5 more sketching

Tuesday, February 4, 2020 1:54 AM

LogLog counters

Let's count up to n in binary. Need $\lg(n) = O(\log n)$ bits for a counter.

What if we allow for some error.

First idea: each time, we flip a coin and increment only if heads.

Then need to store $\frac{n}{2}$, so $\lg(n)-1$ bits. Still $O(\log n)$.

More advanced idea. Flip as many coins as the current value of counter k.

i.e. increment counter w.p. $\frac{1}{2^k}$.

Then in expectation, need 2^k items to increment the counter from k to $k+1$.

Then the expected number of items to produce a value k

is $1+2+4+\cdots+2^{k-1} = 2^k - 1$.

Thus, in $\lg(k)$ bits, can store $n = 2^k$.

Alternately, need $O(\lg \lg n)$ bits to approximate n.

Inspiration using hashing for non-idealized FM sketch.

Non-idealized Flajolet-Martin counting [1985]

(round up if necessary)

1. Pick h from 2-wise family $[n] \to [n]$ for n a power of 2
2. Maintain $X = \max_{i \in \text{stream}} \text{lsb}(h(i))$, where lsb is least significant 1-bit of a number.
3. Output 2^X.

$\text{lsb}(10110111100) = 3$

$\text{lsb}(10110000) = 6$

For fixed j, let $Z_j = |\{i \in \text{stream} \mid \text{lsb}(h(i)) = j\}|$, the number of i in stream with $\text{lsb}(h(i)) = j$.

Let $Z_{>j} = \bigcup_{\ell > j} Z_\ell$.

Let $Y_i = \begin{cases} 1 & \text{if } \text{lsb}(h(i)) = j \\ 0 & \text{else.} \end{cases}$

$\mathbb{E} Y_i = \frac{1}{2^{j+1}}$

$\text{Var}(Y_i) = \mathbb{E} Y_i^2 - (\mathbb{E} Y_i)^2 = \frac{1}{2^{j+1}} - \frac{1}{2^{2j+2}} < \frac{1}{2^{j+1}}$

Then $Z_j = \sum_{i \in \text{str}} Y_i$. $\mathbb{E} Z_j = \frac{t}{2^{j+1}}$, where $t = |\text{uniq}(\text{str})|$

$\mathbb{E} Z_{>j} = t\left(\frac{1}{2^{j+2}} + \frac{1}{2^{j+3}} + \cdots\right) = \frac{t}{2^{j+1}}$

and also

$$\mathrm{Var}(Z_j) = \mathrm{Var}\left(\sum Y_i\right) = \mathbb{E}\left(\sum Y_i\right)^2 - \left(\mathbb{E}\sum Y_i\right)^2$$

$$\mathbb{E}\left[(Y_1 + Y_2 + \cdots + Y_t)(Y_1 + \cdots + Y_t)\right] = \mathbb{E}\left[Y_1^2 + \cdots + Y_t^2 + 2\sum_{i_1 \neq i_2} Y_{i_1} Y_{i_2}\right]$$

$$\left(\mathbb{E}(Y_1 + \cdots + Y_t)\right)^2 = \left(\mathbb{E}Y_1 + \cdots + \mathbb{E}Y_t\right)^2 = (\mathbb{E}Y_1)^2 + \cdots + (\mathbb{E}Y_t)^2 + 2\sum_{i_1 \neq i_2} (\mathbb{E}Y_{i_1})(\mathbb{E}Y_{i_2})$$

by 2-independence $= 2\sum_{i_1 \neq i_2} \mathbb{E}(Y_{i_1} Y_{i_2})$

$$= \sum \mathbb{E}Y_i^2 - (\mathbb{E}Y_i)^2 = \sum \mathrm{Var}(Y_i) < \frac{t}{2^{j+1}}.$$

