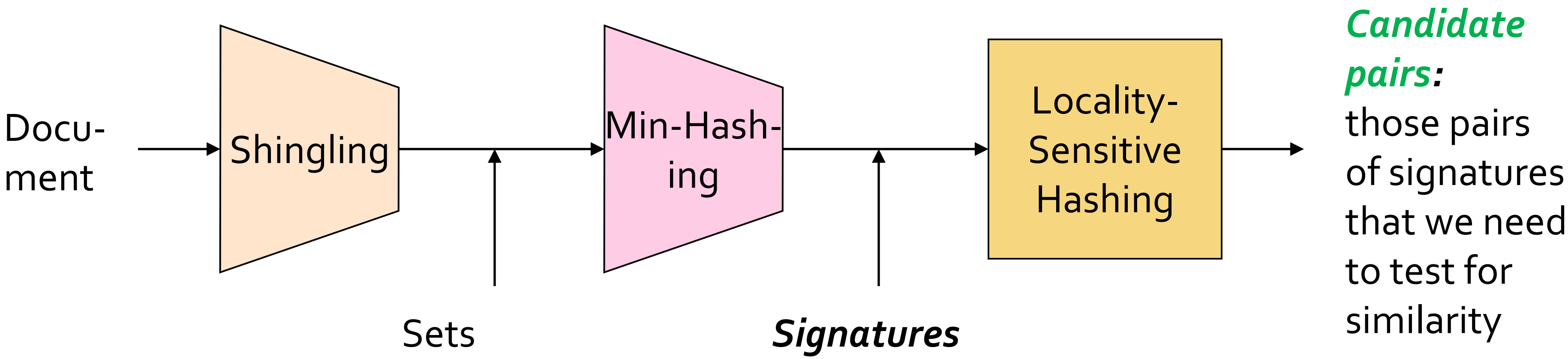
#20 – Finding Similar Items (3)

Determine candidate pairs



Locality Sensitive Hashing

Locality-Sensitive Hashing:

LSH: General Idea

- **Goal:** Find documents with Jaccard similarity at least ***s***
  - for some similarity threshold, e.g., ***s***=0.8

- **LSH – General idea:**
  - Use a function ***f(x,y)*** that tells whether ***x*** and ***y*** is a ***candidate pair***:
    - a pair of elements whose similarity must be evaluated

LSH: General Idea

- **For Min-Hash matrices:**
  - Hash columns of signature matrix ***M*** to many buckets
    - Remember that columns represent documents
  - Each pair of documents that hashes into the same bucket is a **candidate pair**

Candidates from Min-Hash

- **Pick a similarity threshold *s* (0 < s < 1)**
- Columns ***x*** and ***y*** of ***M*** are a **candidate pair** if their signatures agree on at least fraction ***s*** of their rows:
  M **(*i*, *x*)** **=** ***M*** **(*i*, *y*)** for at least frac. ***s*** values of ***i***
- We expect documents ***x*** and ***y*** to have the same (Jaccard) similarity as their signatures

LSH for Min-Hash

- **Big idea: <u>Hash</u> columns of signature matrix *M* <u>several times</u>**
- Arrange that (only) **similar columns** are likely to **hash to the same bucket**, with high probability
- **Candidate pairs are those that hash to the same bucket**

