CSE 321 Discrete Structures

March 3rd, 2010

Lecture 22 (Supplement): LSH

Jaccard

- Jaccard similarity: J(S, T) = |S*T| / |S U T|

- Problem: given large collection of sets S1, S2, …, Sn, and given a threshold s, find all pairs Si, Sj s.t. J(Si, Sj) > s

Application: Collaborative Filtering

- We have n customers: 1, 2, …, n
- Each customer i buys a set of items Si
- We would like to recommend items bought by customer j, if J(Si, Sj) > s

Example

S1	S2	S3	S4	S5
Toothpaste floss	Floss mouthwash	Ipod PowerBook VideoAdapter	Ipod mouthwash	Floss toothpaste mouthwash

New customer buys mouthwash, floss;
What do you recommend ?

Application: Similar Documents

- Given n documents 1, 2, …, n
- Let Si be the set of q-grams for document I
- Want to find all pairs of “similar” documents, i.e. for which J(Si, Sj) > s

Example

- You work for a copyright violation detection company
- Customers: have documents 1, 2, 3, …., 10^6
- Web: has pages 1, 2, 3, …., 10^{11}

- Your job is to find "almost identical documents"
- What do you do ???

The Signature Method

- For each Si, compute a signature Sig(Si) s.t.

J(Si,Sj) > s ←→ Sig(Si) * Sig(Sj) ≠ emptyset

With high probability

The Signature Method

- Step 1: compute all pairs i,j for which Sig(Si)*Sig(Sj)≠emptyset
  - This is a join operation !
- Step 2: for all such pairs, return (i,j) if J(Si,Sj) > s
  - Hopefully only a few such pairs

Both false positives and false negatives are possible

The Signature Method

Will construct the signature in two steps:

1. Minhashes
2. LSH

Minhashing

- Let π be an arbitrary permutation of the domain

- For each i, let:

$$\boxed{mh(Si) = \{\text{the smallest element in } Si \text{ according to } \pi\}}$$

Example

- The entire domain is {a,b,c,d,e,f,g,h}
- The set Si is

$$\boxed{Si = \{a,b,c,e,f\}}$$

- Suppose we choose the permutation:

$$\boxed{\pi = d,g,c,h,b,f,a,e}$$

Then what is mh(Si) = ?

Minhashing

Main property:

Probability(mh(Si) = mh(Sj)) = J(Si,Sj)

Why ?

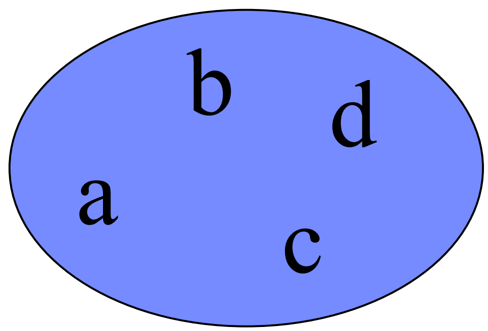
Warmup Question

- Choose a *random* permutation π of {a,b,c…,z}
- Consider the set S={a,b,c,d}
- What is the probability that mh(S) = c ?

Warmup Question

- Choose a *random* permutation π of {a,b,c…,z}

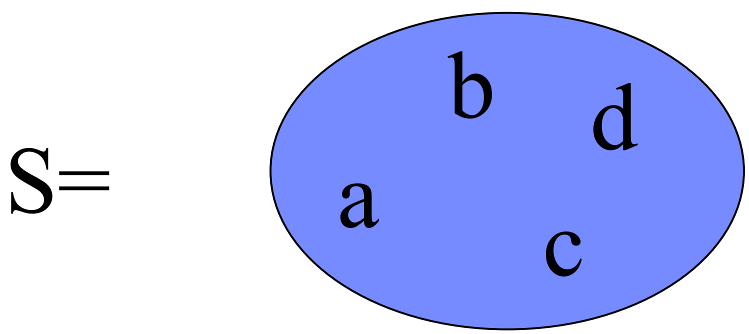
S=



What is the probability that mh(S) = c ?

