

Near Duplicate Detection

UCSB 293S, 2022

Tao Yang

Some of slides are from text book [MRS] and
Rajaraman/Ullman's data mining book

Table of Content

- **Motivation**
- **Shingling for duplicate comparison**
- **Minhashing**
- **LSH (Location-Sensitive Hashing)**

Applications of Duplicate Detection and Similarity Computing

- **Duplicate and near-duplicate documents occur in many situations**
 - Copies, versions, plagiarism, spam, mirror sites
 - 30-60+% of the web pages in a large crawl can be exact or near duplicates of pages in the other 70%
 - Duplicates consume significant resources during crawling, indexing, and search
- **Similar query suggestions**
- **Advertisement: coalition and spam detection**
- **Product recommendation based on similar product features or user interests**

Exact Duplicate Detection

- **Exact duplicate detection is relatively easy**
 - Content fingerprints
 - SHA-1, MD5, *cyclic redundancy check* (CRC)
 - **Checksum techniques**
 - A checksum is a value that is computed based on the content of the document
 - e.g., sum of the bytes in the document file
- | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|------------|
| T | r | o | p | i | c | a | l | | f | i | s | h | <i>Sum</i> |
| 54 | 72 | 6F | 70 | 69 | 63 | 61 | 6C | 20 | 66 | 69 | 73 | 68 | 508 |
- Possible for files with different text to have same checksum

Example of Near-Duplicate: News Articles

SFGate.com

SFGATE HOME • NEWS **NEW!** • BUSINESS • SPORTS • ENTERTAINMENT • TRAVEL CLASSIFIEDS • JOBS • REAL ESTATE • CARS

SEARCH SFGate Web Search by YAHOO! Sign In | Register

Ap Associated Press

Obama Takes on Question of Faith

By NEDRA PICKLER, Associated Press Writer
Monday, January 21, 2008

PRINTABLE E-MAIL SHARE COMMENTS (0) FONT | SIZE: - + TOOLS SPONSOR: verizon wireless

(01-21) 04:22 PST Columbia, S.C. (AP) --

Barack **Obama** is stepping up his effort to correct the misconception that he's a Muslim now that the presidential campaign has hit the Bible Belt.

At a rally to kick off a weeklong campaign for the South Carolina primary, **Obama** tried to set the record straight from an attack circulating widely on the Internet that is designed to play into prejudices against Muslims and fears of terrorism.

MOST READ **MOST E-MAILED** **TOP STORIES**

1. TGI Friday's employee found slain in San Mateo restaurant
2. Girl shot to death in Oakland by boy trying to scare her, police say
3. 5 Dead As Planes Collide in SoCal
4. Rainy week ahead for Bay Area, with snow on the hills
5. Cranky Pants traded in for Gary Coleman's
6. Giants want to develop lot next to AT&T Park
7. More men turning to implants for chests of gold

HOME PAGE MY TIMES TODAY'S PAPER VIDEO MOST POPULAR TIMES TOPICS Get Home Delivery Log In Register Now

The New York Times U.S.

U.S. All NYT Search Ameriprise Financial

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE HEALTH SPORTS OPINION ARTS STYLE TRAVEL JOBS REAL ESTATE AUTOS

POLITICS WASHINGTON EDUCATION

Obama Takes on Question of Faith

By THE ASSOCIATED PRESS
Published: January 21, 2008

Filed at 7:16 a.m. ET

COLUMBIA, S.C. (AP) -- Barack Obama is stepping up his effort to correct the misconception that he's a Muslim now that the presidential campaign has hit the Bible Belt.

At a rally to kick off a weeklong campaign for the South Carolina primary, Obama tried to set the record straight from an attack circulating widely on the Internet that is designed to play into prejudices against Muslims and fears of terrorism.