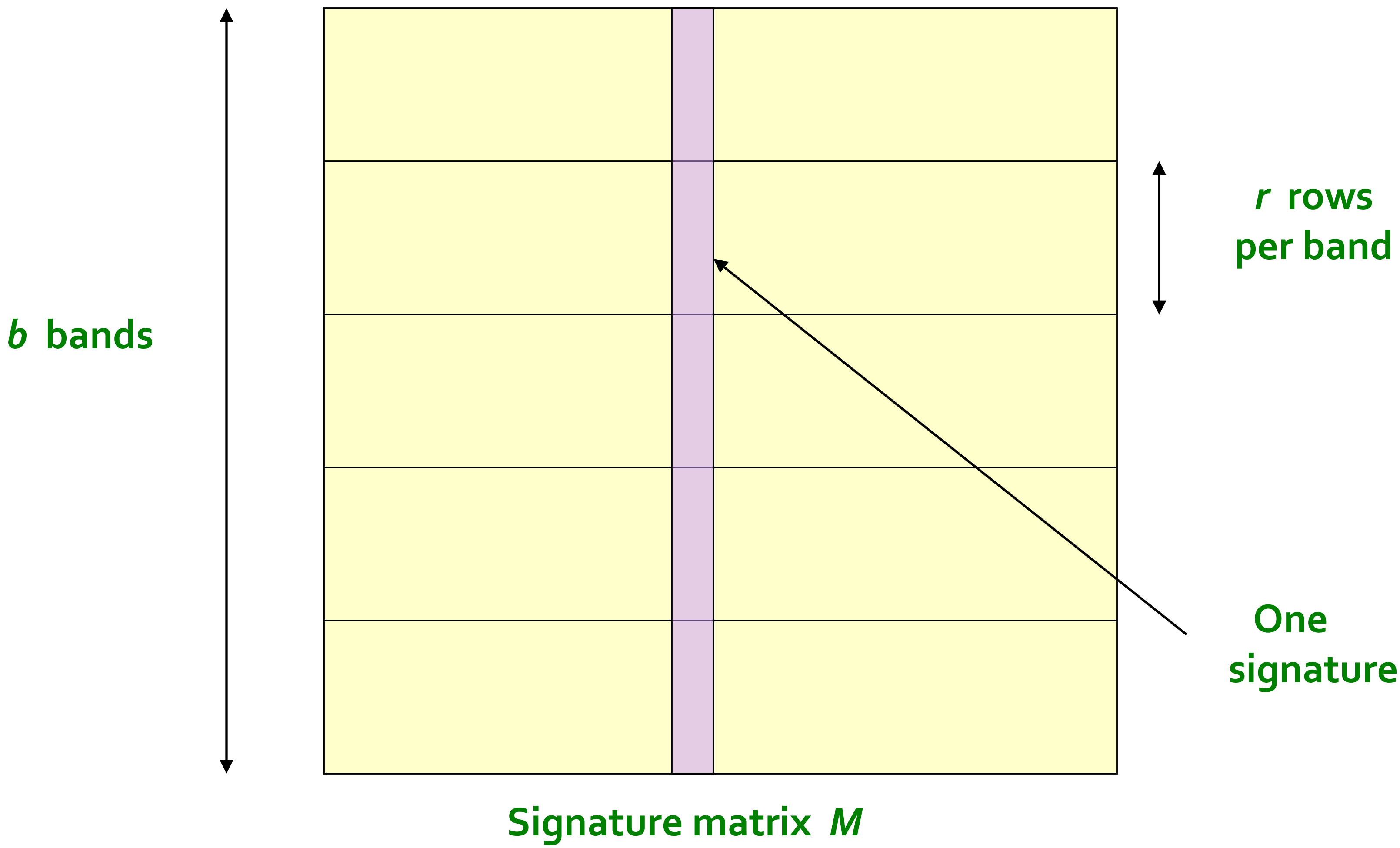
Hash several times …

- Divide matrix ***M*** into ***b*** bands of ***r*** rows

- For each band, hash its portion of each column to a hash table with ***k*** buckets