Warmup Question

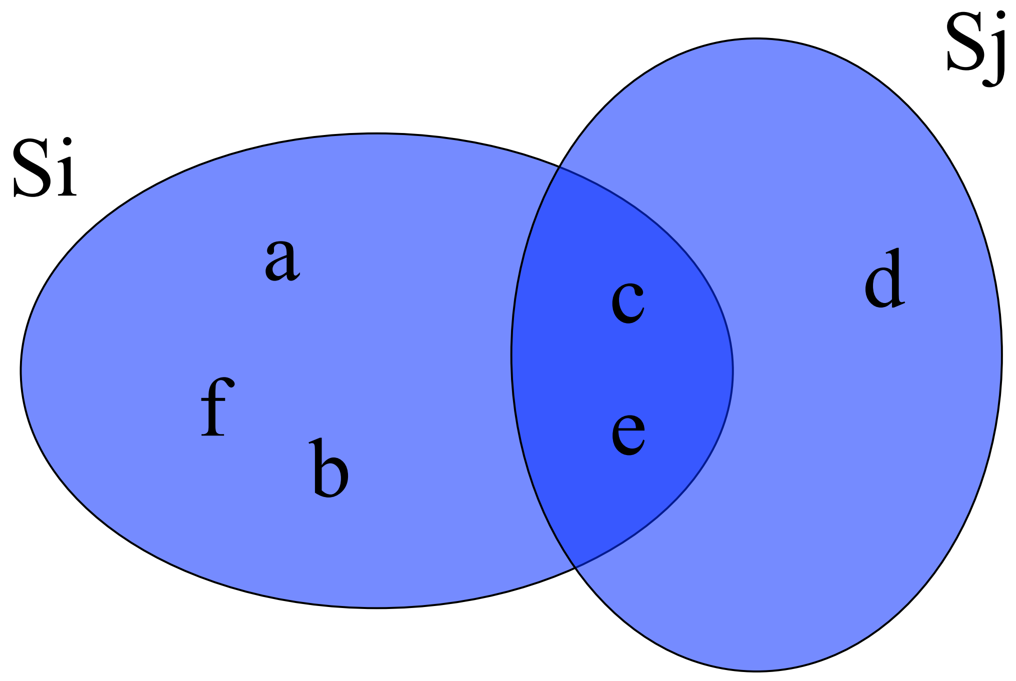
- Choose a *random* permutation π of {a,b,c…,z}



What is the probability that mh(S) = c ?

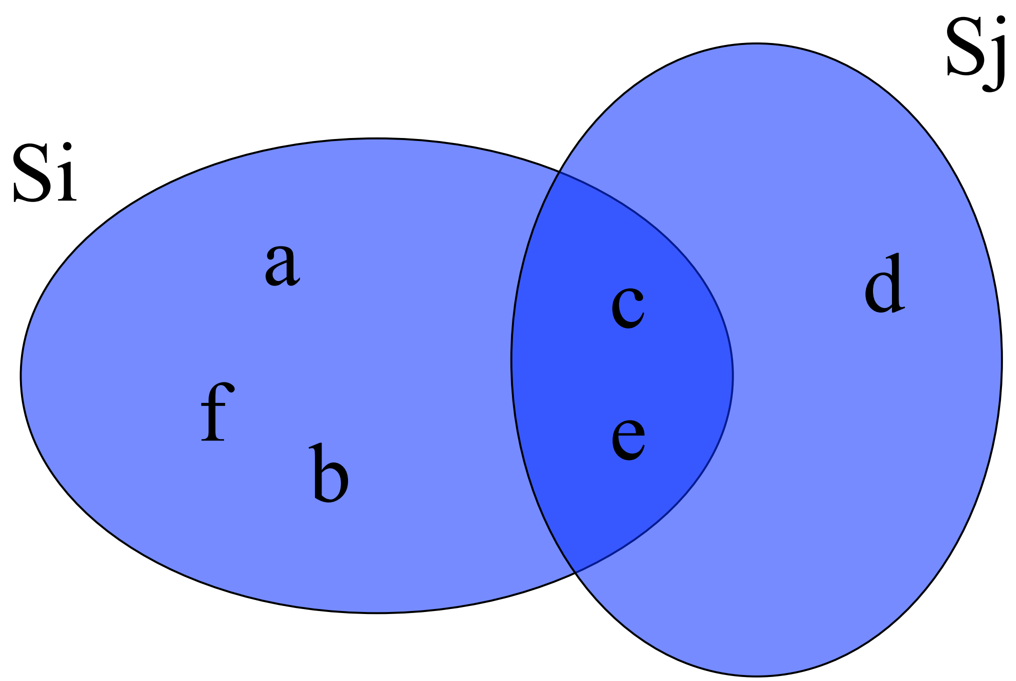
Answer: P = ¼ (each of a,b,c,d can be the min)

Main Property



What is Prob(mh(Si) = mh(Sj)) ?

