Unit 11: Compression of databases

Problem Statement

Given entries $x_1, \ldots, x_n$ from $\mathcal{X}$, ($|\mathcal{X}|$ is very large in comparison to n)
how do we store them efficiently (small memory) to answer (efficiently) queries about $x_1, \ldots, x_n$?

→ Search query: $y \in \mathcal{X}$ and need check if $y \in \{x_1, \ldots, x_n\}$ or not (Yes or No)

Naive Scheme 1:
* Compress and store each data point x_i, $1 \le i \le n$. (requires $n \log |\mathcal{X}|$ bits of storage)
* To search for a $y \in \mathcal{X}$, we need to go over the entire list. (Requires n steps)

Naive Scheme 2:
* Consider an array A of $|\mathcal{X}|$ bits. Set
$$A[x] = \begin{cases} 1, & \text{if } x \in \{x_1, \ldots, x_n\} \\ 0, & \text{o.w.} \end{cases}$$
(Requires $|\mathcal{X}|$ bits of memory)
* For search, check if $A[y] = 1$ (requires only 1 step)

A A hash based scheme for compression of databases

hash → randomized hash (keyed hash function)

Recall: Collision Resistant Hash function

(*) $F: \mathcal{X} \to \{0,1\}^\ell$
$$\mathbb{P}(F(x) = F(x')) \le 2^{-\ell}, \quad x \neq x'.$$

(A randomly chosen function satisfies this property)

Definition (Universal hash family)
A family of mappings $\mathcal{F}$ from $\mathcal{X}$ to $[m]$ is a UHF if

$$\mathbb{P}(F(x) = F(x')) \le 1/m, \quad \forall\ x \neq x'$$

where $F \sim \text{unif}\{\mathcal{F}\}$.

Ex 1: $\mathcal{F} = \{f : \mathcal{X} \to [m]\}$ (the set of all mappings)
$|\mathcal{F}| = m^{|\mathcal{X}|}$

Ex 2: $\mathcal{X} = \{0,1\}^k$, $m = 2$
$\mathcal{F} = \{f_v : v \in \{0,1\}^k\}$, $f_v(x) = \sum_{i=1}^{k} v_i x_i \bmod 2$ (GF(2))

$|\mathcal{F}| = 2^k = |\mathcal{X}|$.

Also, $F \sim \text{unif}\{\mathcal{F}\}$,

$$\mathbb{P}\big(F(x) = F(x')\big) = \mathbb{P}\Big(\sum_{i=1}^{k} B_i x_i = \sum_{i=1}^{k} B_i x_i' \bmod 2\Big)$$

where $B_1, \ldots, B_k$ are random bits

$\sum_{i=1}^{k} B_i (x_i \oplus x_i') \bmod 2 = 0$ for $x \neq x'$

Can happen with prob. $= 1/2$. Why?

Therefore, $\mathbb{P}\big(F(x)=F(x')\big) = \frac{1}{2} = \frac{1}{m} \Rightarrow \mathcal{F}$ is UHF.

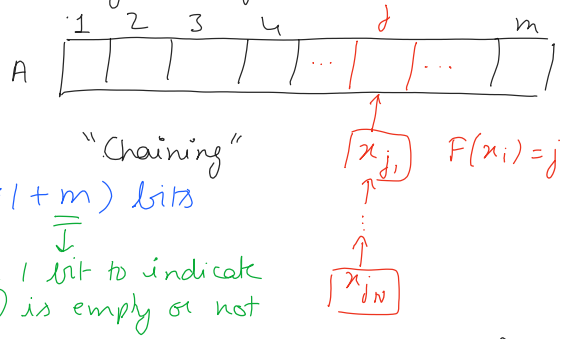
Ex 3 If we use r-fold product of the previous family (the same as $F_1, \ldots, F_r$ indep. random hash, where $F_i \sim \text{unif}(\mathcal{F})$), then $|\mathcal{F}^{\otimes r}| = 2^{rk}$ and $\mathbb{P}(F(x) = F(x')) = \prod_{i=1}^{r} \mathbb{P}(F_i(x) = F_i(x'))$

$$= \frac{1}{2^r}.$$

Scheme 3

* Let $\mathcal{F}$ be a UHF with range-size m, and $F \sim \text{unif}(\mathcal{F})$

* Consider an array A of length m, where $A[j]$ is a "linked list" containing all x_i's s.t. $F(x_i) = j$, $1 \le j \le m$.



Memory: $(n \log |\mathcal{X}| + m)$ bits

We use 1 bit to indicate if $A[j]$ is empty or not

* For a query y, we go to the $A[F(y)]$ and look for y in all the elements stored there.

→ Suppose m is chosen large enough so that with prob. 0.999 there is at most 1 element in a cell $A[j]$. (*)

⇒ with prob. 0.999 the computational complexity of search is $O(1)$.