$$\Rightarrow \mathrm{Var}(Z_j) < \frac{t}{2^{j+1}}.$$

Now for $j^* = \lfloor \lg t - 5 \rfloor$, we have

$$\lg t - 6 \leq j^* \leq \lg t - 5$$

$$\frac{t}{2^{\lg t - 4}} \leq \mathbb{E}Z_{j^*} \leq \frac{t}{2^{\lg t - 5}}$$

$\lg t - 5 \leq j^* \leq \lg t - 4$

$$\frac{t}{2^{\lg t - 3}} \leq \mathbb{E}Z_{j^*} \leq \frac{t}{2^{\lg t - 4}}$$

$$16 \leq \mathbb{E}Z_{j^*} \leq 32$$

$$\mathbb{P}(Z_{j^*} = 0) \leq \mathbb{P}(|Z_{j^*} - \mathbb{E}Z_{j^*}| \geq 16) \leq \frac{\mathrm{Var}(Z_{j^*})}{256} < \frac{\frac{t}{2^{\lg t - 5}}}{256} = \frac{1}{8}$$

For $j = \lfloor \lg t + 5 \rfloor$, we have $j > \lg t + 4$

$$\mathbb{E}Z_{>j} \leq \frac{t}{2^{\lg t + 5}} = \frac{1}{32}$$

$$\mathbb{P}(Z_{>j} \geq 1) < \frac{1}{32} \quad \text{by Markov.}$$

$= \frac{1}{2^{c-2}}$, storing $j^* = \lfloor \lg t - c \rfloor$

Thus, with probability $1 - \frac{1}{8} - \frac{1}{32}$, the max lsb will be at least j^* but below J, in a constant range.

i.e. we get a $O(1)$ constant approximation with high probability.

(64)

Our estimate $\tilde{t}$ satisfies $\frac{t}{C} \leq \tilde{t} \leq Ct$ for some constant C,

for the max lsb.

Our estimate $\tilde{t}$ satisfies $\frac{t}{C} \le \tilde{t} \le Ct$ for some constant C,
with probability $1 - \frac{8}{C}$ using $O(\log \log C)$ bits space. for the max lb.
and $O(\log n)$ bits for the hash function.

Refine to $1+\varepsilon$ solution

Trivial solution TS stores first $\frac{C}{\varepsilon^2}$ distinct elements.
TS is a $(1+\varepsilon)$-solution if $t \le \frac{C}{\varepsilon^2}$.

Algorithm

1. Instantiate $TS_0, \dots, TS_{\lg n}$
2. Pick $g : [n] \to [n]$ from 2-wise family.
3. Feed i to $TS_{\text{lsb}(g(i))}$.
4. Output 2^{j+1} where $Q_j = |TS_j| \approx \frac{1}{\varepsilon^2}$

Let B_j be the number of distinct elements hashed by g to TS_j.
Then $\mathbb{E} B_j = \frac{t}{2^{j+1}} = Q_j$.

By Chebyshev $B_j = Q_j \pm O(\sqrt{Q_j})$ with good probability.
$= (1 \pm O(\varepsilon)) Q_j$ if $Q_j \approx \frac{1}{\varepsilon^2}$

$\left[1 \pm \left(\frac{O(\sqrt{Q_j})}{Q_j}\right)\right] Q_j$

$\frac{O(\frac{1}{\varepsilon})}{\frac{1}{\varepsilon^2}} = O(\varepsilon)$

Note that the B_j's decrease by factors of 2, and
with good prob. one of the Q_j's is close to $\frac{1}{\varepsilon^2}$.

Final space $\frac{C}{\varepsilon^2} \cdot (\lg n) \cdot (\lg n) = O\left(\frac{1}{\varepsilon^2} \lg^2 n\right)$ bits

Final space $\frac{1}{\varepsilon^2} \cdot (\lg n) \cdot (\lg n) = O\left(\frac{1}{\varepsilon^2} \lg^2 n\right)$ bits

- $\frac{1}{\varepsilon^2}$: # unique
- $\lg n$: storing entries + hash
- $\lg n$: copies of TS

Can do better. e.g. $O\left(\frac{1}{\varepsilon^2} \log\log n + \log n\right)$ for HyperLogLog ($\log n$: hash)

[Kane, Nelson, Woodruff, 2010, PODS]: $O\left(\frac{1}{\varepsilon^2} + \log n\right)$ optimal & achieveable.

Majority & Frequent Items

Ex. n people voting for m candidates.