Main Property



What is Prob(mh(Si) = mh(Sj)) ?

Answer: 2/6 = J(Si,Sj)

Computing Minhashes

- We use a hash function (which we assume is random)

```
mh(S) {
    v = ∞;
    forall x in S do
        if h(x) < v then {v = h(x); y = x;}
    return y;
}
```

Example

- The set Si is Si = {a,b,c,e,f}
- Compute h:
  h(a)=77,h(b)=55,h(c)=33,h(e)=88,h(f)=66
- Then what is mh(Si) = ?

Usage Idea

- Recall: we have n sets S1, …, Sn

- Compute mh(S1), …, mh(Sn)
- Consider only those pairs for which mh(Si)=mh(Sj): compute their Jaccard similarity
- But too many false negatives !
- How can we improve ?

Improvement

- Independent hash functions h1, …, hm

- For each Si, compute MH(Si) = the m minhashes for each j=1,…,m

Example

- The set Si is Si = {a,b,c,e,f}
- Compute h1:
  h1(a)=77,h1(b)=55,h1(c)=33,h1(e)=88,h1(f)=66
- Compute h2:
  h2(a)=22,h2(b)=66,h2(c)=55,h2(e)=11,h2(f)=44
- Then what is MH(Si) = ?

Example

- The set Si is Si = {a,b,c,e,f}
- Compute h1:
  h1(a)=77,h1(b)=55,h1(c)=33,h1(e)=88,h1(f)=66
- Compute h2:
  h2(a)=22,h2(b)=66,h2(c)=55,h2(e)=11,h2(f)=44
- Then what is MH(Si) = ?
- Answer: MH(Si) = (c, e)   (an ordered pair)

Using Minhashes

- Compute MH(S1), …, MH(Sn)
- J(Si, Sj) ≈ the fraction of positions where MH(Si) and MH(Sj) agree

Example

n=7

S1	S2	S3	S4	S5	S6	S7

m=6

1	1	4	3	5	4	4
4	6	4	6	9	4	4
8	7	8	9	5	6	8
4	8	7	8	4	3	7
8	7	8	9	5	6	8
6	8	6	8	4	5	6

Estimate J(S1,S2) = ?, then J(S1,S3)=?

Note: Minhashes still require Jaccard

- How do we compute the fraction of positions where MHi and MHj agree ?
- Annotate each position with its position number
- Then this is precisely J(MHi, MHj)

Note:
Minhashes still require Jaccard

1	1		4
2	4		4
3	8		8
..	4		7
..	8		8
m	6		6

MH1={(1,1),(2,4),(3,8),(4,4),(5,8),(6,6)}
MH2={(1,4),(2,4),(3,8),(4,7),(5,8),(6,6)}

J(MH1, MH2) = 4/ (2m-4)

Fraction of equal positions = 4/m

Comments on Minhashes

- It is not a signature yet !
- We have only reduced the problem of computing J(Si,Sj) to the problem of computing J on smaller sets, of size m
- The signature is provided by LSH (next)

Locality Sensitive Hashing

- We have n strings MH1, …, MHn, each of length m
- Compute a signature Sig(MHi) for each string Mhi

- Desired property:

$$\boxed{\mathrm{Sig}(MHi)*\mathrm{Sig}(MHj) \neq \text{empty} \Leftrightarrow J(MHi,MHj) > s}$$

LSH

- Each MHi consists of m values
- Each signature Sig(MHi) will have size b
- Divide the m values into b “bands” of size r “rows”, i.e. m = b*r
- For each band j=1,…,r, apply a hash function hj to the string of values in band j in MHi ➔ hj
- Then Sig(MHi) = (h1, h2, …, hb)