SIGN IN TO E-MAIL OR SAVE THIS
PRINT
ARTICLE TOOLS SPONSORED BY THE SAVAGES

MOST POPULAR
E-MAILED BLOGGED SEARCHED

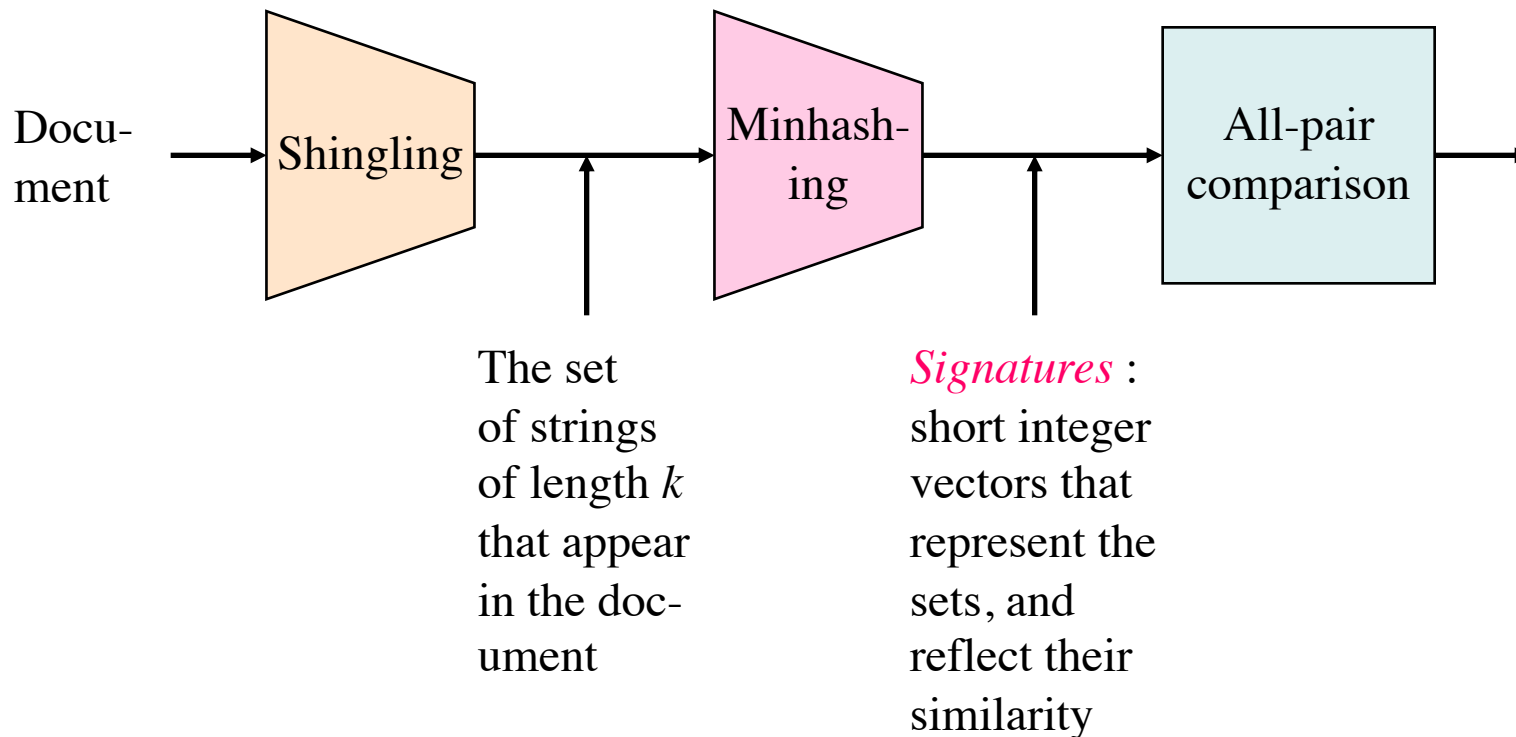
1. Nicholas D. Kristof: Hillary, Barack, Experience
2. Paul Krugman: Debunking the Reagan Myth
3. Pregnancy Problems Tied to Caffeine
4. Maureen Dowd: Red, White and Blue Tag Sale
5. Roger Cohen: U.S. Soldiers and Shoppers Hit the Wall
6. Stocks Plunge Worldwide on Fears of a U.S. Recession
7. New York Measuring Teachers by Test Scores
8. Op-Ed Contributor: Radical Love Gets a Holiday
9. A Cutting Tradition

Near-Duplicate Detection

- **More challenging task**
 - Are web pages with same text context but different advertising or format near-duplicates?
- ***Near-Duplication: Approximate match***
 - Compute syntactic similarity with an edit-distance measure
 - Use similarity threshold to detect near-duplicates
 - E.g., Similarity > 80% => Documents are “near duplicates”
 - Not transitive though sometimes used transitively
 - Expensive to find all near-duplicate pairs in N documents. $O(N^2)$ comparisons

Two Techniques for Faster Similarity Computation

1. **Shingling** : convert text documents to fingerprint sets.
2. **Minhashing** : convert a large set of fingerprints to short signatures, while preserving similarity.



Computing Similarity with Shingles

- **Shingles (n-gram terms) [Brin95, Brod98]**

Document “a rose is a rose is a rose” =>

a_rose_is_a

rose_is_a_rose

is_a_rose_is

- **Derive a set of shingles for each document**
- **Measure similarity between two docs (= sets of shingles)**
 - $\text{Size_of_Intersection} / \text{Size_of_Union}$

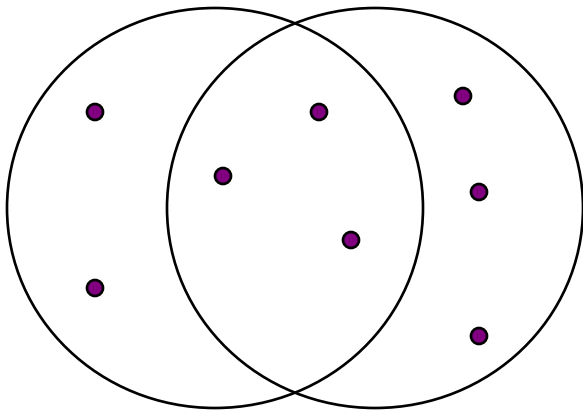


Jaccard measure

The diagram shows a light blue rectangular box with a black border. Inside the box, the text 'Jaccard measure' is written in black. Above the box, there is a black line that forms a wide, shallow 'V' shape, with two short horizontal lines extending from the top of the box to the ends of the 'V' shape.

Jaccard similarity to measure resemblance

- The **Jaccard similarity** of two sets is the size of their intersection divided by the size of their union.
 - $Sim(C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|$.



3 in intersection.

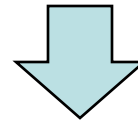
8 in union.

Jaccard similarity
= $3/8$

Fingerprint Example for Web Documents

Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species.

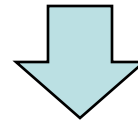
(a) Original text



Shingling

tropical fish include, fish include fish, include fish found, fish found in, found in tropical, in tropical environments, tropical environments around, environments around the, around the world, the world including, world including both, including both freshwater, both freshwater and, freshwater and salt, and salt water, salt water species

