F_p using $\tilde{O}(n^{1-1/p})$ space. Optimal space is $\tilde{O}(n^{1-2/p})$.

- Use ℓ_2 sampling algorithm to generate $(i, |\tilde{x}_i|)$
- Estimate $\|x\|_2^2$
- Output $T = \|x_2\|^2 |\tilde{x}_i|^{p-2}$ as estimate

To simplify analysis/notation assume sampling is exact.

$\mathbf{E}[T] = \|x\|_2^2 \sum_i \frac{x_i^2}{\|x\|_2^2} |x_i|^{p-2} = \sum_i |x_i|^p$

$Var[T] \leq \|x\|_2^4 \sum_i \frac{x_i^2}{\|x\|_2^2} x_i^{2(p-2)} \leq \|x\|_2^2 \sum_i x_i^{2p-2} \leq n^{1-2/p} (\sum_i |x_i|^p)^2$.

Now do average plus median.