Example

$n=7$

S1	S2	S3	S4	S5	S6	S7
1	1	4	3	5	4	4
4	6	4	6	9	4	4
8	7	8	9	5	6	8
4	8	7	8	4	3	7
8	7	8	9	5	6	8
6	8	6	8	4	5	6

$m=6$
$b=3$
$r=2$

r rows

Each rectangle (=band) becomes one new hash value

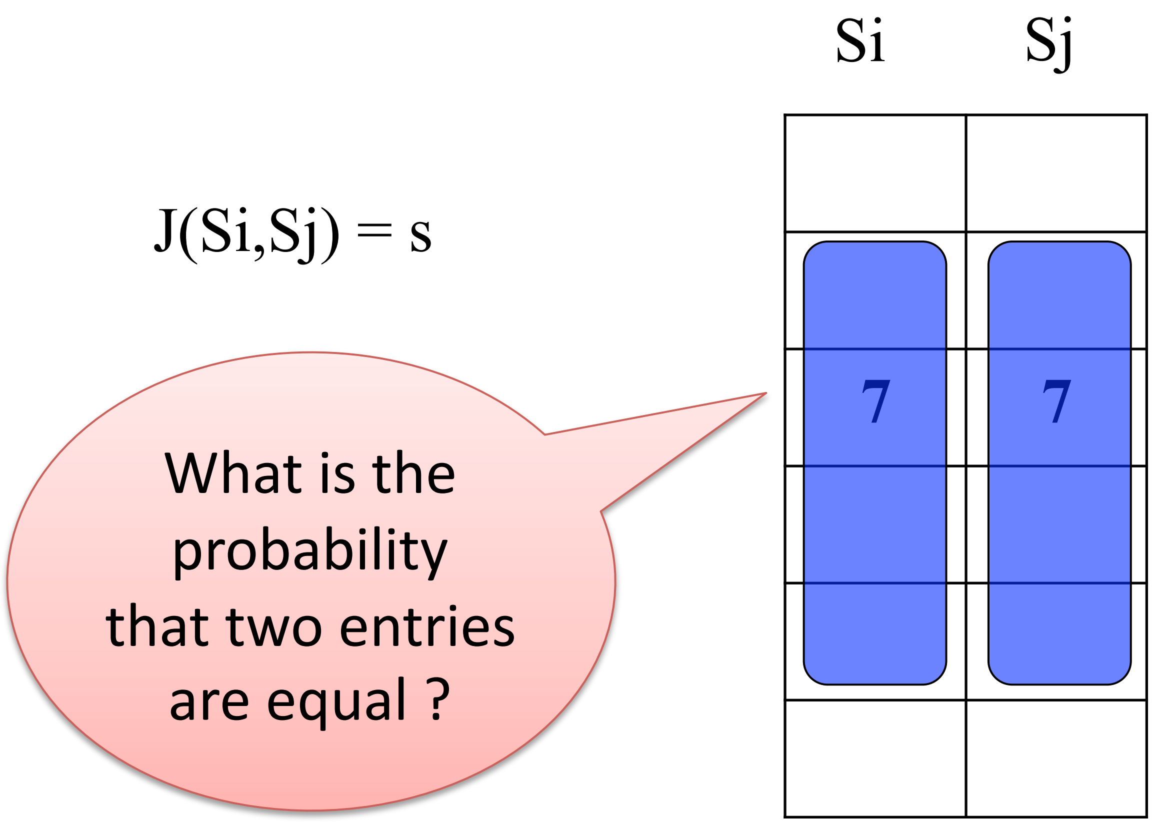
Analysis

- Goal: want to compute the probability that Sig(MH(Si)) * Sig(MH(Sj)) ≠ emptyset, as a function of s = J(Si, Sj)

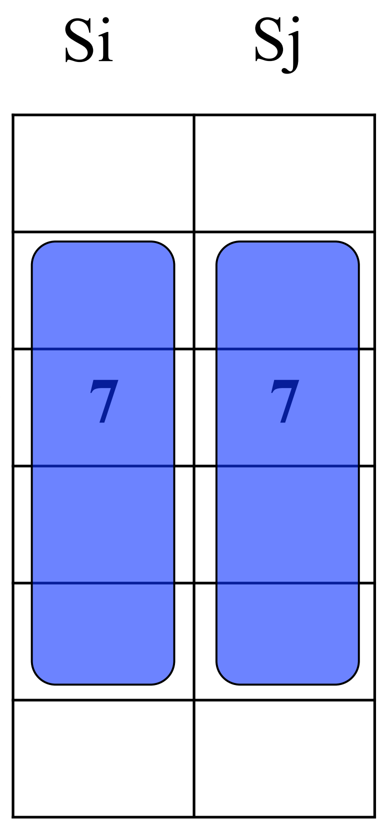
- So let s = J(Si, Sj) = fraction of equal positions