(b) 3-grams



Hashing

938 664 463 822 492 798 78 969 143 236 913 908 694 553 870 779

(c) Hash values

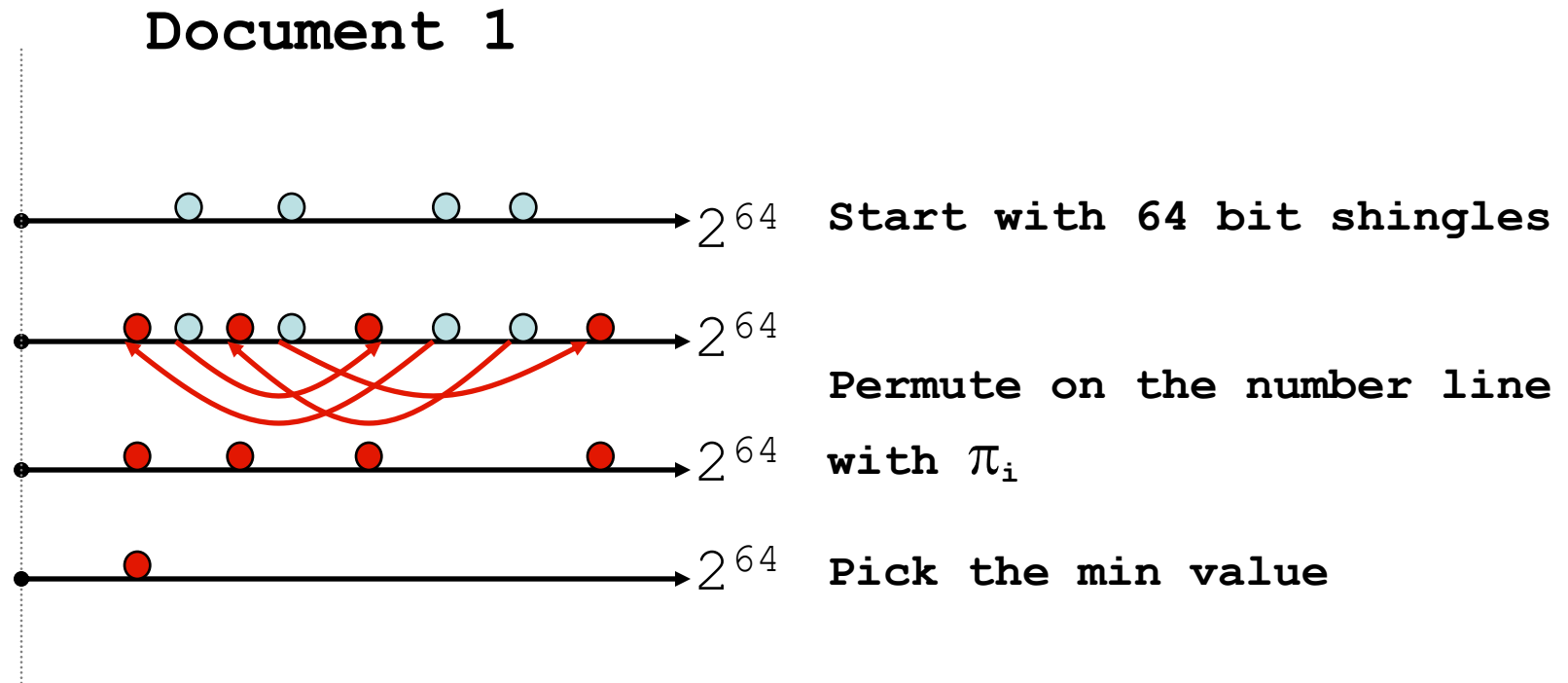
Steps of General Fingerprint Generation with Shingling for Web Pages and Text Documents

1. The document is parsed into words. Non-word content, such as punctuation, HTML tags, and additional whitespace, is removed.
2. The words are grouped into contiguous *n-grams* for some *n*. These are usually overlapping sequences of words, although some techniques use non-overlapping sequences.
3. Some of the *n-grams* are selected to represent the document.
4. The selected *n-grams* are hashed to improve retrieval efficiency and further reduce the size of the representation.
5. The hash values are stored, typically in an inverted index.
6. Documents are compared using overlap of fingerprints

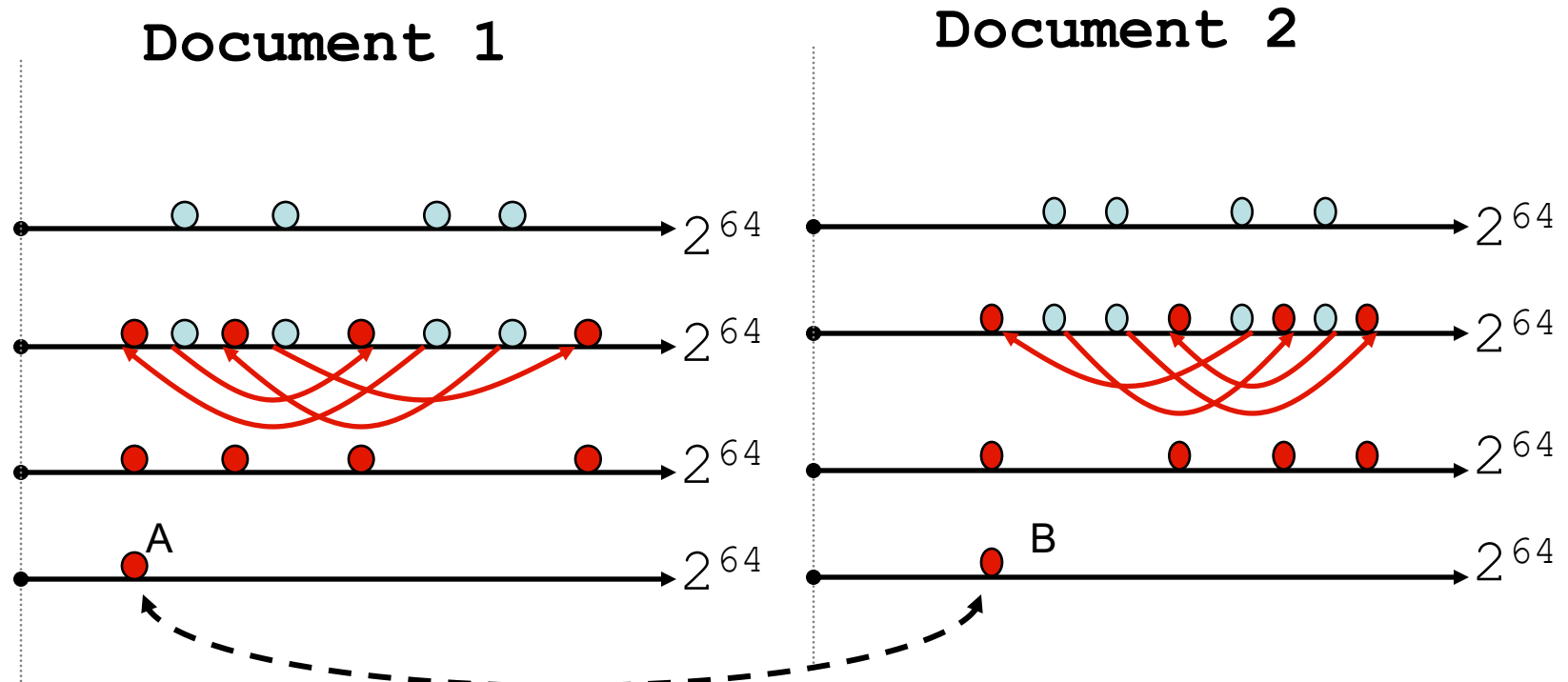
Approximated Representation with Sketching and Minhashing