Does any candidate have $> \frac{n}{2}$ votes?

i.e. Let $a_1, \ldots, a_n \in [m]$. Determine if $\exists s \in [m]$ s.t. s occurs $> \frac{n}{2}$ times?

Claim: Any deterministic streaming alg requires $\Omega(m, n)$ space, if we require that it output if there is a majority element, and if so, what it is.

proof. Suppose n is even and the last $n/2$ items are identical.

Every possible set of unique $\frac{n}{2}$ first items must have a different memory config, otherwise, can make mistake by choosing second half to belong to one subset but not the other.

If $\frac{n}{2} \geq m$, then $2^m - 1$ subsets $\log(2^m - 1) = \Omega(m)$

$\frac{n}{2} \leq m$, then $\geq \frac{m!}{(m - \frac{n}{2})!}$ subsets $\log\left(\frac{m!}{(m-\frac{n}{2})!}\right) = \Omega(n)$.

Loosening the requirement that we have a defined response if there is no majority element, we can do better.

i.e. can be false positives, but no false negatives.

Majority algorithm - With undefined behavior when no majority.

Initialize $B \leftarrow a_1$ and $c \leftarrow 1$.

Initialize $B \leftarrow a_1$ and $c \leftarrow 1$.

For a_i in $i \in \{2, \ldots, n\}$

If $B = a_i$, $c \leftarrow c+1$.

Else if $c > 0$, $c \leftarrow c-1$

Else if $c = 0$, $B \leftarrow a_i$, $c \leftarrow 1$.

} paired eliminations of items

If $c > 0$, output B.

If there was a majority item, it appears $> \frac{n}{2}$ times, and so cannot be eliminated by other items

Misra-Gries Algorithm Frequent

Initialize $B_1, \ldots, B_k = 0$ buckets and $c_1, \ldots, c_k = 0$ counters.

For $i \in [n]$,

If $\exists j$ s.t. $B_j = a_i$, $c_j \leftarrow c_j + 1$

Else

If $\exists j$ s.t. $B_j = 0$, $B_j \leftarrow a_i$, $c_j \leftarrow c_j + 1$

Else (Decrement)

For $j \in [m]$,

$c_j \leftarrow c_j - 1$

If $c_j = 0$, $B_j = 0$.

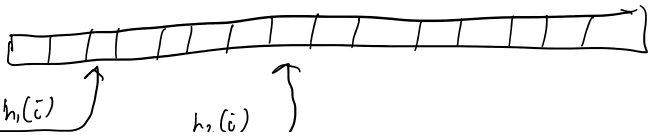
Theorem 6.2 At the end of Misra-Gries, for each B_k with true count x_k, $c_k \in \left[x_k - \frac{n}{k+1}, x_k\right]$. If some $s \neq B_k$ for any k, then $x_k \leq \frac{n}{k+1}$.

proof. Left as exercise. Idea, count number of decrements.

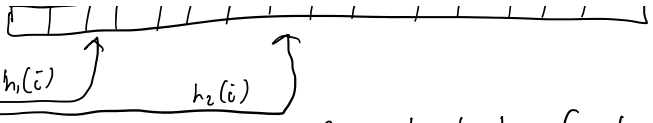
Count-Min-Sketch

Recall Bloom filters. Probabilistic set membership query.

Maintain bit-vector



T ⊥ ide... ;



Insert item i by setting bits corresponding to multiple ind. hash functions.

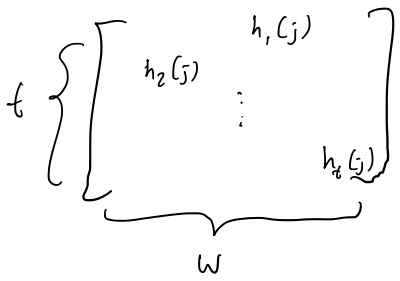
Query item i by checking if bits are set.

Might accidentally return yes because of hash collision, but unlikely given the right parameters.

Let's adapt this idea to frequency using the Count-Min sketch.

Let $a_1, a_2, \ldots \in [n]$, and let $\vec{x} \in \mathbb{R}^n$ be a vector specifying the frequencies of each item in $[n]$. We want to estimate $\vec{x}$.

Maintain $t \times w$ matrix of counters.