Analysis

$J(Si,Sj) = s$



Analysis



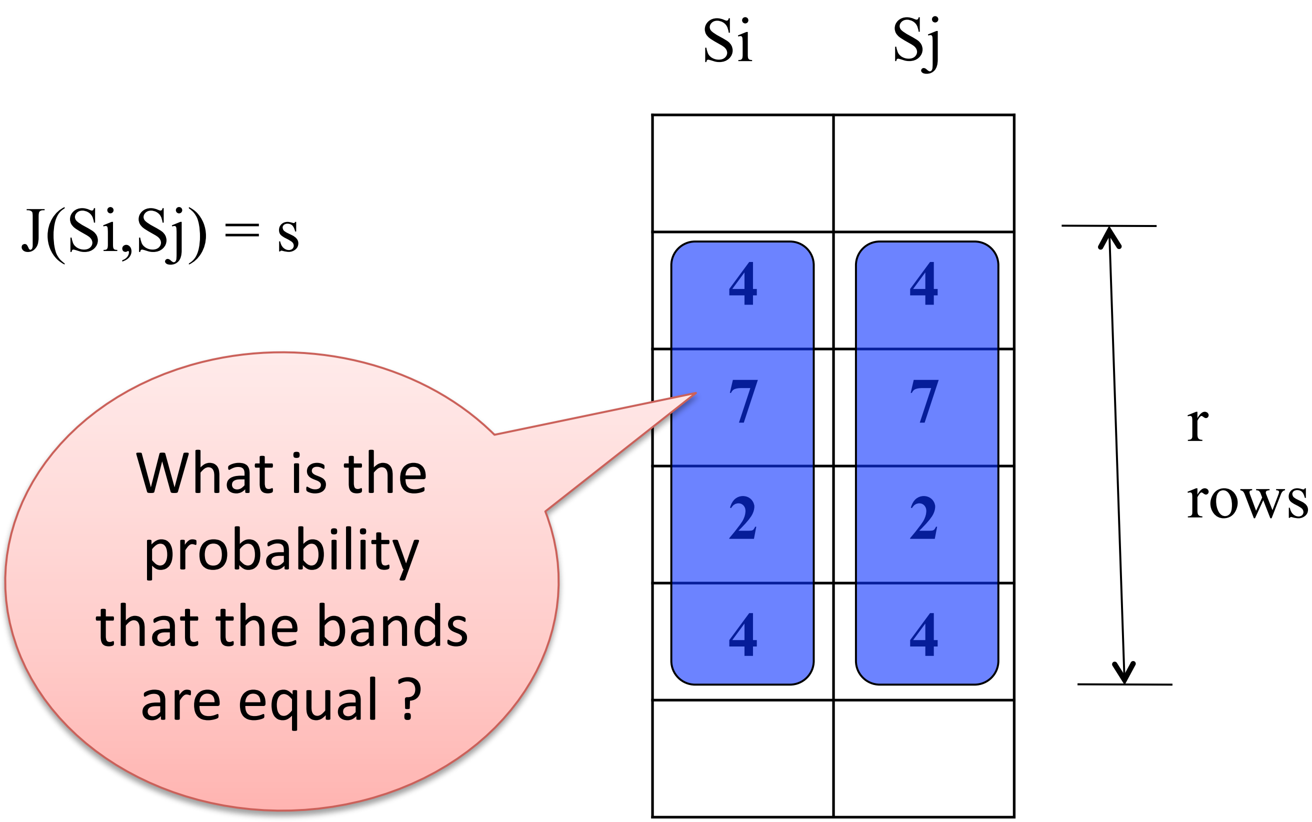
J(Si,Sj) = s

What is the probability that two entries are equal ?