- Make ***k*** as large as possible

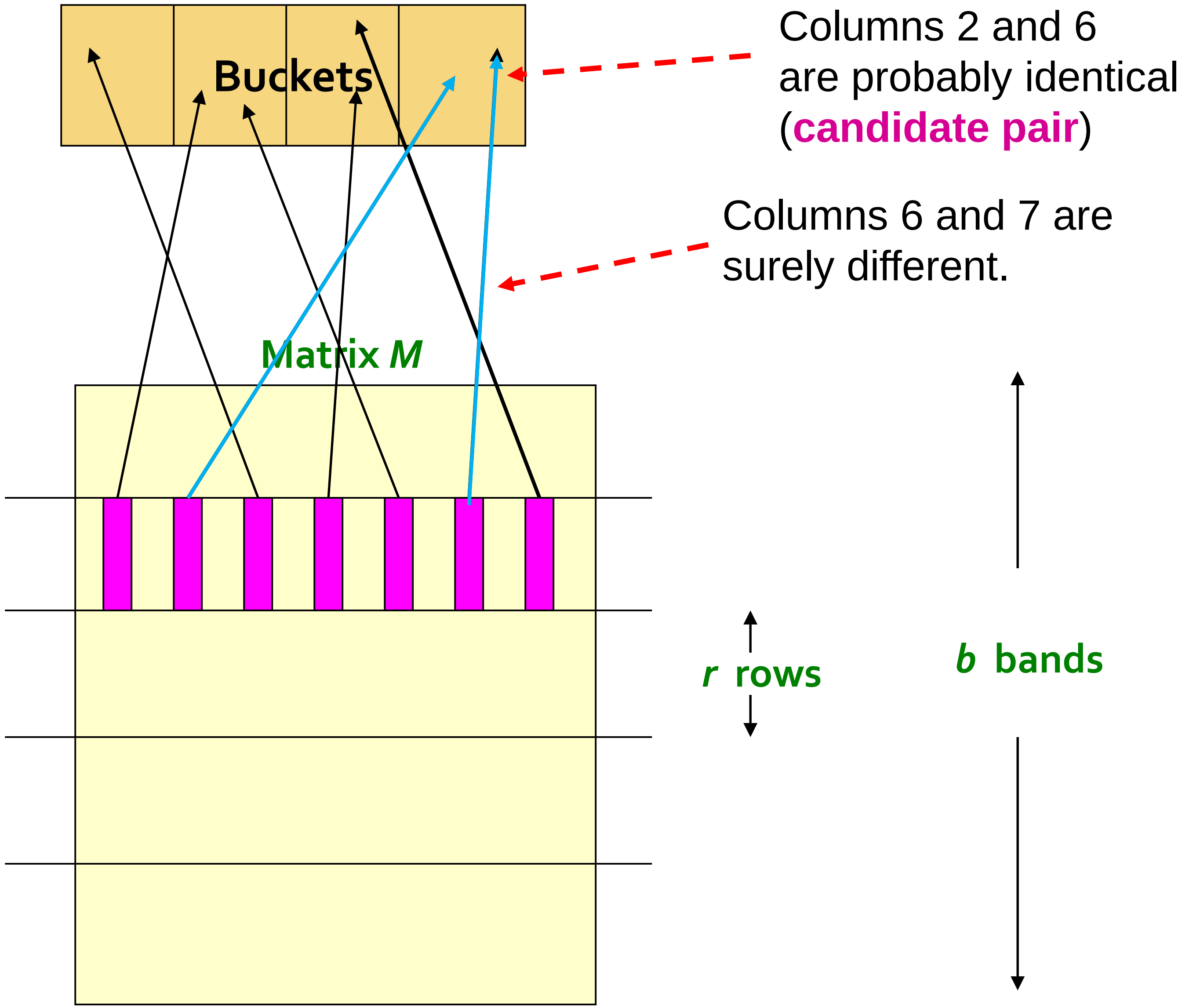
Partition *M* into *b* Bands



Partition M into Bands

- ***Candidate*** **column pairs are those that hash to the same bucket for $\geq$ 1 band**
- Tune ***b*** and ***r***
  - to catch most similar pairs
  - but few non-similar pairs

Hashing Bands



Analysis of the Banding Technique

Simplifying Assumption

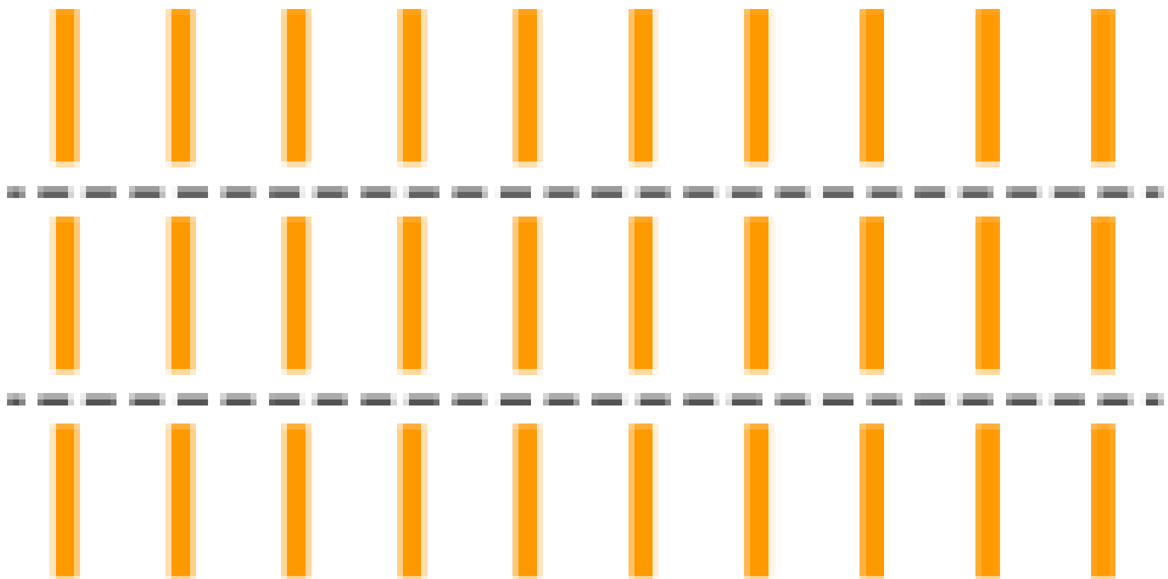
- IMPORTANT:
  - There are **enough buckets** that columns are unlikely to hash to the same bucket unless they are **identical** in a particular band