For each row, associate a hash function $h_j : [n] \to [w]$ independently drawn from a 2-wise family.

Insert item i by incrementing all counters $C_{j, h_j(i)}$ for $j \in [t]$.

Output $\text{PointQuery}(i) = \min_{j \in [t]} C_{j, H_j(i)}$.

Claim If $t > \lg\left(\frac{1}{\delta}\right)$ and $w \geq \frac{2}{\varepsilon}$, then

$$\mathbb{P}\left(\text{PointQuery}(i) \in \left[x_i - \varepsilon \|\vec{x}\|_1,\ x_i + \varepsilon \|\vec{x}\|_1\right]\right) \geq 1 - \delta.$$

proof. For any $j \in [n]$,

$$C_{j, H_j(i)} = x_i + \sum_{\substack{r : h_j(r) = h_j(i) \\ r \neq i}} x_r$$

$$= x_i + \sum_{r \neq i} \delta_r x_r,$$

$$= x_i + \underbrace{\sum_{r \neq i} \delta_r x_r}_{\text{noise}},$$

where δ_r is the indicator function with value 1 if $h_j(r) = h_j(i)$, 0 otherwise.

$$\mathbb{E} \sum_{r \neq i} \delta_r x_r = \frac{1}{w} \sum_{r \neq i} x_r \leq \frac{\varepsilon}{2} \|\vec{x}\|_1.$$

↑ 2-wise independence of h_j

By Markov's inequality & since $x_i \geq 0$,

$$\mathbb{P}\left(\text{noise} > \varepsilon \|\vec{x}\|_1\right) \leq \frac{1}{2}.$$

So $C_{j,H_j(i)} \geq x_i$ and w.p. $> \frac{1}{2}$, $C_{j,H_j(i)} \leq \varepsilon \|\vec{x}\|_1$.

Since we are repeating $t = \lg\left(\frac{1}{\delta}\right)$ times,

$$\mathbb{P}\left(\min_{j \in [t]} C_{j,H_j(i)} > x_i + \varepsilon \|x\|_1\right) = \mathbb{P}\left(\forall j \in [t],\ C_{j,H_j(i)} > \varepsilon \|x\|_1\right)$$

$$< \frac{1}{2^t} < \delta.$$

Only useful for heavy hitters $x_i > \varepsilon \|\vec{x}\|_1$, so useful for $\sim \frac{1}{\varepsilon}$ of the values at most.

Matrix sampling & sketches

Let $A \in \mathbb{R}^{m \times n}$
$B \in \mathbb{R}^{n \times p}$.

We want to approximate AB.
Naive approach $O(mnp)$ time.

Let $A(:,k)$ be the kth column of A
$B(k,:)$ be the kth row of B.

$B(k,:)$ be the kth row of B.

Then $AB = \sum_{k=1}^{n} A(:,k)\,B(k,:)$ (outer product)

Let's try to sample AB by taking components with prob. p_k.

i.e. Let $z = k$ w.p. p_k for $k \in [n]$, a random variable

Define $X = \frac{1}{p_z} A(:,z)\,B(z,:)$, a matrix r.v.

Then the entry-wise expectation

$$\mathbb{E}X = \sum_{k=1}^{n} \mathbb{P}(z=k)\,\frac{1}{p_k} A(:,k)\,B(k,:) = \sum_{k=1}^{n} A(:,k)\,B(k,:) = AB.$$

(this cancellation is the reason we scale by $\frac{1}{p_k}$)

Define $\mathrm{Var}(X) = \mathbb{E}\left(\|AB - X\|_F^2\right)$, the entry-wise variance.

Then $\mathrm{Var}(X) = \sum_{i=1}^{m}\sum_{j=1}^{p} \mathrm{Var}(x_{ij}) = \sum_{ij} \mathbb{E}(x_{ij}^2) - \mathbb{E}(x_{ij})^2 = \left(\sum_{ij}\sum_{k=1}^{n} p_k \cdot \frac{1}{p_k^2} a_{ik}^2 b_{kj}^2\right) - \|AB\|_F^2$.

doesn't matter for minimizing p_k.

