K , L are parameters

→ Input: → $D = \{x_i\}_{i=1}^{N}$

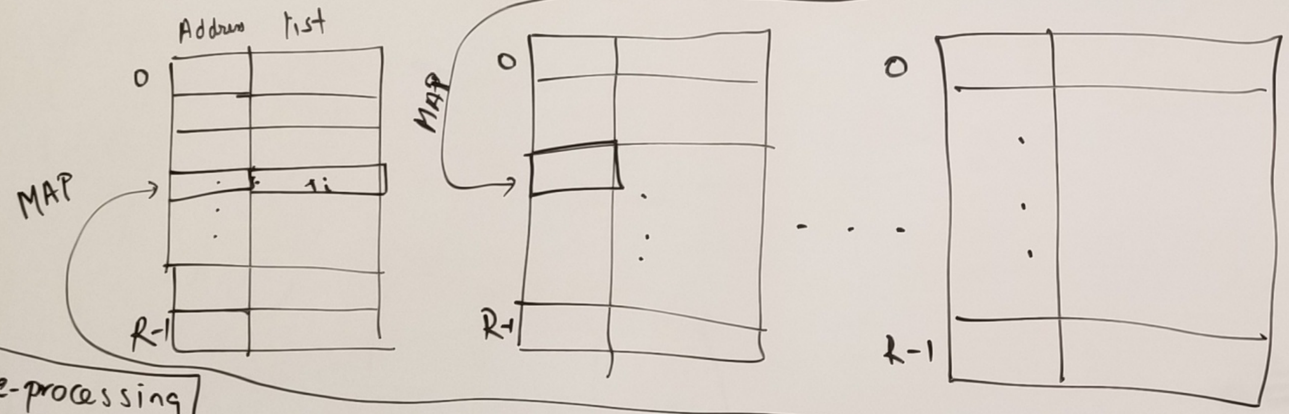
→ Preprocess D s.t.

- Given a query q find $x_i \in D$ s.t $sim(q, x_i)$ is high (very high)
- << $O(N)$ time.

→

- Initialize K X L independent hash functions. → LSH [minwise hashing]

- Create L hash tables. [0 - (R-1)]



MAP ($I_1, I_2 \ldots I_K$)

return

$((a_1 I_1 + a_2 I_2 + \ldots a_K I_K + a_{K+1} \mod P) \mod R$

a_is are chosen randomly and fixed

Pre-processing

- for each $x_i \in D$.

Compute K X L Hash of x_i

for j = 1 to L

MAP $[h_{j1}, h_{j2}, \ldots h_{jK}]$ in hash table j

↓ [0 - (R-1)]

HASH TABLE 1 | HASH TABLE 2

$h_{11}(x_i), h_{12}(x_i), \ldots h_{1K}(x_i)$ | $h_{21}(x_i), h_{22}(x_i), \ldots h_{2K}(x_i)$ ..... $h_{LK}(x_i)$

Query:→

Given → q, $K \times L$ hash functions, L Hash tables from pre-processing.

- Compute $K \times L$ LSH hashes of query q.

Returned set = ϕ

for $j = 1$ to L

Returned set = Returned set $\underset{\text{Union}}{\cup}$ $H_j[\text{MAP}(h_{j,1}(q), h_{j,2}(q), \ldots h_{j,K}(q))]$ (Hash table$_j$)

→ return Returned set. (candidates to search)

→ return top-k $x_b \in$ Returned set with highest $\text{sim}(q, x_b)$

→ Given q, the probability of $x_j \in$ Returned Set

$$\Rightarrow \quad 1 - \left(1 - \text{Collprob}(q, x_j)^K\right)^L$$

$$\text{Collprob}(q, x_j) = Pr\left(h(x_j) = h(q)\right) = \text{Jaccard Similarity if } h \text{ is minwise hashing.}$$

→ Monotonic in $\text{Collprob}(q, x_j)$

$\Rightarrow$ x_j is returned with higher probability than x_i iff.

$$Sim(x_j, q) > Sim(x_i, q)$$

→ Given q, the probability of $x_j \in$ Returned Set

$$\Rightarrow \quad 1 - \left(1 - \text{Collprob}(q, x_j)^K\right)^L \left[\text{Collprob}(q, x_j)^K + \left(1 - \text{Collprob}(q, x_j)^K\right] \times \frac{1}{R}\right]$$