- Hereafter, we assume that "**same bucket**" means "**identical in that band**"

- Assumption needed only to simplify analysis, not for correctness of algorithm

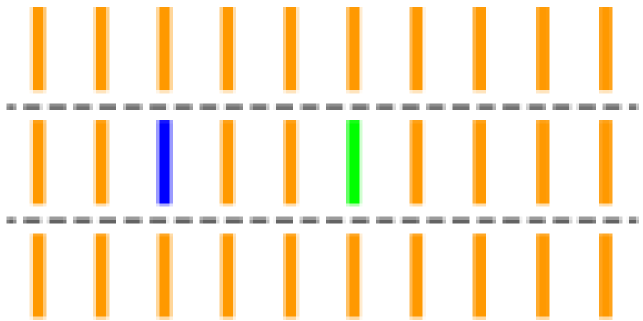
b bands, *r* rows/band

- It is convenient to represent the bands in an abbreviated way
- Next example refers to 10 docs and 3 bands



b bands, *r* rows/band

- Consider the two blocks marked (with blue and green) in the figure:



- The probability of all elements of the blue block being equal to the corresponding elements of the green block is $\mathbf{J^r}$
  - J is the Jaccard index/similarity of the two objects
    - Columns C_1 and C_2 have similarity $\mathbf{J}$
  - r is the number of rows in each band

Probabilities

- Probability that not all elements in blue block and green block are equal:
  - $\mathbf{1 - J^{r}}$
- Probability that not all elements are equal for the several bands is:
  - $\mathbf{(1 - J^{r})^{b}}$
    - Where b is the number of bands
- Probability of at least 1 band identical

$$\mathrm{P} = \mathbf{1 - (1 - J^{r})^{b}}$$

Example of application

Assume the following case:

- Suppose 100,000 columns of ***M*** (100k docs)
- Signatures of 100 integers (rows)
- Therefore, signatures take 40Mb

- Choose ***b*** = 20 bands of ***r*** = 5 integers/band

- **Goal:** Find pairs of documents that are at least ***s*** *= 0.8* similar

Case 1: C_1, C_2 are 80% Similar

- **Find pairs of ≥ *s***=0.8 similarity, set **b**=20, **r**=5
- **Assume:** sim(C_1, C_2) = 0.8
  - Since sim(C_1, C_2) ≥ **s**, we want C_1, C_2 to be a **candidate pair**:
    - We want them to **hash to at least 1 common bucket** (at least one band identical)

Case 1 : 80 % similar

- **Probability C_1, C_2 identical in one particular band:**
  - J^r = $(0.8)^5$ = 0.328

- Probability C_1, C_2 are ***not*** similar in all of the 20 bands: $(1\text{-}0.328)^{20}$ = 0.00035
  - i.e., about 1/3000th of the 80%-similar column pairs are **false negatives**

- **We would find 99.965% pairs of truly similar documents**

Case 2: C_1, C_2 are 30% Similar

- **Find pairs of ≥ *s***=0.8 similarity, set **b**=20, **r**=5
- **But now Assume:** sim(C_1, C_2) = 0.3
- Since sim(C_1, C_2) < **s** we want C_1, C_2 to hash to **NO common buckets**
  - all bands should be different

Case 2 (cont.): 30 % similar

- **Probability C_1, C_2 identical in one particular band:**
  - $(0.3)^5 = 0.00243$, as before
- Probability C_1, C_2 **identical in at least 1 of 20 bands**:
  - $1 - (1 - 0.00243)^{20} = 0.0474$
  - Approximately 4.74% pairs of docs with similarity 0.3% end up becoming **candidate pairs**
  - They are **false positives** since we will have to examine them (they are candidate pairs)
    - but then it will turn out their similarity is below threshold **s**

LSH Involves a Tradeoff

- **Pick:**
  - The number of Min-Hashes (rows of ***M***)
  - The number of bands ***b***, and
  - The number of rows ***r*** per band

to balance false positives and false negatives

Example (with less bands)

- Only 15 bands of 5 rows
- What happens to false positives ?
- And to false negatives ?