We want to minimize variance, by choosing appropriate p_k.

$$\sum_{ij}\sum_{k} p_k \cdot \frac{1}{p_k^2} a_{ik}^2 b_{kj}^2 = \sum_{k} \frac{1}{p_k}\left(\sum_i a_{ik}^2\right)\left(\sum_j b_{kj}^2\right) = \sum_k \frac{1}{p_k} |A(:,k)|^2\,|B(k,:)|^2$$

Note that for any $c_k \geq 0$, $\sum_k \frac{c_k}{p_k}$ is minimized by $p_k \propto \sqrt{c_k}$.

(proof by taking derivatives $p_1 + \cdots + p_n = 1$)

$$\frac{\partial f}{\partial p_k} = \frac{\partial}{\partial p_k}\left(\frac{c_1}{(1-(p_2+\cdots+p_n))^2} + \frac{c_k}{p_k}\right)$$

$$= \frac{c_1}{(1-(p_2+\cdots+p_n))^2} - \frac{c_n}{p_k^2} = 0$$

$$\frac{p_k}{1-(p_2+\cdots+p_n)} = \sqrt{\frac{c_k}{c_1}}$$

$$p_k = \sqrt{c_k} \cdot \frac{1-(p_2+\cdots+p_n)}{\sqrt{c_1}} \quad \forall\, k \neq 1. \;\Big)$$

Thus, we want to pick $p_k \sim |A(:,k)|\,|B(k,:)|$.

Note, when $B = A^T$, $p_k \sim |A(:,k)|^2$, the squared length of the columns.

Even if $B \neq A^T$, we can still use that as an easy to analyze upper bound.

Use $p_k = \dfrac{|A(:,k)|^2}{\|A\|_F^2}$.

$$\Rightarrow \mathbb{E}\left(\|AB - X\|_F^2\right) = \mathrm{Var}(X) \le \|A\|_F^2 \sum_k |B(k,:)|^2 = \|A\|_F^2 \|B\|_F^2.$$

Repeat with s independent trials, getting $X_1, \ldots, X_s$.

Then $\mathrm{Var}(\bar{X}) = \frac{1}{s}\sum_{i=1}^{s} X_i = \frac{1}{s}\mathrm{Var}(X) \le \frac{1}{s}\|A\|_F^2 \|B\|_F^2$.

$$\begin{bmatrix} & A & \\ m & \times & n \end{bmatrix} \begin{bmatrix} & B & \\ n & \times & p \end{bmatrix} \approx \begin{bmatrix} \text{Sampled} \\ \text{scaled} \\ \text{columns} \\ \text{of} \\ A \\ m \times s \end{bmatrix} \begin{bmatrix} \text{Corresponding} \\ \text{scaled rows of} \\ B \\ s \times p \end{bmatrix}$$

Note $\dfrac{1}{s}\sum_{i=1}^{s} X_i = \dfrac{1}{s}\left(\dfrac{A(:,k_1)B(k_1,:)}{p_{k_1}} + \cdots + \dfrac{A(:,k_s)B(k_s,:)}{p_{k_s}}\right)$

$= CR$, where

C has columns $\dfrac{A(:,k_i)}{\sqrt{s\,p_{k_i}}}$. Note $\mathbb{E}(CC^T) = AA^T$

R has rows $\dfrac{B(k_i,:)}{\sqrt{s\,p_{k_i}}}$. $\mathbb{E}(R^TR) = B^TB$

r rows

R has rows $\dfrac{B(k_i,:)}{\sqrt{s p_{k_i}}}$ $\qquad \mathbb{E}(R^T R) = B^T B$.

Theorem 6.5 Suppose $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times p}$. The product AB can be estimated by CR as given above, and the error is bounded by

$$\mathbb{E}\left(\|AB - CR\|_F^2\right) \leq \frac{\|A\|_F^2 \|B\|_F^2}{s}.$$

Thus, to ensure $\mathbb{E}\left(\|AB - CR\|_F^2\right) \leq \varepsilon^2 \|A\|_F^2 \|B\|_F^2$, it suffices to make $s \geq \frac{1}{\varepsilon^2}$. If $\varepsilon = \Omega(1)$, $s = O(1)$, so CR can be computed in $O(mp)$ time.