- Computing exact set intersection of shingles between all pairs of documents is expensive
 - Approximate using a subset of shingles (called sketch vectors) for each document
 - Create a sketch vector for doc d using minhashing.
 - Each element $\text{sketch}_d[i]$ is computed as follows:
 - Let f map all shingles in the universe to $0..2^m$
 - Let π_i be a specific random permutation on $0..2^m$
 - Pick $\text{MIN } \pi_i(f(s))$ over all shingles s in this document d
 - Repeat above process for n rounds to have a sketch vector of size n
 - Documents which share more than t (say 80%) in sketch vector's elements are **similar**

Computing Sketch[i] for Doc1 with Minhashing



Test if $\text{Doc1.Sketch}[i] = \text{Doc2.Sketch}[i]$



Are these equal?

Test for $i=1,2, \dots, 200$ random permutations: $\pi_1, \pi_2, \dots, \pi_{200}$

Example: Permutation and Min-hash

Original shingle ordering = banana < cat < dog < mouse

Mapping function $f(x) = x$

Round 1:

ordering after permutation $\pi_1 = \text{cat} < \text{dog} < \text{mouse} < \text{banana}$

Document 1 with unigram
shingle: {mouse, dog}

With π_1

MH-signature = dog

Document 2 with unigram
shingle : {cat, mouse}

With π_1

MH-signature = cat

Example: Min-hash with another hashing function (permutation)

Original shingle ordering = banana < cat < dog < mouse

Mapping function $f(x) = x$

Round 2:

ordering after permutation π_2 = banana < mouse < cat < dog

Document 1 with unigram shingle: {mouse, dog}

With π_2

MH-signature = mouse

Document 2 with unigram shingle : {cat, mouse}

With π_2

MH-signature = mouse

Approximated similarity after two rounds with $\pi_1, \pi_2 = 1/2$

Summary: Shingling with Minhashing

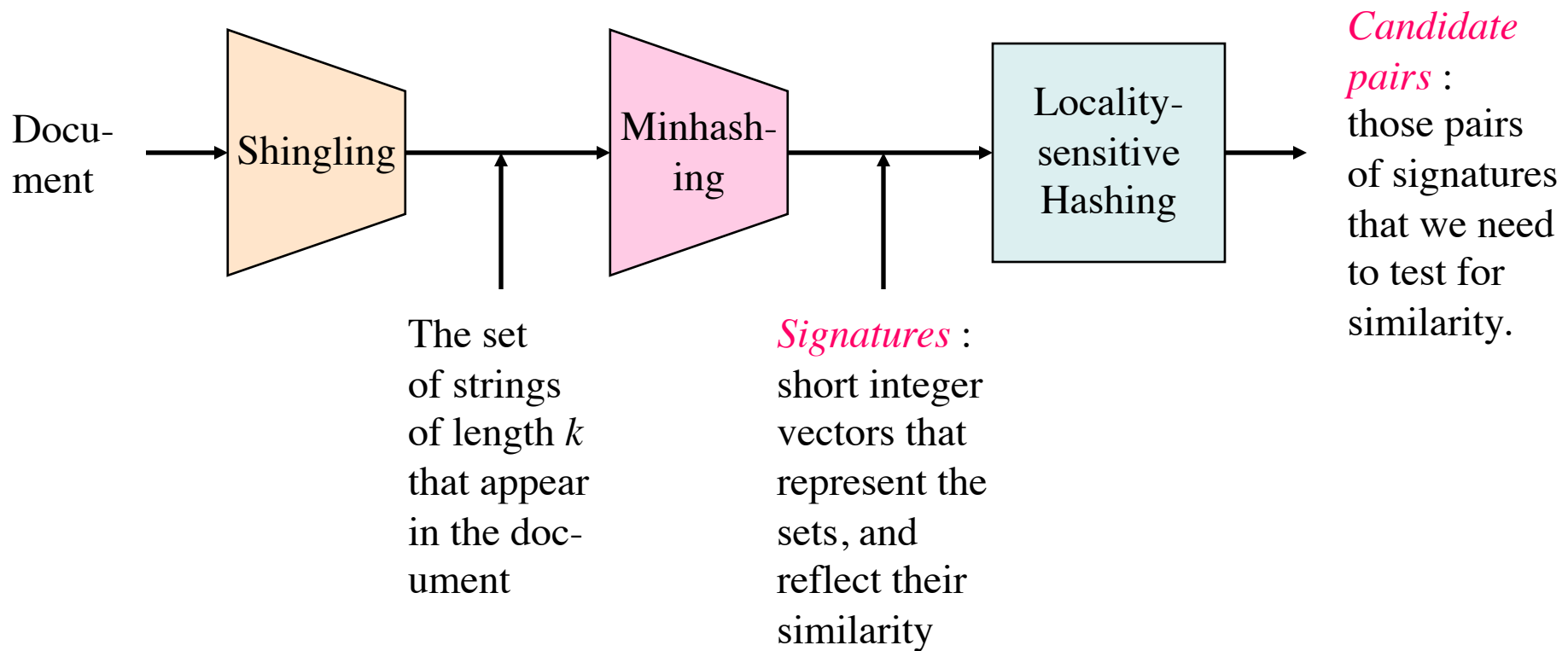
- **Given two documents d_1, d_2 .**
- **Let S_1 and S_2 be their shingle sets**
 - Document Resemblance =
$$|\text{Intersection of } S_1 \text{ and } S_2| / |\text{Union of } S_1 \text{ and } S_2|.$$
- **Let $\text{Alpha} = \min (\pi (f(S_1)))$ $\text{Beta} = \min (\pi(f(S_2)))$**
- **Probability ($\text{Alpha} = \text{Beta}$) = Resemblance**
 - Computing this by sampling (e.g. 200 times).
 - For example, 100 times are equal out of 200 samplings.
 - \rightarrow Resemblance (document similarity) is 0.5
- Sometime we use one mapping function as a combination of two functions $\pi(f())$

