Week 6

min Hash

represents each document as a small number of numbers from which we can easily estimate the Jaquard Distance

N documents/string (d average size)

↓ Preprocess them

Given a query document report if any document stored is close to it

m = # of queries

"Jaquard distance"

Break each document into fragments (size 7-8 typical)

$$d(x,y) = \frac{\text{fragments in } x \text{ and } y}{\text{fragments in } x \text{ or } y}$$

LSH "Locality-Sensitive hashing"

This allows efficient search for things that are similar

n documents (average size d)

m queries

$O(m \cdot n \cdot d)$

get rid of this I

get rid of this II

min Hash

Have n documents.

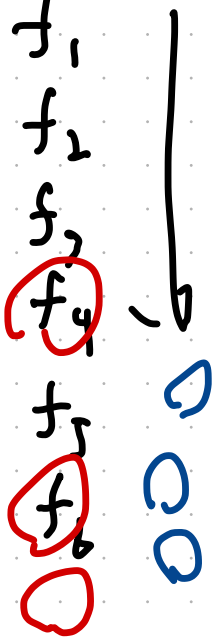
F = the set of all their fragments in random order

score(x) = The index of the first fragment in F that is a fragment of x

x: document

$Pr[score(x) = score(y)]$ = Jaquard distance between x and y

$E\begin{bmatrix} 1 & score(x) = score(y) \\ 0 & score(x) \neq score(y) \end{bmatrix} =$

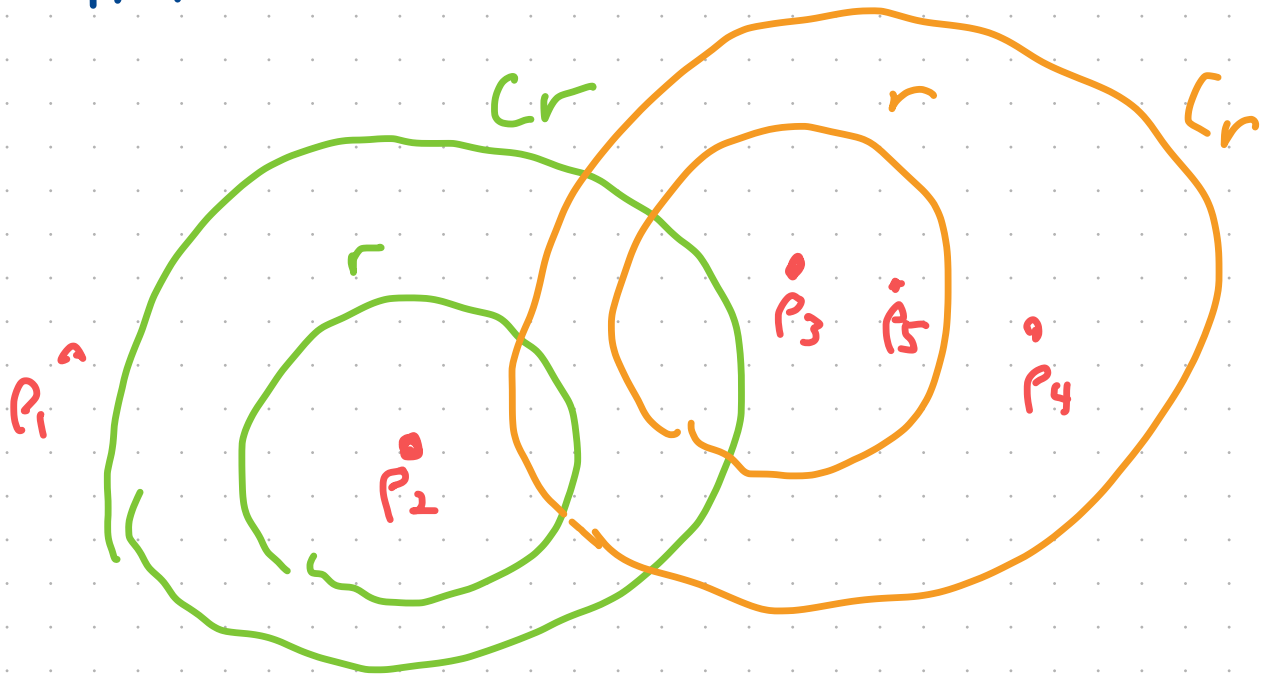


LSH: Locality Sensitive Hashing

- Suppose you have a set H of hash functions
- Same data
- Same distance function on the data

- If $d(x,y) \leq r$ then $h(x) = h(y)$ with prob $\geq p_1$
- If $d(x,y) \geq cr$ then $h(x) = h(y)$ with prob $\leq p_2$