Claim. $m = 1000\binom{n}{2}$ satisfies (*)

Proof. $\mathbb{P}\big(\exists\, i,j \text{ s.t. } x_i \neq x_j \text{ and } F(x_i) = F(x_j)\big)$

$$\le \sum_{i,j : x_j \neq x_i} \mathbb{P}\big(F(x_i) = F(x_j)\big)$$

$\leq 1/m$

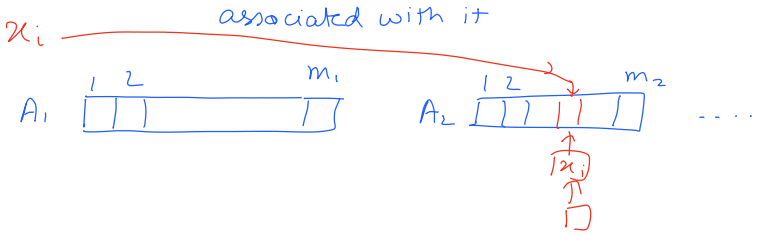
$= \binom{n}{2}\frac{1}{m} \leq \frac{1}{1000}$ if $m \geq 1000\binom{n}{2}$.

Therefore, we have a scheme with $O(1)$ computational complexity and $1000\binom{n}{2} + n\log|\mathcal{X}| \approx 1000n^2 + n\log|\mathcal{X}|$ bits memory requirement

Can we further reduce the memory requirement? Yes!

Scheme 4 Hierarchical data structure

* Makes m arrays $A_1, \ldots, A_m$
* Each array A_j is of size m_j, $1 \leq j \leq m$, and operates the same way as the array scheme 3.
* We store x_i with $F(x_i) = j$ in array A_j
  ↳ Array A_j has its own hash $F_j : [N_j] \to [m_j]$



Suppose there are N_j elements assigned to A_j.
Then, if we use $m_j = 1000\binom{N_j}{2}$, the probability of having more than one element in any cell of $A_j \leq \frac{1}{1000}$.

→ We only want to ensure this for the array A_j where $j = F(y)$.

Thus, for $m_j = 1000\binom{N_j}{2}$, $1 \leq j \leq m$, the search complexity is $O(1)$.

The expected no. of bits stored $= n\log|\mathcal{X}| + \mathbb{E}\left[\sum_{j=1}^{m} 1000\binom{N_j}{2}\right]$

$$\mathbb{E}\left[\sum_{j=1}^{m}\binom{N_j}{2}\right] \leq \mathbb{E}\left[\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\mathbb{1}\{F(x_i)=j\}\right)^2\right]$$

$$= \mathbb{E}\left[\sum_{j=1}^{m}\sum_{i,i'=1}^{n}\mathbb{1}\{F(x_i)=j\}\,\mathbb{1}\{F(x_{i'})=j\}\right]$$

$$= \sum \mathbb{E}\left[\sum^{m}\mathbb{1}\{F(x_i)=F(x_{i'})=j\}\right]$$

$$= \sum_{i,i'} \mathbb{E}\left[\mathbb{1}\{F(x_i)=F(x_{i'})\}\right]$$

$$= \sum_{i,i'} \mathbb{P}\left(F(x_i)=F(x_{i'})\right)$$

$$= n + \sum_{i=1}^{n} \sum_{i' \neq i} \underbrace{\mathbb{P}\left(F(x_i)=F(x_{i'})\right)}_{\leq \frac{1}{m} \text{ if } x_1,\ldots,x_n \text{ are distinct}}$$

$$\leq n + \frac{n(n-1)}{m}$$

In particular, for $m = n$, the expected no. of bits stored $\leq \boxed{2n + n \log|\mathcal{X}|}$ $\Longrightarrow$ Scheme 4 attains memory complexity of scheme 1 and computational complexity of Scheme 2.

B Information theoretic lower bound

$x_1, x_2, \ldots, x_n \in \mathcal{X}$

Store them so that we can
- efficiently answer a search query (Is $y \in \{x_1, \ldots, x_n\}$)
- you less memory

storage $\approx O(n \log|\mathcal{X}|)$

computational efficiency = $O(1)$ per query

Our algorithm was randomized and worked with prob. $\geq$

Recall that our algorithm works for the worst-case input $(x_1, x_2, \ldots, x_n, y)$. "Therefore", it will also work for any random input for a distribution of our choice.

Suppose we fix a set $\{x_1, \ldots, x_n\}$ of distinct elements fro

Let $B_1, \ldots, B_n \sim$ iid $\text{Ber}(1/2)$ rvs, where x_i was stored.

Further, let y be given by x_J where $J \sim \text{unif}\{1, \ldots, n\}$ a

J is independent of $B_1, \ldots, B_n$.

We have $\boxed{\mathbb{P}\left(\hat{B}(M,J) = B_J\right) \geq 1-\delta}$

Note $\ell \geq H(M) \geq I(M \wedge B^n)$

$$= H(B^n) - H(B^n|M)$$

$$= n - \sum_{j=1}^{n} H(B_j|M, B^{j-1})$$

$$\geq n - \sum_{j=1}^{n} H(B_j|M)$$

$$= n\left(1 - \frac{1}{n}\sum_{j=1}^{n} H(B_j|M)\right)$$

(Since $J \perp\!\!\!\perp (M, B^n)$) $= n\left(1 - H(B_J|M,J)\right)$

By Fano's inequality,

$$H(B_J|M,J) \leq h\left(\mathbb{P}\left(\hat{B}(M,J) \neq B_J\right)\right)$$

$$\leq h(\delta)$$

$\Rightarrow \boxed{\ell \geq n(1-h(\delta))}$