15 bands of 5 rows – False Positives

- **Probability C_1, C_2 identical in one particular band:** $(0.3)^5 = 0.00243$
- Probability C_1, C_2 identical in at least 1 of 15 bands: $1 - (1 - 0.00243)^{15} = 0.0358$
  - In other words, approximately 3.6% pairs of docs with similarity 0.3% end up becoming **candidate pairs**
    - They are **false positives**
- **False positives decreased**
  - **It was 4.74 % for b=20**

15 bands of 5 rows – false negatives

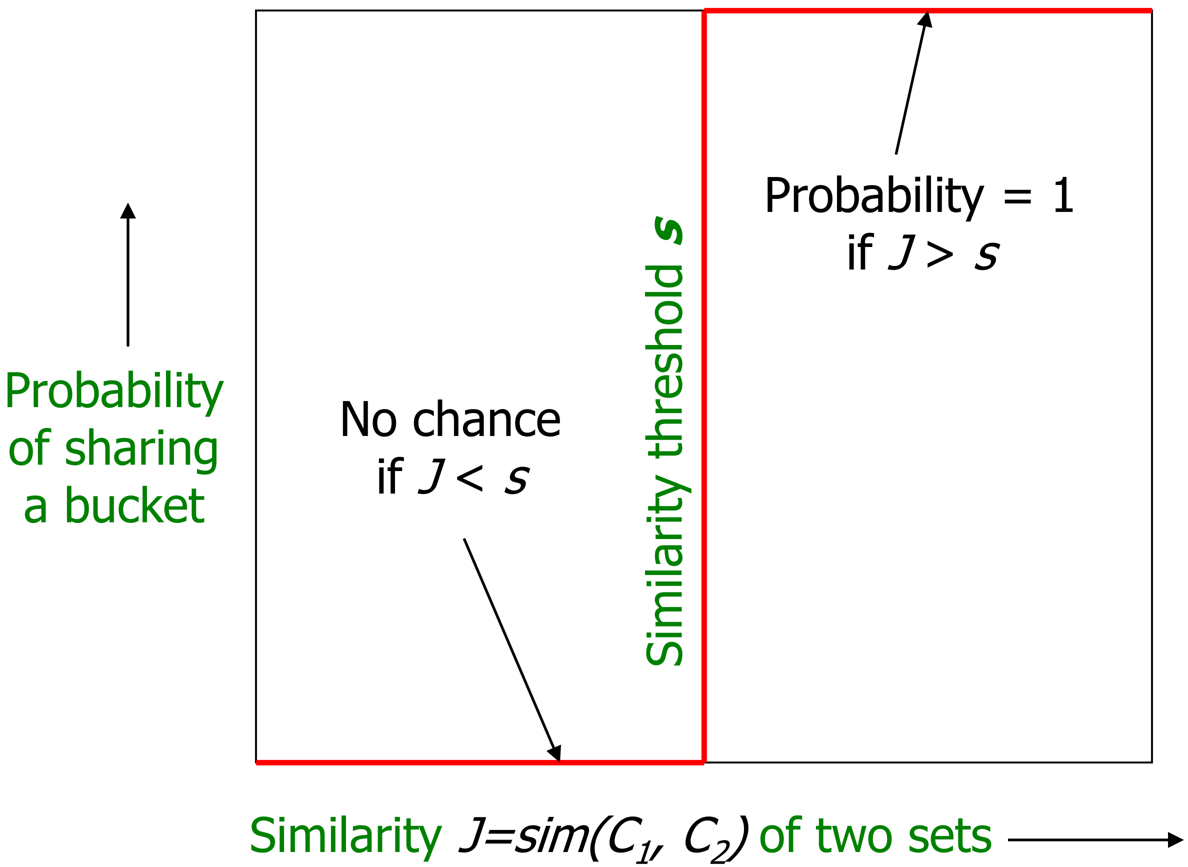
- **Probability C_1, C_2 identical in one particular band:** $(0.8)^5 = 0.328$
- Probability C_1, C_2 are ***not*** similar in all of the 15 bands: $(1\text{-}0.328)^{15} = 0.0026$
  - i.e., about 1/400th of the 80%-similar column pairs are **false negatives** (we miss them)
  - **We would find 99.74% pairs of truly similar documents**