Locality-Sensitive Hashing

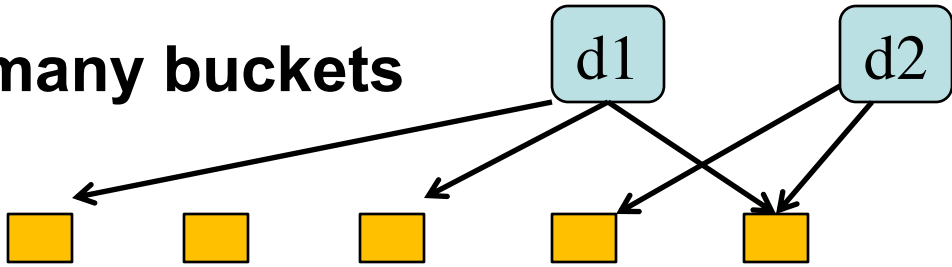
All-pair comparison is expensive

- We want to compare objects, finding those pairs that are sufficiently similar.
- Complexity of comparing the signatures of all pairs of objects is quadratic in the number of objects
- **Example:** 10^6 objects implies $5 \cdot 10^{11}$ comparisons.
 - At 1 microsecond/comparison: 6 days.
- **Minhashing is useful, still not fast enough. We need more sampling based techniques**

The Big Picture for Siminar Document Search/Clustering



Locality-Sensitive Hashing

- **General idea:** Create a function $f(x,y)$ that tells whether or not x and y is a *candidate pair*: a pair of elements whose similarity must be evaluated.
- **Map each document to many buckets**

```
graph TD; d1[d1] --> b1[ ]; d1 --> b3[ ]; d1 --> b4[ ]; d2[d2] --> b2[ ]; d2 --> b4[ ]; d2 --> b5[ ]
```
- **Observation:**
 - Similar documents should be mapped to one bucket after a few rounds of tries
 - Dissimilar documents should never be mapped to the same bucket
- **Make elements of the same bucket candidate pairs.**
 - $f(x,y)$ is true if x and y are mapped into the same bucket

LSH with minhash for similar document detection/clustering

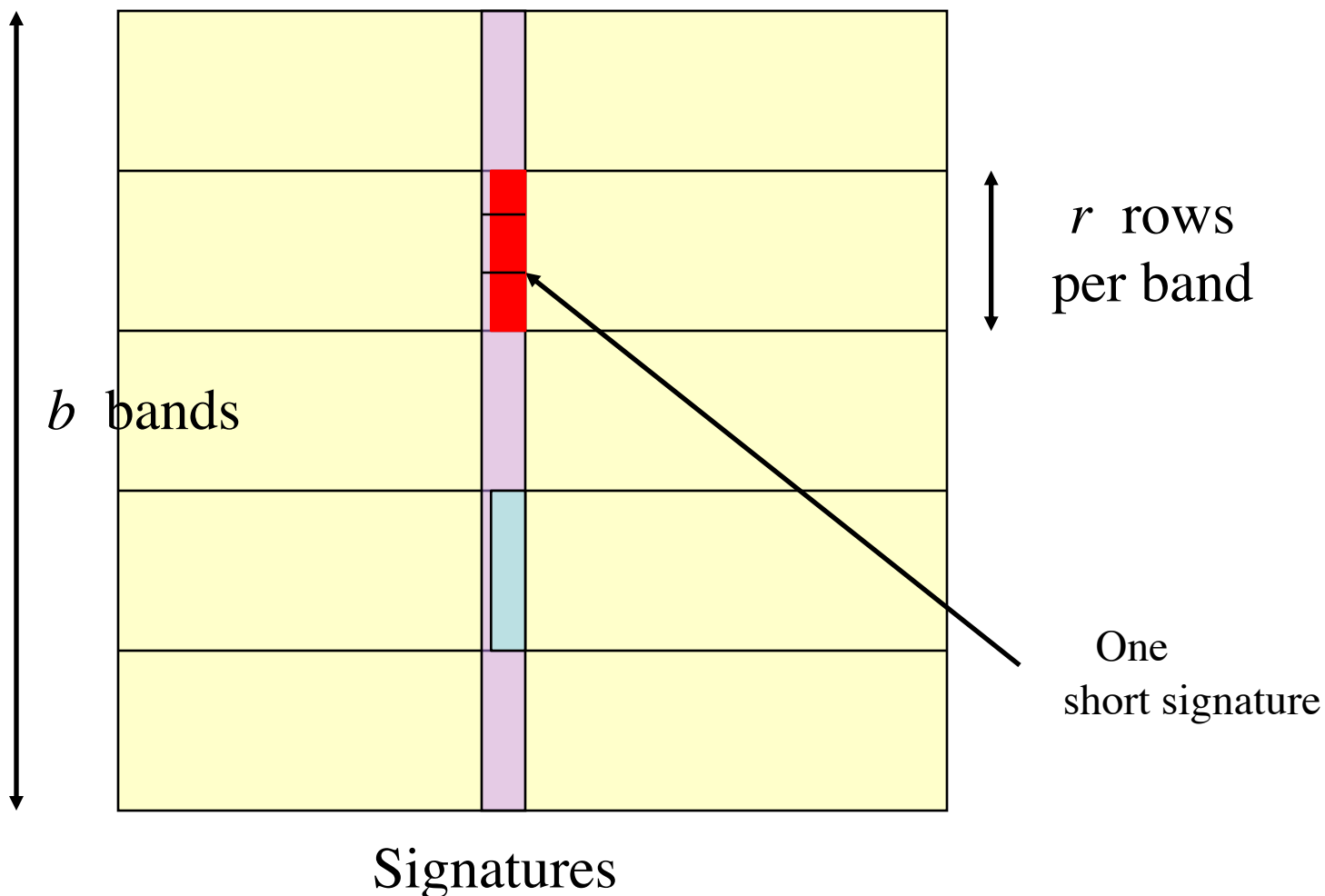
- Generate *a set of* LSH signatures for each doc to produce b bands of signatures. Each band uses r of the min-hash values
- For $i = 1$ to b**
- Randomly select r min-hash functions and concatenate their values to form i 'th LSH signature (called band)
- Pair (u, v) is a candidate to be similar if u and v have an LSH signature in common in any round (i.e. one of the bands)
 - Parameter r is the length of each band; b is the number of bands
 - Property
 - $\Pr(\text{lsh}(u) = \text{lsh}(v)) = [\Pr(\text{minhash}(u) = \text{minhash}(v))]^r$
 - Notice we use the same minhash functions to compare u and v
 - Documents u and v are not similar if their LSH signatures are not same for all b rounds of their LSH signature comparison