Answer: s

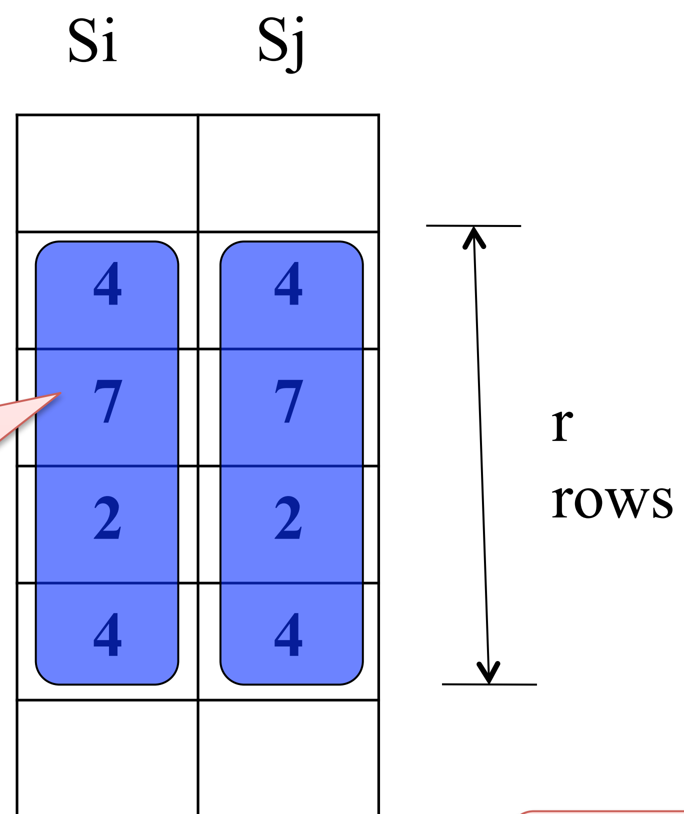
Analysis

$J(Si,Sj) = s$



Analysis

$J(Si,Sj) = s$

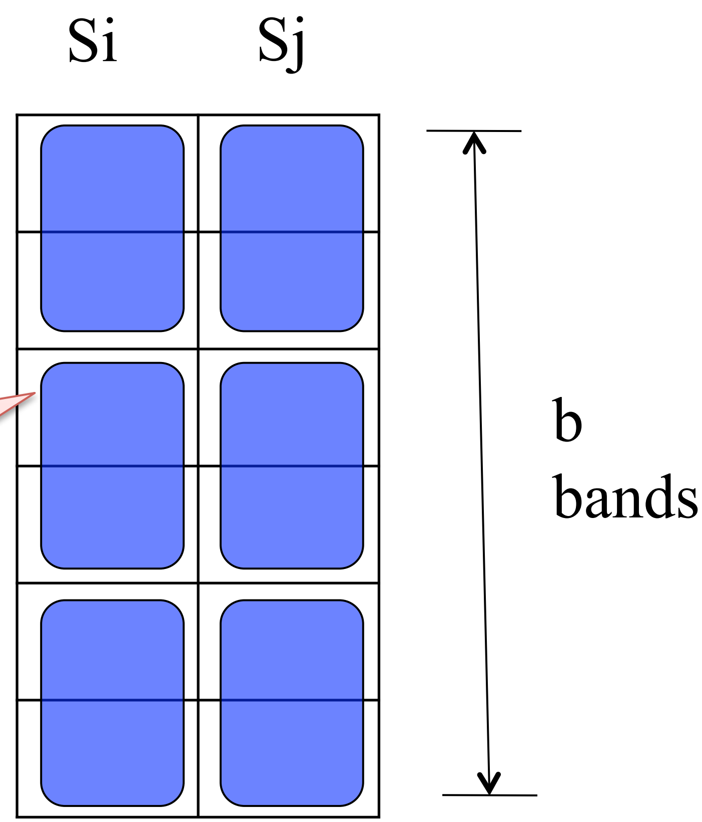


What is the probability that the bands are equal ?