Effect of decreasing bands

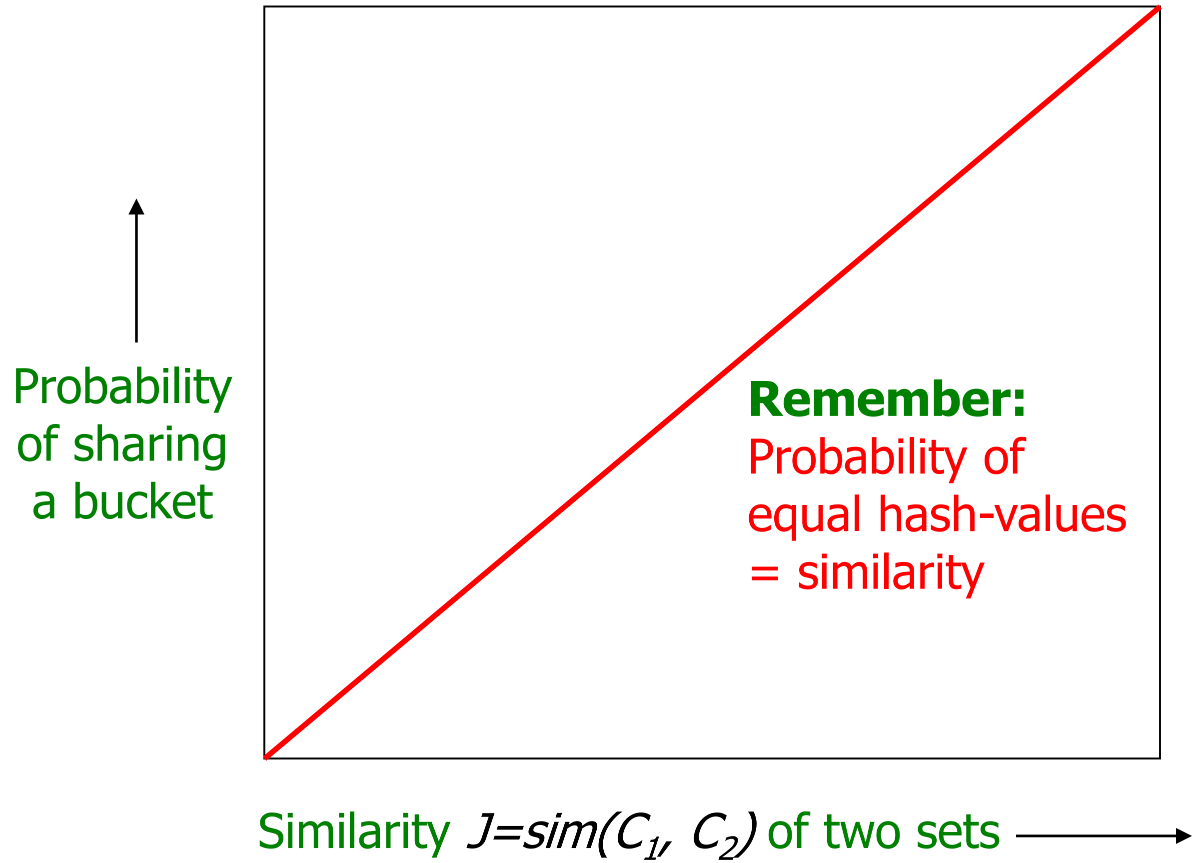
- The number of false positives goes down

- But the number of false negatives goes up
  - From 1/3000 to 1/400 (for similarity = 0.8)

