13. Majority and frequent items

Friday, October 8, 2021 3:13 PM

Last time: Frequency moment sketches

Today: Majority + Frequent item items

Majority item.

Ex n people voting for m candidates

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

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

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

proof. Suppose n is even and the last $\frac{n}{2}$ items are identical.

Every possible set of unique $\frac{n}{2}$ first items must have a different memory config, otherwise we can make a 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 $\log(2^m - 1) = \Omega(m)$ bits

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

Majority Alg – With undefined behavior when no majority

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

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

- 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 elim. of items

If $c > 0$, output B

If $\exists$ majority item, appears $> \frac{n}{2}$ times, and so it cannot be eliminated.

Misra - Gries Algorithm Frequent

Initialize $B_1, \dots, B_k = 0$ buckets, and $c_1, \dots, c_k = 0$ 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 all j

        $c_j \leftarrow c_j - 1$

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

Thm. 6.2 At the end of Misra-Gries, for each B_k with true count x_k,

$$c_k \in \left[x_k - \frac{n}{k+1}, x_k \right]$$

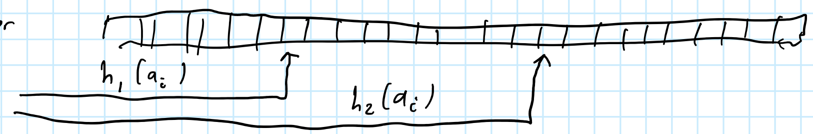
If some $s \neq B_k$ for any k, then $x_k \leq \frac{n}{k+1}$.

Count - Min - Sketch

Recall Bloom filters. Probabilistic set membership query.

Maintain bit vector

Insert item a_i

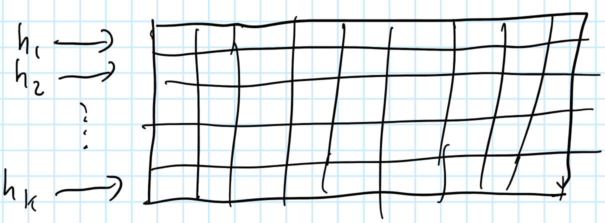


by setting bits corresponding to multiple ind. hash functions.

Query item s by checking if bits $h_1(s), \dots, h_k(s)$ are set.

Might accidentally return yes because of hash collisions, but unlikely given right params

Can also imagine each hash function having its own range.



Let's adapt this to freq. counting using Count-Min Sketch.

Let $a_1, a_2, \dots, a_n \in [m]$ and let $\vec{x} \in \mathbb{R}^m$ be the freq. of each item in $[m]$.

We want to estimate $\vec{x}$.

Maintain $t \times w$ matrix of counters

$$C = t\left\{ \underbrace{\begin{bmatrix} & h_1(j) & \\ h_2(j) & & \\ & \vdots & \\ & & h_t(j) \end{bmatrix}}_{w} \right.$$

For each row, associate a hash function $h_j : [m] \to [w]$ 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

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

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

$$C_{j, h_j(i)} = x_i + \sum_{\substack{r \text{ s.t.} \\ h_j(r) = h_j(i) \\ r \neq i}} x_r = x_i + \overbrace{\sum_{r \neq i} \delta_r x_r}^{\text{noise}}$$

where $\delta_r = \begin{cases} 1 & \text{if } h_j(r) = h_j(i) \\ 0 & \text{otherwise} \end{cases}$

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

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

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

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

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

$$\text{Prob}\left(\min_{j \in [t]} C_{j, h_j(i)} > x_i + \varepsilon \|\vec{x}\|_1 \right) = \text{Prob}\left(\forall\ j \in [t],\ C_{j, h_j(i)} > \varepsilon \|x\|_1 \right)$$

$$\mathrm{Prob}\left(\min_{j\in[t]} C_{j,h_j(i)} - x_i + \varepsilon \|x\|_1\right) = \mathrm{Prob}\left(\forall\ j\in[t],\ C_{j,h_j(i)} - 2\|x\|_1\right)$$

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

Only useful for heavy hitters (very frequent items) $x_i > \varepsilon \|\vec{x}\|_1$,

so useful for $\sim \frac{1}{\varepsilon}$ of the values at most.

Set-similarity

Let $A, B \subset U$ be two subsets. Let $n = |A \cup B|$.

Then Jaccard Similarity $J = J(A,B) = \dfrac{|A\cap B|}{|A \cup B|}$ is a measure of ~~sets~~ set similarity

MinHash (idealized)

1. Let $h_i : U \to [q]$ be ind. unif. random hash functions, $i \in [k]$

   using oracle actual random hash functions

2. Let
$$\delta_i = \begin{cases} 1, & \min_{a\in A} h_i(a) = \min_{b \in B} h_i(b) \\ 0, & \text{else.} \end{cases}$$

3. Then output $\hat{J} = \frac{1}{k}\sum_{i=1}^{k} \delta_i$ as estimate for J.

Claim: If $q > \frac{kn^2}{\delta}$, and $k > \frac{2}{\varepsilon^2 \delta}$, then

$$\mathrm{Prob}\left(|\hat{J} - J| > \varepsilon\right) < \delta.$$

proof. Recall that if h_i is a universal hash function, all of $h_i(x)$, $x \in A\cup B$ are distinct with probability at least $1 - \frac{\delta}{2k}$. (signatures)

By union bound, all h_i have no collisions w.p. at least $1 - \frac{\delta}{2}$.

Then $h_i(a) = h_i(b)$ only if $a = b$.

$$\Rightarrow \min_{a\in A} h_i(a) = \min_{b\in B} h_i(b) \text{ only if } a = b \in A\cap B.$$

Clearly, the converse also holds, so $\mathbb{E}\,\delta_i = J$, and $\mathbb{E}\,\hat{J} = J$, $\mathrm{Var}(\delta_i) \leq J$.

Then $\mathrm{Prob}\left(|\hat{J} - J| \geq \varepsilon\right) \leq \dfrac{\mathrm{Var}(\hat{J})}{\varepsilon^2} \leq \dfrac{1}{\varepsilon^2}\cdot\dfrac{1}{k}\cdot J < \dfrac{\delta J}{2} \leq \dfrac{\delta}{2}.$ $\square$

Thus, we need $O\left(\frac{1}{\varepsilon^2\delta}\log\frac{kn^2}{\delta}\right) = O\underbrace{\left(\frac{1}{\varepsilon^2}\log\frac{n}{\varepsilon\delta}\right)}_{\text{not tight at all.}}$

Aside: Can do better in idealized setting $O\left(\frac{1}{\varepsilon^2}\left(\log\log n + \log\frac{1}{\varepsilon}\right)\right)$
[Yu, Weber, TKDE, 2020] for ε additive error

Aside: We assume hash functions were fully random

Unfortunately, cannot just use 2- or 4-wise hash families, but need a stronger cond., i.e. that any item is equally likely to be the minimum.

A new criterion called min-wise independence. [Indyk, 1999]

$\hookrightarrow$ almost achievable for ℓ-wise ind hash functions for ℓ large enough.