LSH Illustration: Produce signature with bands

$$\Pr(\text{lsh}(u) = \text{lsh}(v)) = \Pr(\text{mh}(u) = \text{mh}(v))^r$$

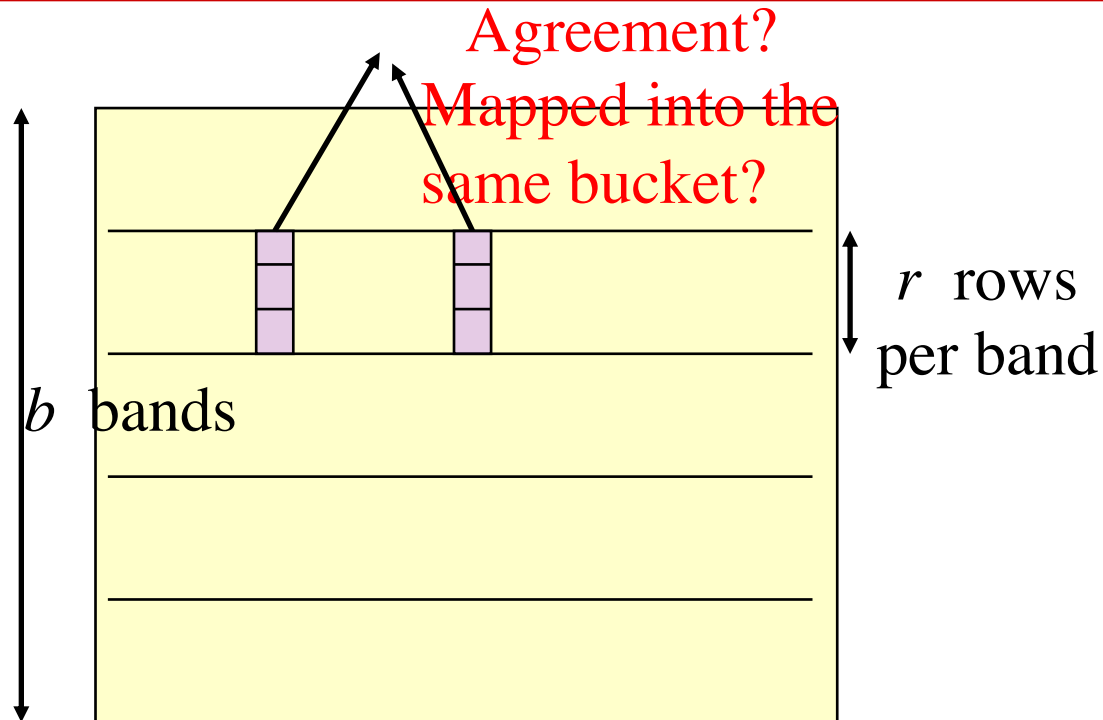
Create b
bands for each
document

Tune b and r
to catch most
similar pairs,
but few
nonsimilar
pairs.

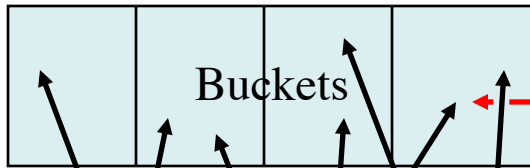


Signature agreement of each pair at each band

- Signature of doc u and v in the same band agrees \rightarrow a candidate pair
- Use r minhash values (r rows) each band
 - Band length is r