$p_1 > p_2$

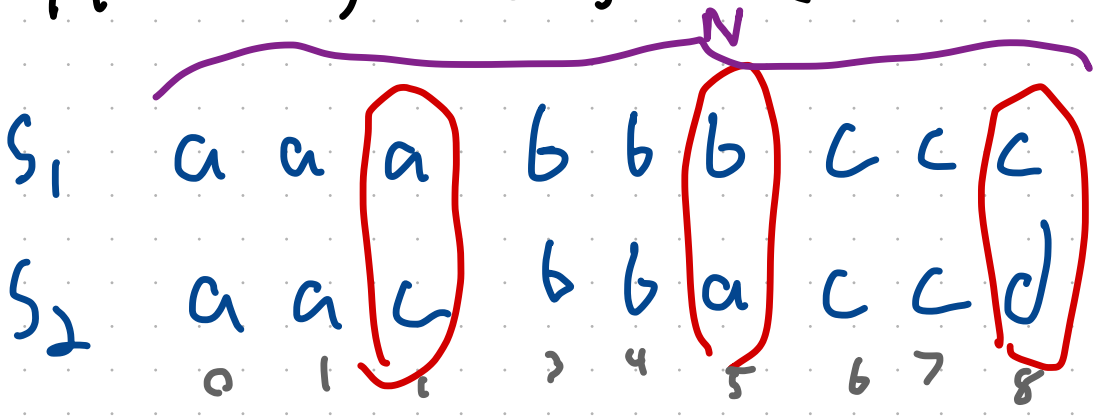


want $p_1 = 1$

$p_2 = 0$

Error = $\min(1-p_1, p_2)$

Hamming Distance



$d_{Hamming}(S_1, S_2) = 3$

Pick a random int i in the range $1..N$

Let $h(s) = s[i]$

$i = 6$

$h(S_1) = c$

$h(S_2) = c$

LSH: Locality Sensitive Hashing

- Suppose you have a set H of hash functions
- Same data
- Same distance function on the data

- If $d(x,y) \leq r$ then $h(x) = h(y)$ with prob $\geq p_1$
- If $d(x,y) \geq cr$ then $h(x) = h(y)$ with prob $\leq p_2$

$p_1 > p_2$

hamming

Choose

$r = 0.8n$ $\quad 0.8 = p_1$

$cr = 0.9n$ $\quad 0.9 = p_2$

0.8

0.8 p_1 = Chance that distance $\leq r$, correct

0.4 p_2 = Chance that distance $\geq cr$, incorrect answer

0.9

"AND" construction.

Hash k times, say distance small if all k are the same

$$p_1 \to p_1^k$$

$$p_2 \to p_2^k$$

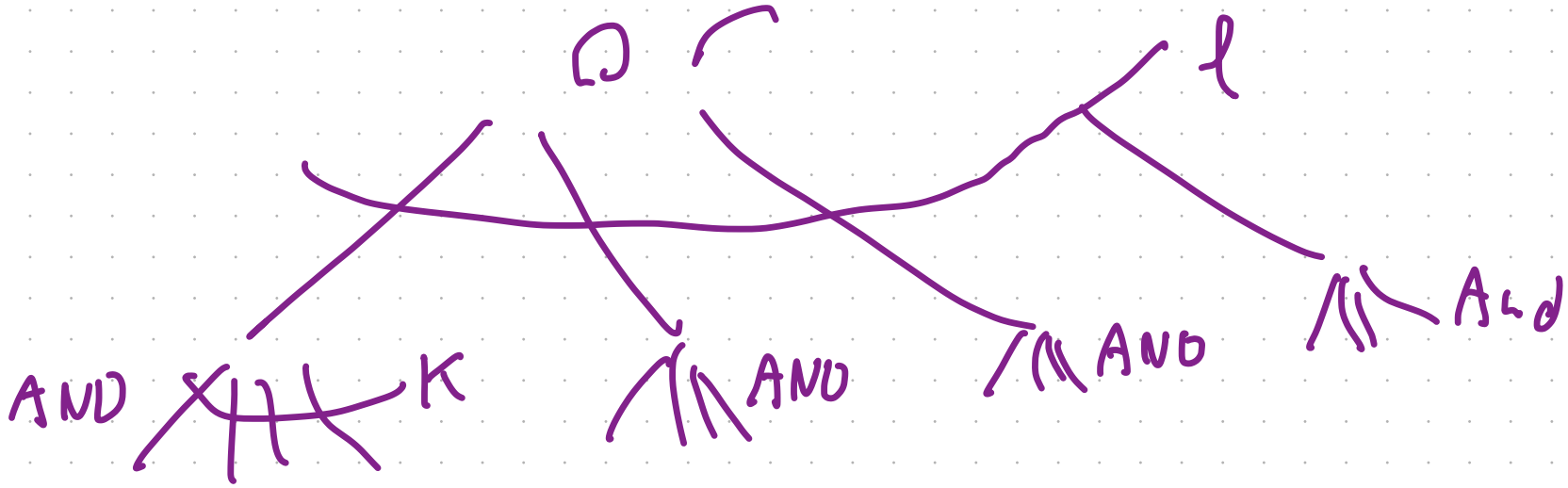
"OR" construction

Hash l times

Distance is small if at least 1 sens small

$$p_1 \to 1-(1-p_1)^k$$

$$p_2 \to 1-(1-p_2)^k$$



$$p_1 \rightarrow 1-(1-p_1^k)^l$$

$$p_2 \rightarrow 1-(1-p_2^k)^l$$

Data

↓

Fragments

↓

Multiple scores (lK)

↓

Group them into l groups of K

↓

for each group, make a dictionary that maps the scores of the group to the string that made it.

To search for S, compute scores, group, for each group look in the dictionary

$D[0] = \{ [1,5] \to S_1$
$[1,7] \to S_2$

$D[1] = \{ [2,3] \to S_1, S_2$

group 0 group 1

S_1 [[1, 5] [2, 3]]

S_2 [[1, 7] [2, 3]]