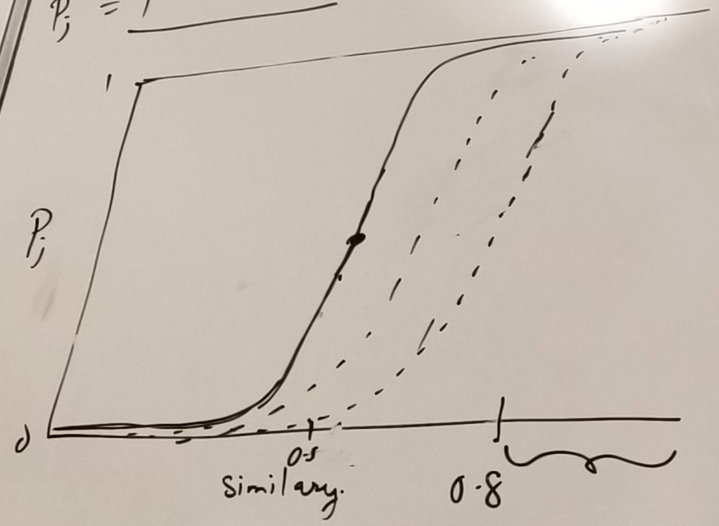
$$\text{Collprob}(q, x_j) = Pr\left(h(x_j) = h(q)\right) = \text{Jaccard Similarity if } h \text{ is minwise hashing.}$$

→ Monotonic in $\text{Collprob}(q, x_j)$

$\Rightarrow$ x_j is returned with higher probability ~~than~~ x_i iff.

$$Sim(x_j, q) > Sim(x_i, q)$$

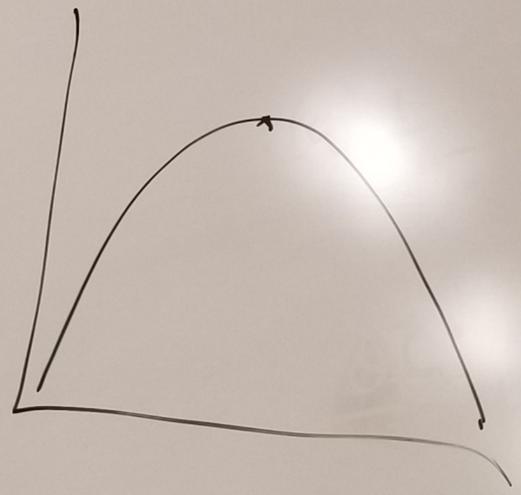
$P_j = 1 - (1 - c_j^K)^L$



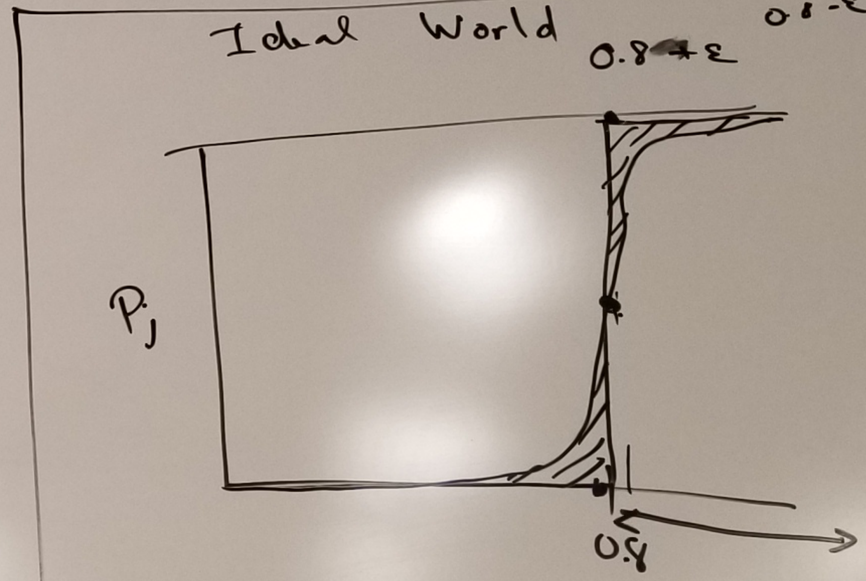
c_j	P_j
.2	.006
.3	0.047
0.5	0.18
0.6	0.8
0.7	.975
.8	.9996

$L = 20$

$K = 5$



Ideal World



- Initialize

- Reducing all pairwise comparisons.