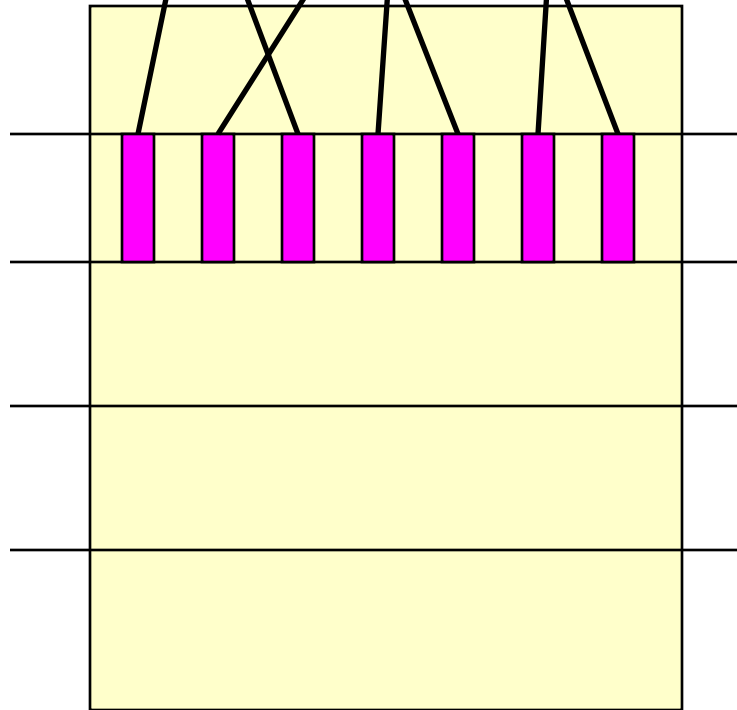
$$\Pr(\text{lsh}(u) = \text{lsh}(v)) = [\Pr(\text{minhash}(u) = \text{minhash}(v))]^r$$



Docs 2 and 6
are probably identical.

Matrix M

Docs 6 and 7 are
surely different.



r rows

b bands

Example: LSH with minhashing $b=2, r=3$

Get 4 MIN hash values to compose for LSH signatures. Then derive $b=2$ LSH signatures and each uses $r=3$ MIN hash values

Document 1:

{mouse, dog, horse, ant}

$MH_1 = \text{horse}$

$MH_2 = \text{mouse}$

$MH_3 = \text{ant}$

$MH_4 = \text{dog}$

$LSH_{134} = \text{horse-ant-dog}$

$LSH_{234} = \text{mouse-ant-dog}$

Document 2:

{cat, ice, shoe, mouse}

$MH_1 = \text{cat}$

$MH_2 = \text{mouse}$

$MH_3 = \text{ice}$

$MH_4 = \text{shoe}$

$LSH_{134} = \text{cat-ice-shoe}$

$LSH_{234} = \text{mouse-ice-shoe}$

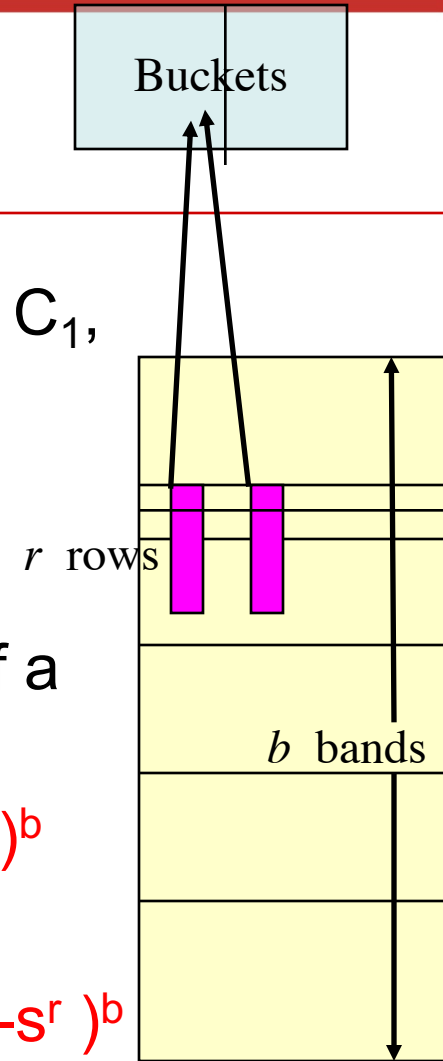
Round 1

Round 2

These two documents are not mapped into the same bucket in both rounds

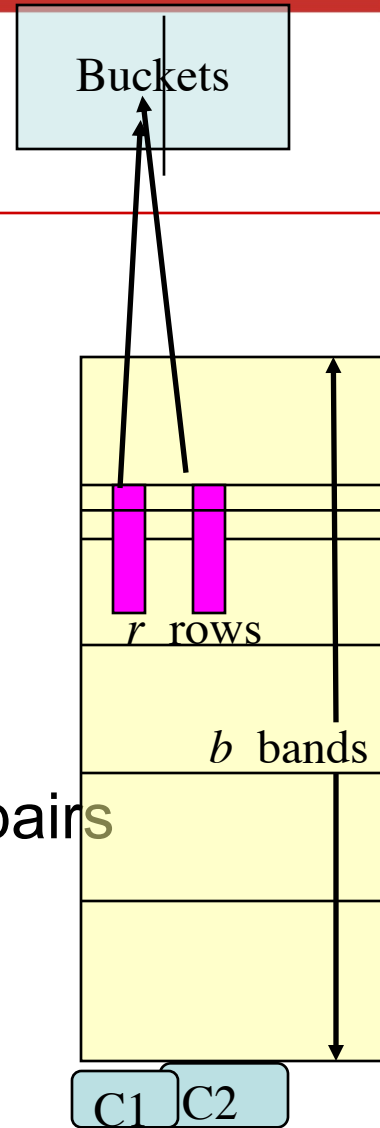
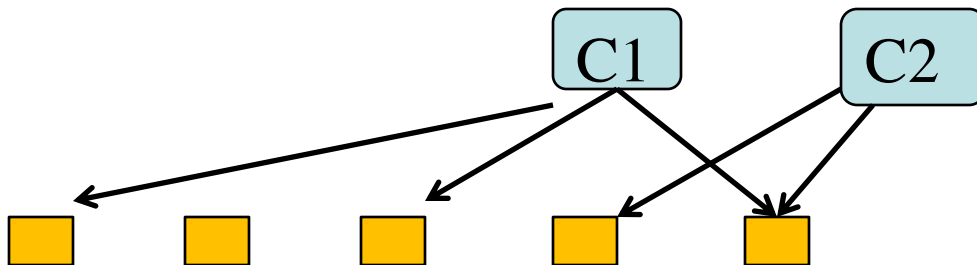
Analysis of LSH

- Probability the minhash signatures of documents C_1 , C_2 agree in one row: s
 - Threshold of two similar documents
- Probability C_1 , C_2 identical in one band: s^r
- Probability C_1 , C_2 do not agree at least one row of a band: $1-s^r$
- Probability C_1 , C_2 do not agree in all bands: $(1-s^r)^b$
 - False negative probability
- Probability C_1 , C_2 agree one of these bands: $1-(1-s^r)^b$
 - Probability that we find such a pair.

