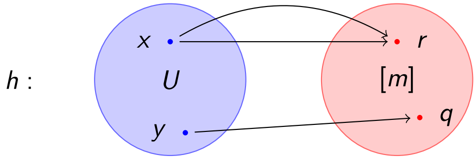
Streaming 2: Distinct element count

Philip Bille Inge Li Gørtz Eva Rotenberg

Reminder: hashing



~~Kiddie definition: A hash function is a function from U to $[m]$.~~

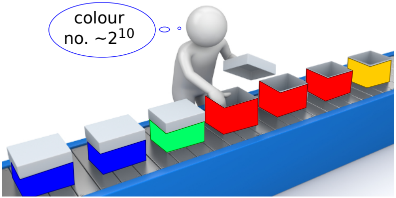
A hash function is a random variable in the set of functions $U \to [m]$.

Question: If $|U| = u$ and $|[m]| = m$, how many functions $U \to [m]$?

In practise, h is chosen uniform at random from a subset of $f : U \to [m]$.

2-independent hashing: For all $x \neq y \in U$, $q, r \in [m]$,
$P[h(x) = r \wedge h(y) = q] = \frac{1}{m^2}$.

Distinct element count



$z \leftarrow 0$,

for a_i in stream **do**

$\quad z = \max\{z, 0s(h(a_i))\}$

end

return $2^{z+0.5}$

Imagine you want to count element types (e.g. colours, see figure).

Challenge: A random dice roll that depends on the input.

Solution: Hashing.

Take a 2-independent hash function h.

Use $z =$ the number of trailing 0s in the hash values $h(x)$ seen so far.

Estimate: count $\simeq 2^{z+\frac{1}{2}}$. (we denote this $\hat{d}$, estimator of d)

Question: After seeing one element, what is the expected value of $\hat{d}$?

A Lower Bound

Assume we have an algorithm taking up s bits space and deterministically, exactly able to report the number of distinct elements.

Then, given any binary sequence x of length n, we can do the following:

Let the algorithm stream through a sequence consisting of $i : x_i = 1$.

Example: $x = 1001101$ Stream: $1, 4, 5, 7$.

Then, the state of the algorithm must be some configuration reflecting this information.

Now, regardless of what x was, we can recover x by streaming the following sequence: $1, 2, 3, 4, \ldots$, each time noticing whether the number of distinct elements goes up.

Thus, the state of the algorithm must have been able to distinguish between all different strings of length n.$\Rightarrow s = n$.

Exercise: How does this encoding encode $x' = 0100111 \in \{0, 1\}^7$?

The Median Trick

Lemma: $\hat{d}$ deviates from d by a factor 3 with prob. $\leq 2\frac{\sqrt{2}}{3}$.

Not very impressive. Still interesting!

What if we run k independent copies of the algorithm and return the median, m?

$m > 3d$ means $k/2$ of the copies exceed $3d$.

Expected: only $k\frac{\sqrt{2}}{3}$ exceed $3d$.

Since they are independent, we can use Chernoff. $\Rightarrow$ prob. $2^{-\Omega(k)}$.

Distinct element count: Analysis.

How well does $\hat{d} = 2^{z+\frac{1}{2}}$ estimate d?

$X_{r,j}$: indicator variable for $\geq r$ zeros in the hash value $h(j)$.

$\mathbb{E}[X_{r,j}] = P[r \text{ coinflips turn head}] = \left(\frac{1}{2}\right)^r$.

$Y_r = \sum_{j \in \text{stream}} X_{r,j}$: number of seen elements with $\geq r$ 0s.

$\mathbb{E}[Y_r] = d \cdot \mathbb{E}[X_{r,*}] = \frac{d}{2^r}$

$Var[Y_r] = \sum_j Var[X_{r,j}] \leq \sum_j \mathbb{E}[X_{r,j}^2] = \sum_j \mathbb{E}[X_{r,j}] = \frac{d}{2^r}$ $(j \in \text{stream})$

$P[Y_r > 0] = P[Y_r \geq 1] \overset{\text{Markov}}{\leq} \frac{\mathbb{E}[Y_r]}{1} = \frac{d}{2^r}$

$P[Y_r = 0] \leq P[|Y_r - \mathbb{E}[Y_r]| \geq \frac{d}{2^r}] \overset{\text{Chebysh.}}{\leq} \frac{\mathbb{E}[Y_r]}{(d/2^r)^2} \leq \frac{1}{(d/2^r)}$

Now, the probability of $\hat{d}$ being within a factor 3 of d.

$P[\hat{d} \geq 3d] = P[z \geq a]$ for some a with $2^{a+1/2} \geq 3d$.

$= P[Y_a > 0] \leq \frac{d}{2^a} = \frac{3 \cdot d \cdot \sqrt{2}}{3 \cdot 2^a \cdot \sqrt{2}} = \frac{\sqrt{2}}{3} \cdot \frac{3d}{2^{a+\frac{1}{2}}} \leq \frac{\sqrt{2}}{3}$.

Similarly, $P[\hat{d} \leq d/3] \leq \frac{\sqrt{2}}{3}$.