Answer: s^r

Analysis

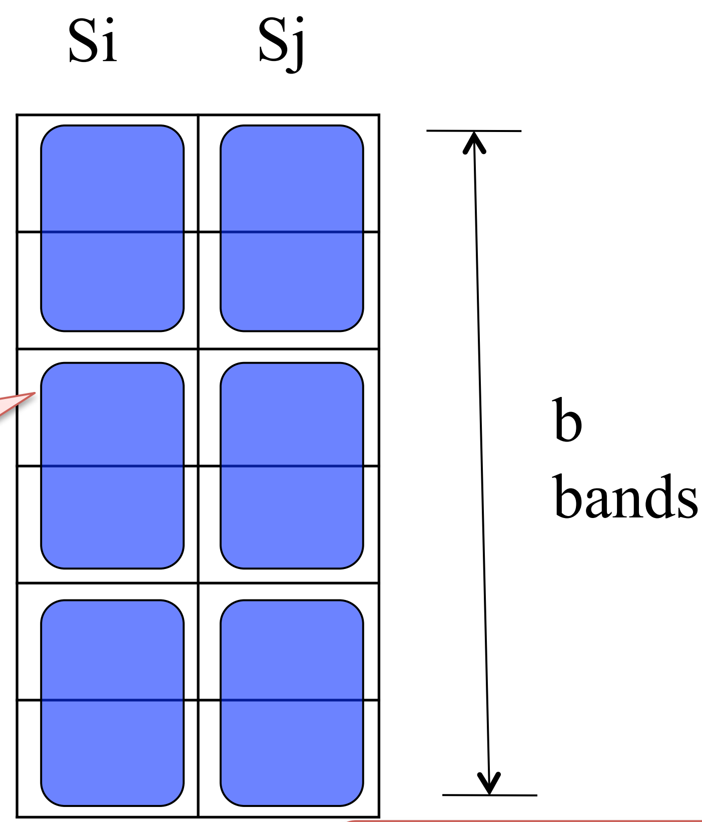
$J(Si,Sj) = s$



What is the probability that some pair of bands are equal ?

Analysis

$J(Si,Sj) = s$

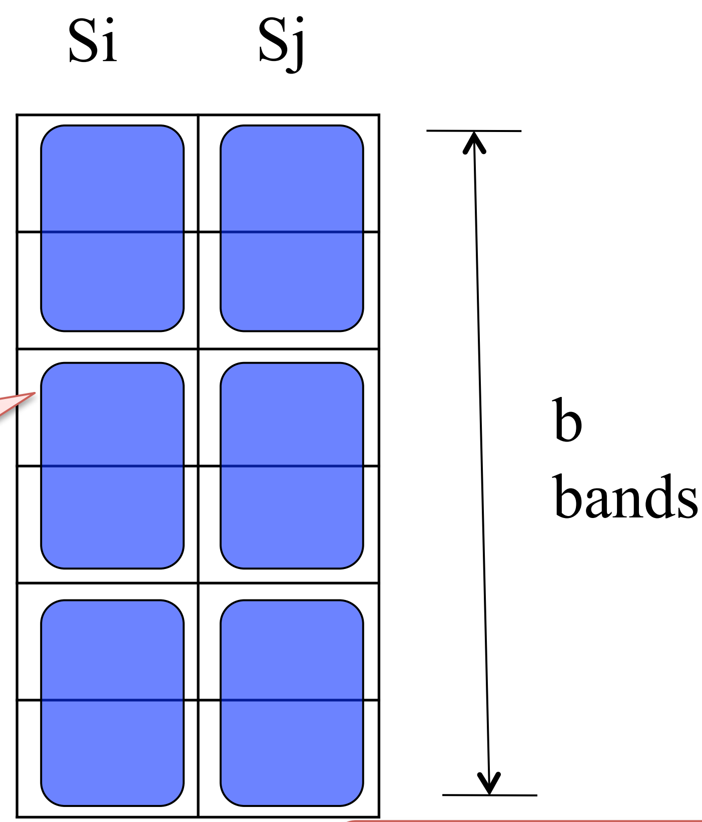


What is the probability that some pair of bands are equal ?

