Lab 9

Hashing

Birthday paradox

How many people do we need to have at least two of them sharing a birthday with 0.5 probability.

Birthday paradox

How many people do we need to have at least two of them sharing a birthday with 0.5 probability.

Define E_{ij} be the event people i and j have different birthdays.

$$\Pr[E_{ij}] = \frac{364}{365}$$

If there are n=23 people, the probability that all of them have different birthdays is

$$\Pr[\cap_{i,j} E_{ij}] \approx \Pr\left[E_{ij}\right]^{n(n-1)/2} = \frac{364}{365}^{253} \approx 0.5$$

Not exact, why?

Birthday paradox

Real probability: $$P = 1 \times \frac{364}{365} \times \frac{363}{365} \times \dots \frac{366-n}{365}$$

With Most Useful Inequality $1 + x \approx e^x$, we get

$$P = \left(1 - \frac{1}{365}\right) \times \left(1 - \frac{2}{365}\right) \times \dots \left(1 - \frac{n-1}{365}\right) \approx \frac{1}{e^{(1+2+\dots n-1)/365}} = \frac{1}{e^{n(n-1)/730}}$$

Plug in n=23, P=0.49999.

- Generalize the problem to pick n people from T items, to have collision probability being at least 50%,

$$\frac{1}{e^{n(n-1)/2T}} = 0.5 \implies n^2 \approx -2 \cdot \ln\left(\frac{1}{2}\right) \cdot T$$

$\implies$ A hash function is likely to have collision with only $\sqrt{T}$ distinct elements.

2 wise independent hash functions

A 2-wise independent hash function $f : [m] \rightarrow [T]$ is a randomized function that, for any 2 distinct elements $e_1, e_2 \in [m]$ and any 2 possible values $t_1, t_2 \in [T]$,

$$\Pr[f(e_1) = t_1 \, and \, f(e_2) = t_2] = \frac{1}{T^2}$$

Lemma: Define $f(j) = a \cdot j + b \mod T$, where a and b are chosen uniformly and independently from [T]. If T is prime, then f(j) is 2-wise independent.

- Proof sketch: Consider any distinct $e_1, e_2 \in [m]$, and any $t_1, t_2 \in [T]$. What are the values of a and b when the following holds?

$$a \cdot e_1 + b \equiv t_1 \mod T, \, and \, a \cdot e_2 + b \equiv t_2 \mod T$$

L_2 norm estimation

Let x_j be the number of occurrences of element j in a stream with m possible distinct elements. The L2 norm of the stream is defined as follows:

$$||x||_2 = \left(\sum_{j \in [m]} |x_j|^2 \right)^{1/2}$$

Exact calculation requires recording the frequencies of all elements. => O(m) memory usage.

L_2 norm estimation

$$||x||_2 = \left(\sum_{j \in [m]} |x_j|^2 \right)^{1/2}$$

Algorithm:

- For each element j, we choose r_j to be either 1 or -1 independently with equal probability.
- Make a pass over the stream and compute the following

$$Z = \sum_{j \in [m]} r_j x_j$$

- Output Z^2 as the answer.

$$E[Z^2] = E\left[\left(\sum_{j \in [m]} r_j x_j \right)^2 \right] = \sum_{j_1, j_2} E[r_{j_1} r_{j_2} x_{j_1} x_{j_2}]$$

L_2 norm estimation

$$E[r_{j_1} r_{j_2}] = 1 \text{ when } j_1 = j_2, \text{ and } 0 \text{ otherwise}$$

Therefore,

$$E\left[Z^2\right] = E\left[\left(\sum_{j \in [m]} r_j x_j\right)^2\right] = \sum_{j_1, j_2} E[r_{j_1} r_{j_2} x_{j_1} x_{j_2}] = \sum_j x_j^2 = ||x||_2^2$$

$$Var\left[Z^2\right] \leq E\left[Z^4\right] = \sum_{j_1, j_2, j_3, j_4} E[r_{j_1} \ldots r_{j_4}] x_{j_1} \ldots x_{j_4} \leq \binom{4}{2} \sum_{j_1, j_2} x_{j_1}^2 x_{j_2}^2 = 6E\left[Z^2\right]^2$$

$$E[r_{j_1} r_{j_2} r_{j_3} r_{j_4}] = 0 \text{ when some j appears exactly one or three times, and } 1 \text{ otherwise}$$

L_2 norm estimation

$$E[Z^2] = E\left[\left(\sum_{j\in[m]} r_j x_j\right)^2\right] = \sum_{j_1,j_2} E[r_{j_1} r_{j_2} x_{j_1} x_{j_2}] = \sum_j x_j^2 = ||x||_2^2$$

2-wise independence

4-wise independence

$$Var[Z^2] \le E[Z^4] = \sum_{j_1,j_2,j_3,j_4} E[r_{j_1} \dots r_{j_4}] x_{j_1} \dots x_{j_4} \le \binom{4}{2} \sum_{j_1,j_2} x_{j_1}^2 x_{j_2}^2 = 6E[Z^2]^2$$

$E[r_{j_1} r_{j_2} r_{j_3} r_{j_4}] = 0$ when some j appears exactly one or three times, and 1 otherwise

L_2 norm estimation

Then we can use Chebyshev's inequality

$$\Pr\left[\left|Z^2 - E\left[Z^2\right]\right| \geq \epsilon||x||_2^2\right] \leq \frac{Var\left[Z^2\right]}{\epsilon^2||x||_2^4} = \frac{6}{\epsilon^2}$$

Finally, boost the success probability by repeating (run $\frac{6}{\epsilon^2\delta}$ independent instances in parallel) and taking the average.

- This works because the variance is reduced linearly.
- Total memory usage: $O\left(\frac{\ln n}{\epsilon^2\delta}\right)$, where n is the length of the stream.