LSH – What We Want

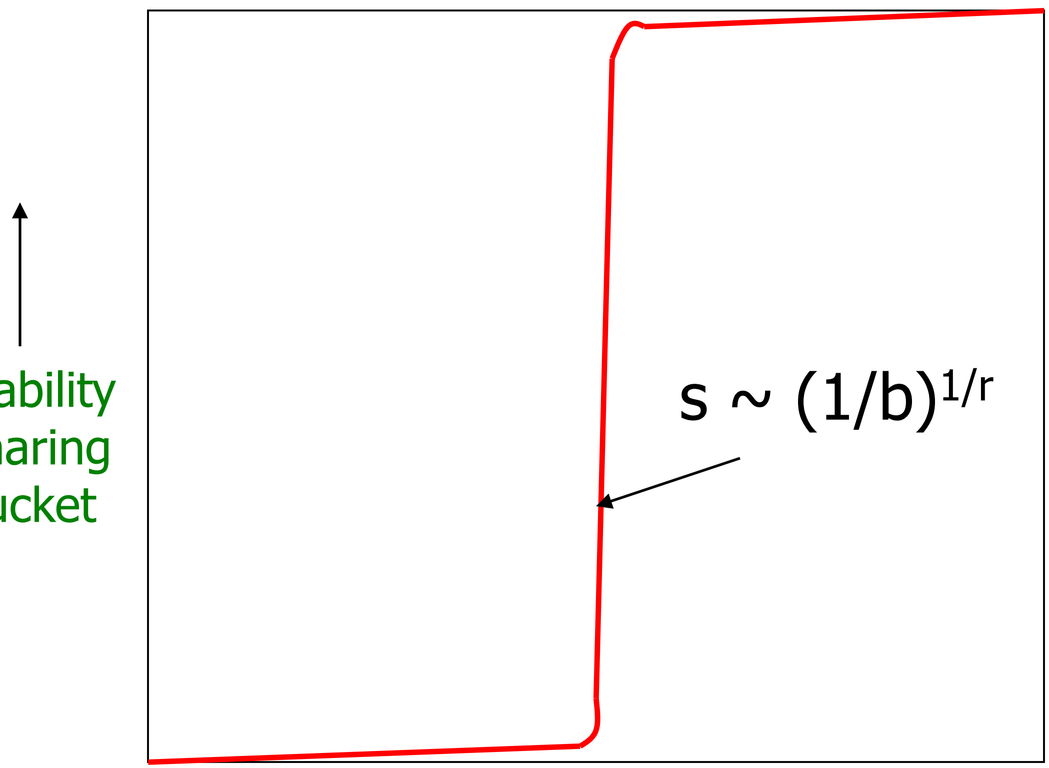


What 1 Band of 1 Row Gives You



What b Bands of r Rows Gives You

Probability of sharing a bucket



Similarity $J=sim(C_1, C_2)$ of two sets

At least one band identical

No bands identical

$$\mathbf{1 - (1 - \boldsymbol{J}^{\boldsymbol{r}})^{\boldsymbol{b}}}$$

Some row of a band unequal

All rows of a band are equal

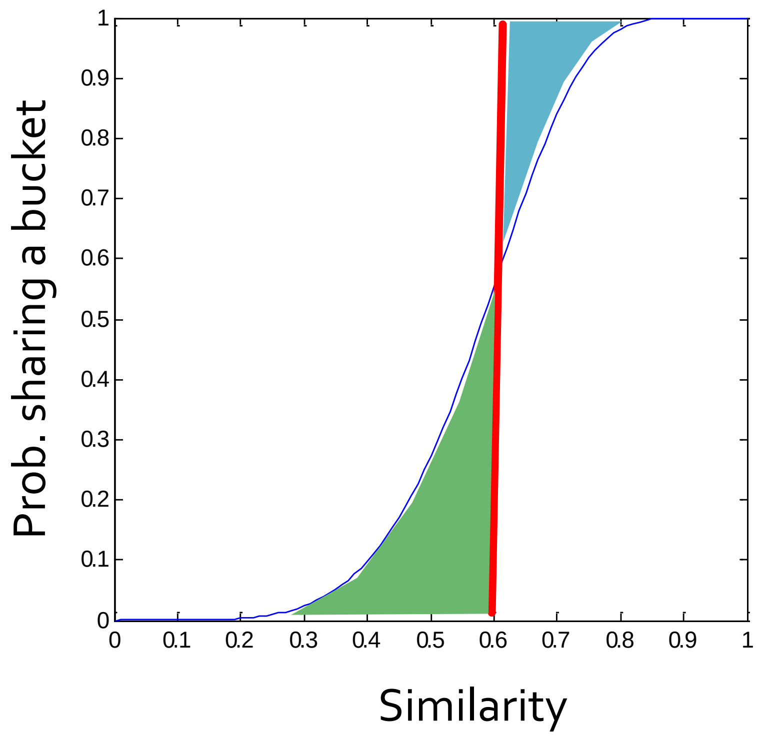
Similarity threshold s

- Example: $b = 20$; $r = 5$
- **Prob. that at least 1 band is identical:**

s	$1-(1-s^r)^b$
.2	.006
.3	.047
.4	.186
.5	.470
.6	.802
.7	.975
.8	.9996

Picking *r* and *b*: The S-curve

- **Picking *r* and *b* to get the best S-curve**
  - 50 hash-functions (r=5, b=10)