Answer: $1-(1-s^r)^b$

This is precisely
the probability that
Sig(Si) * Sig(Sj)≠ emptyset

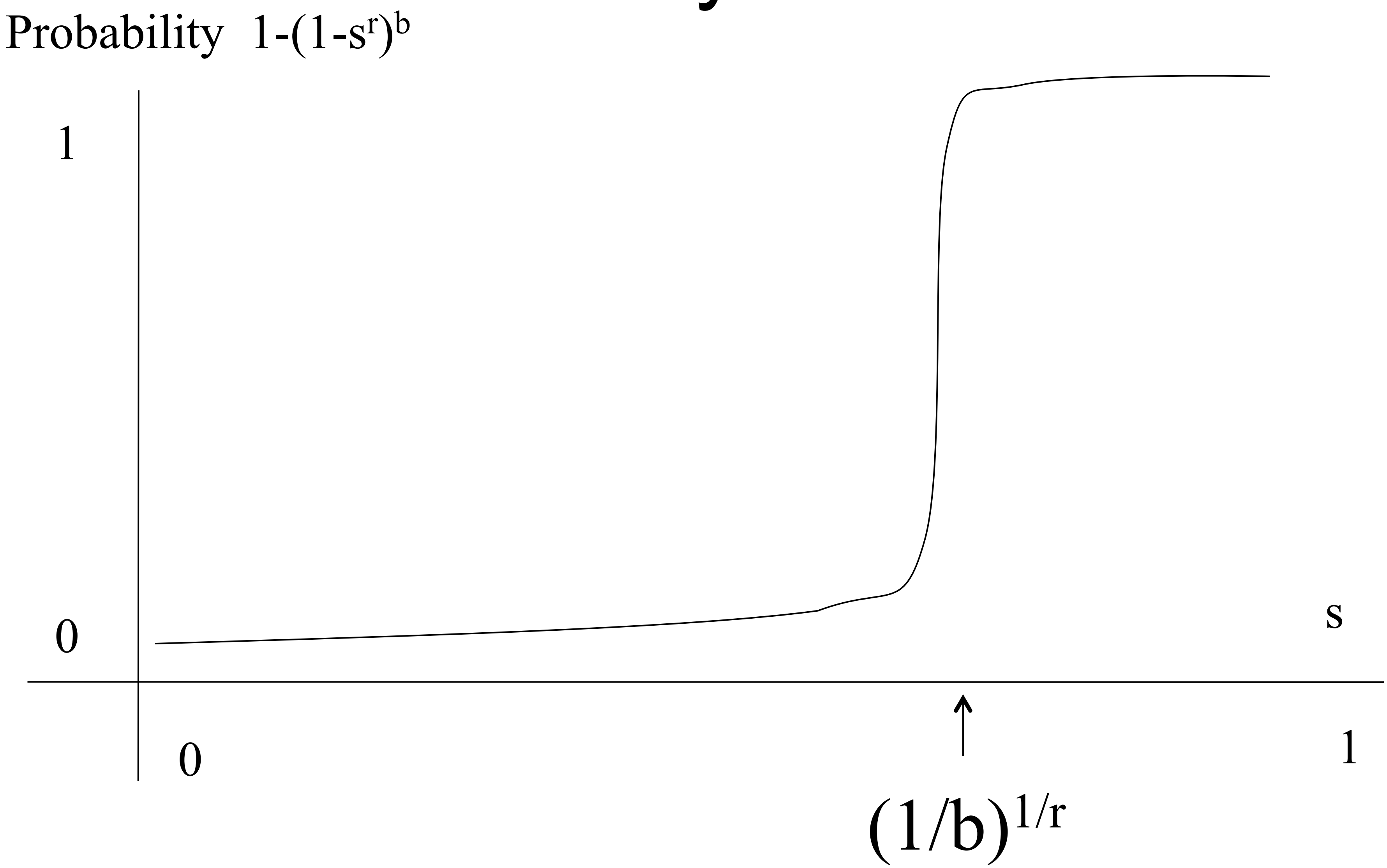
$J(Si,Sj) = s$



What is the probability that some pair of bands are equal ?

Answer: $1-(1-s^r)^b$

Analysis



Putting it together

- You work for a copyright violation detection company
- Customers: have documents 1, 2, 3, …., 10^6
- Web: has pages 1, 2, 3, …., 10^{11}

- Your job is to find "almost identical documents"
- What do you do ???

Step 1: Q-grams

Doc

DocID	Qgram
1	To
1	o b
1	be
1	be
1	e o
1	or
1	or
1	r n
1	no
. . .	. . .
. . .	
2	. .
. . .	

Web

URL	Qgram
abc.com	Oba
abc.com	Bam
abc.com	ama
abc.com	ma
abc.com	a h
. . .	. . .
bcd.com	. . .
. . .	

Step 2: Compute m Min-hashes

docID	mh1	mh2	mh3	mh4	…	mh500
1	To	que	Ion	or		Not
2	. . .	. . .				

url	mh1	mh2	mh3	mh4	…	mh500
abc.com	sec	ret	def	ens		…
bcd.com	. . .	. . .				

Note: m = a few hundreds

Step 3: Compute Signatures

docID	h(mh1,…,mh20)	h(mh21,…,mh40)	. . .
1	2345234	3232	. . .
2	. . .		

docSIG

docID	Sig
1	2345234@1
1	3232@2
1	…
1	452342@25
2	23423@1
2	. . .

webSIG

url	Sig
abc.com	676876@1
abc.com	3232@2
abc.com	…
abc.com	787892@25
bcd.com	23423@1
…	. . .

Step 4: Find docs with common signatures

Need two indexes:

for every signature s, doc[s] = set of documents containing s

for every webpage w, web[] = set of webpages containing s

```
For all s in Sig do
   for d in doc[s], for w in web[s]  do
        print(d, w)
```