Lec31-count-distinct-and-MinHash

Sunday, December 3, 2023 7:35 PM

Distinct-element sketching

Consider sequence $a_1, a_2, \ldots, a_n \in [m]$, with $n, m \gg 0$.

Want to know number of unique items, $F_0 = |\text{uniq}\{a_1, \ldots, a_n\}|$

$O(m)$ space solution — bit vector

$O(n \log m)$ space sol - store list of items seen

Goal: $O(\log n \cdot \log m)$ space.

Thm: (lower bound). Any exact deterministic algorithm must use at least m bits of memory on some seq. of length $n+1$.

proof. Assume that an alg uses $< m$ bits of memory on all such seq.

Recall that the power set $|\mathcal{P}([m])| = 2^m$ and $\text{uniq}\{a_1, \ldots, a_n\}$ can be any subset except $\emptyset$, so $2^m - 1$ possibilities.

But, we only have 2^{m-1} memory states possible since we use $< m$ bits.

Thus, two diff. subsets $S_1, S_2 \in \mathcal{P}([n]) \setminus \emptyset$ must have same memory state.

$\Rightarrow |S_1| = |S_2|$ because otherwise our alg would be wrong.

Let $b \in S_1$ but $b \notin S_2$. Then $S_1 \cup \{b\} = S_1$ so $|S_1 \cup \{b\}| = |S_1|$

and $S_2 \cup \{b\} \neq S_2$, so $|S_2 \cup \{b\}| = |S_2| + 1$.

But alg will have same mem state after adding b, so it must be wrong on one of them

Consider Idealized Streaming Algorithm (ISA)

1. Pick random hash function $h : [m] \to [0,1]$ (where $h(x) \sim \text{Unif}[0,1]$ iid)
2. Calculate $z = \min_{i \in \text{stream}} h(a_i)$
3. Output $\frac{1}{z} - 1$.

Let $S = \text{uniq}\{a_1, \ldots, a_n\} = \{b_1, \ldots, b_d\}$ ← d distinct elements

$h(b_1), \ldots, h(b_d) = X_1, \ldots, X_d$ i.i.d. Unif $[0,1]$

$Z = \min_{i=1}^{d} \{X_i\}$

Lemma: If $X : \Omega \to [0, +\infty)$ is a nonnegative r.v., then

$$\mathbb{E}X = \int_{[0,+\infty)} \text{Prob}(X > y)\, dy$$

Claim: $\mathbb{E}Z = \frac{1}{d+1}$

proof: $\mathbb{E}Z = \int_0^\infty \text{Prob}(Z > \lambda)\, d\lambda = \int_0^1 \text{Prob}(\forall i,\ X_i > \lambda)\, d\lambda$

$$= \int_0^1 \prod_{i=1}^{d} \text{Prob}(X_i > \lambda)\, d\lambda = \int_0^1 (1-\lambda)^d\, d\lambda = \left[\frac{-(1-\lambda)^{d+1}}{d+1}\right]_0^1 = \frac{1}{d+1}$$

Claim: $\mathbb{E}Z^2 = \frac{2}{(d+1)(d+2)}$

proof: $\mathbb{E}Z^2 = \int_0^1 \text{Prob}(Z^2 > \lambda)\, d\lambda = \int_0^1 \text{Prob}(Z > \sqrt{\lambda})\, d\lambda$

$$= \int_0^1 (1-\sqrt{\lambda})^d\, d\lambda = 2\int_1^0 u^d(u-1)\, du = 2\int_1^0 (u^{d+1} - u^d)\, du$$

let $u = 1 - \sqrt{\lambda}$

$u^2 - 2u + 1 = \lambda$

$(2u-2)\, du = d\lambda$

$$= 2\left[\frac{1}{d+2}u^{d+2} - \frac{1}{d+1}u^{d+1}\right]_1^0$$

$$= 2\left[\frac{-1}{d+2} + \frac{1}{d+1}\right] = \frac{2}{(d+1)(d+2)}$$

Thus, $\text{Var}(Z) = \mathbb{E}Z^2 - (\mathbb{E}Z)^2 = \frac{d}{(d+1)^2(d+2)} < \frac{1}{(d+1)^2}$.

Averaging algorithm (AA)

1. Run $q = \frac{1}{\quad}$ ISAs in parallel

1. Run $q = \frac{1}{\varepsilon^2 \eta}$ ISAs in parallel.

2. $\bar{Z} = \frac{1}{q} \sum_{i=1}^{q} Z_i$, where Z_i comes from ISA.

3. Output $\frac{1}{\bar{Z}} - 1$.

Then $\mathbb{E}(\bar{Z}) = \frac{1}{d+1}$, $\mathrm{Var}(\bar{Z}) = \frac{1}{q} \cdot \frac{d}{(d+1)^2 (d+2)} < \frac{1}{q(d+1)^2}$

By Chebyshev, $\mathrm{Prob}\left(\left| \bar{Z} - \frac{1}{d+1} \right| > \frac{\varepsilon}{d+1} \right) < \frac{(d+1)^2}{\varepsilon^2} \cdot \frac{1}{q(d+1)^2} = \eta$.

Claim: $\mathrm{Prob}\left(\left| \frac{1}{\bar{Z}} - 1 \right| - d > O(\varepsilon) d \right) < O(\eta)$.

proof: $\mathrm{Prob}\left(\left| \bar{Z} - \frac{1}{d+1} \right| > \frac{\varepsilon}{d+1} \right) < \eta$ $\iff$ w.p. $1-\eta$, $|\bar{Z}| \le \frac{1+\varepsilon}{d+1}$

$\mathrm{Prob}\left(\left| \bar{Z}d + \bar{Z} - 1 \right| > \varepsilon \right) < \eta$

$\mathrm{Prob}\left(\left| \frac{1}{\bar{Z}} - d - 1 \right| > \frac{\varepsilon}{|\bar{Z}|} \right) < \eta$

$\Rightarrow \mathrm{Prob}\left(\left| \frac{1}{\bar{Z}} - d - 1 \right| > \frac{\varepsilon(d+1)}{1+\varepsilon} \right) < \eta$

$O(\varepsilon)\, d$ $\quad O(\eta)$

With high prob, our estimator is within a factor of $1 + O(\varepsilon)$ of $1+d$.

Space complexity $O\left(\frac{1}{\varepsilon^2 \eta} \cdot \text{space}[0,1] \right)$.

Can do even better analysis, but this is core of idea of LogLog family of estimators.

Can do even better analysis, but this [illegible]

LogLog family of estimators.

Set-similarity

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

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

MinHash (idealized)

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

↑ using oracle access to 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

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

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 $\mathbb{E}\,\hat{J} = J$, $\text{Var}(\delta_i) \le J$.

Then $\text{Prob}\left(|\hat{J} - J| \ge \varepsilon \right) \le \frac{\text{Var}(\hat{J})}{\varepsilon^2} \le \frac{1}{\varepsilon^2} \cdot \frac{1}{k} \cdot J < \frac{\delta J}{2} \le \frac{\delta}{2}$. ∎

Thus, we need $O\left(\frac{1}{\varepsilon^2 \delta} \log \frac{kn^2}{\delta} \right) = O\left(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon \delta} \right)$

not right at all.

Aside: Can do better in idealized setting $O\left(\frac{1}{\varepsilon^2}\left(\log\log n + \log\frac{1}{\varepsilon}\right)\right)$

[Yi, Weber, THDE, 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]

↳ almost achievable for ℓ-wise ind. hash functions for ℓ large enough.