Blue area: False Negative rate
Green area: False Positive rate

Picking *r* and *b*: Example

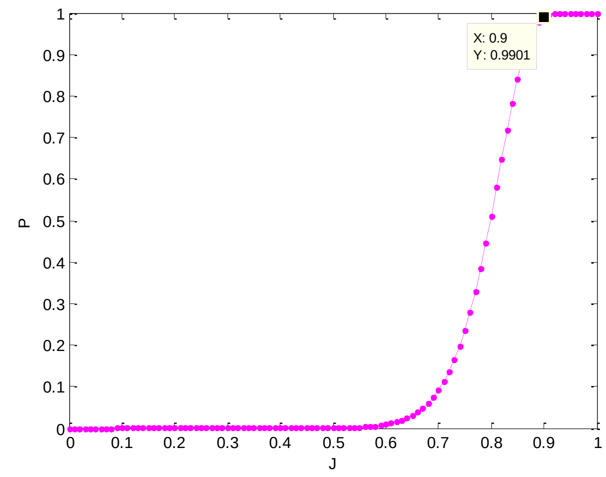
- Imagine we want to select with probability < 0.01 all objects with Jaccard similarity <=60 % ...
- AND we also want to select with probability >0.99 all objects with Jaccard similarity >=90%

- It is possible to solve the equations involving b and r to obtain their values

- The solution for our example:
  - b aprox. 20
  - r aprox. 15

Example (continuation)

- Confirming the results…
- Curve $P(J) = 1 - (1 - J^r)^b$
  - r=15
  - b=20



- Probability < 0.01 for Jaccard similarity <=0.6
- $1 - (1 - 0.6^{15})^{20} \approx 0.0095 \; < 0.01$
  - OK
- Probability >0.99 for Jaccard similarity >=0.9
- $1 - (1 - 0.9^{15})^{20} \approx 0.9901 > 0.99$
  - OK

LSH Summary

- Tune ***M, b, r*** to get almost all pairs with similar signatures
  - but eliminate most pairs that do not have similar signatures

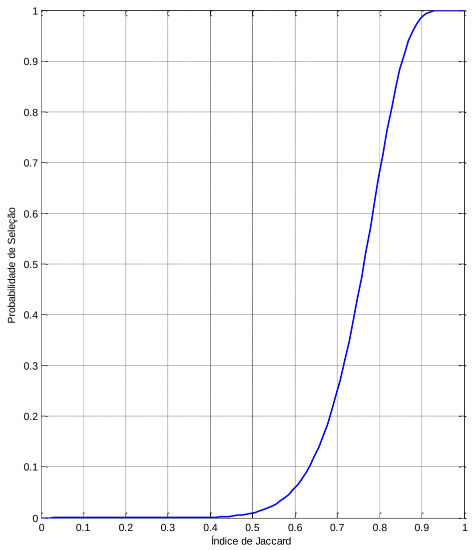
- Check in main memory that **candidate pairs** really do have **similar signatures**

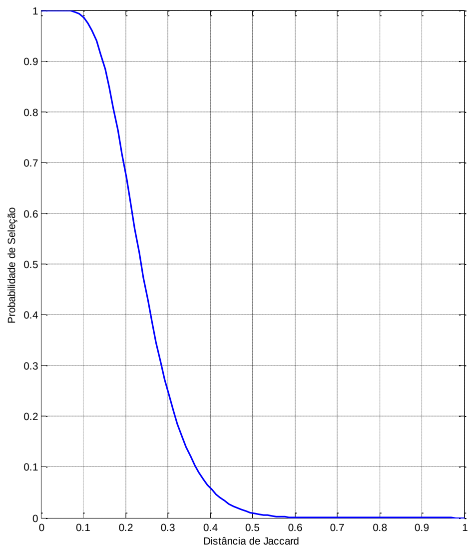
- **Optional:** In another pass through data, check that the remaining candidate pairs really represent similar documents

Application to MovieLens

- Process the MinHash matrix (explained before)
  - They have been calculated previously

- Lets use r=10   b=NumHashFunctions /r





demo ...

Part of the slides Adapted from:
Finding Similar Items: Locality Sensitive Hashing

Mining of Massive Datasets
Jure Leskovec, Anand Rajaraman, Jeff Ullman
Stanford University
http://www.mmds.org