  Near-Duplicate Detection, Entity Resolution

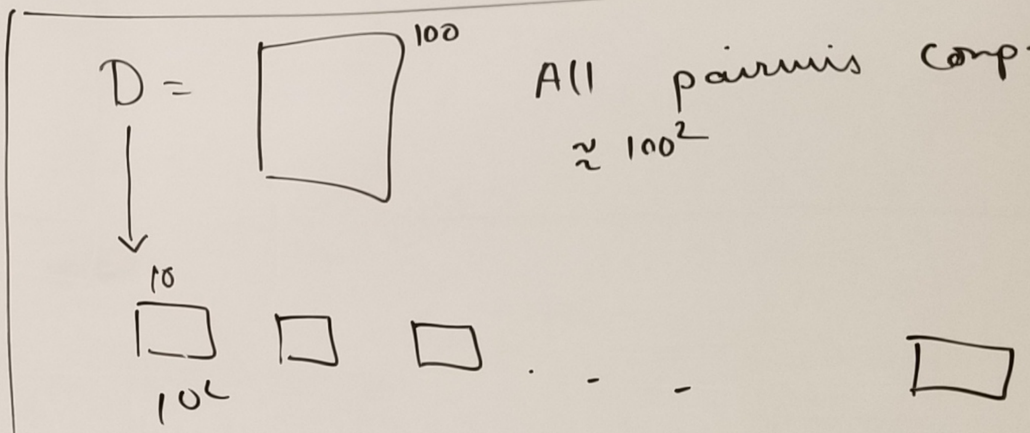
- Estimate Number of Unique Entries in a Given Record.

- $D_1 = \{x_i\}_{i=1}^{N}$; $D_2 = \{y_i\}_{i=1}^{N}$ . . . .

  ↓ text

- 350,000 (190,000)

- ≈ 63 billion



$\rightarrow \Pr(h(x) = h(y)) = f(\text{sim}(x, y))$

↑

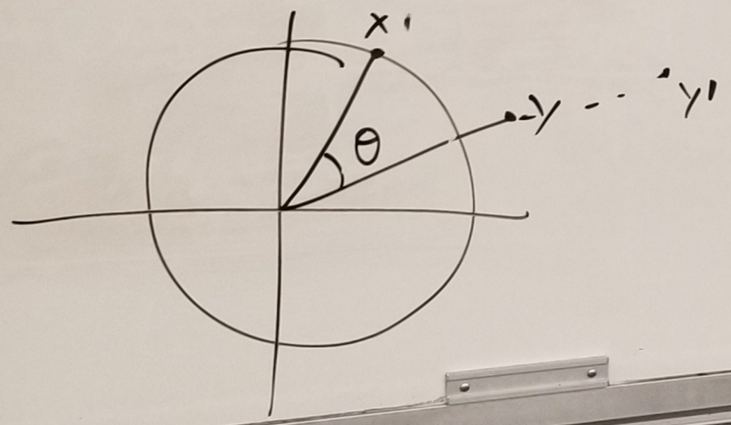
Jaccard $\rightarrow$ MinWise Hashing.

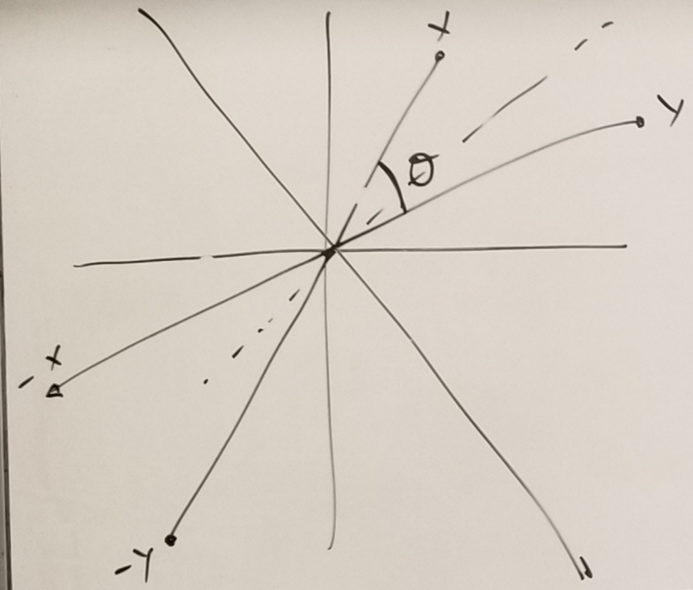
Cosine Similarity between $x, y \in \mathbb{R}^D$

$$C(x, y) = \frac{x^T y}{\|x\|_2 \|y\|_2} \in [-1, 1] = \theta$$

↓

vectors





Hash(x, seed)

Generate $r_i \sim N(0,1)$ using seed.

$r_1, r_2, \ldots r_D = r$

$x \in R^D$

Sign($x^T r$)

$$Pr(h(x) = h(y)) = 1 - \frac{\theta}{\pi} = 1 - \frac{\cos^{-1}\left(\frac{x^T y}{\|x\| \|y\|}\right)}{\pi}$$