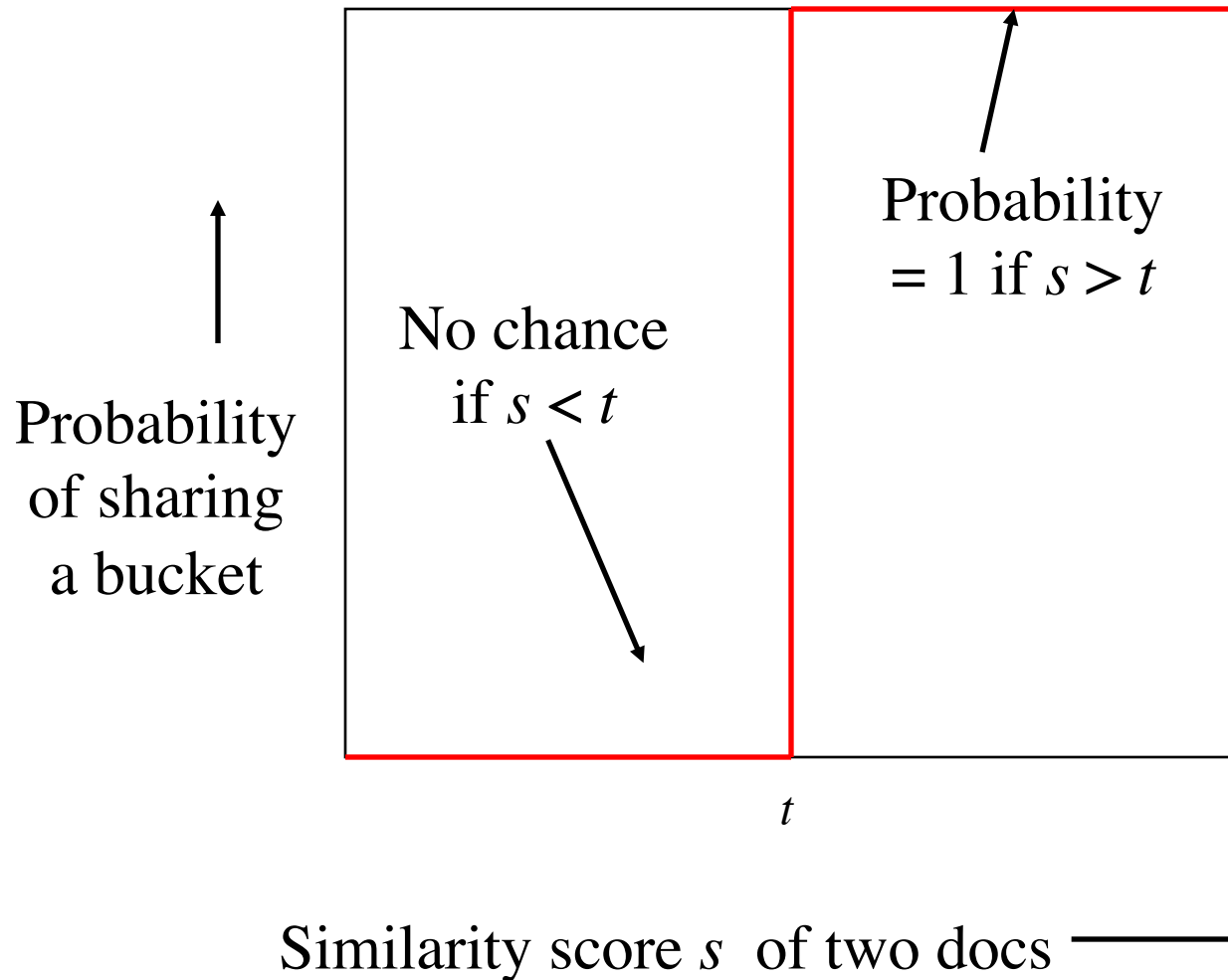


Example

- Suppose documents C_1 , C_2 are 80% Similar
- Choose $b=20$ bands of $r=5$ integers/band.
- Probability C_1 , C_2 identical in one particular band: $(0.8)^5 = 0.328$.
- Probability C_1 , C_2 are *not* similar in any of the 20 bands: $(1-0.328)^{20} = .00035$.
 - i.e., about 1/3000th of the 80%-similar column pairs are false negatives.



Analysis of LSH – What We Want



Picking r and b for the best s-curve

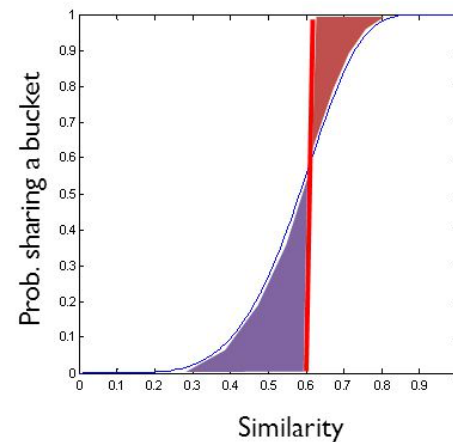
Probability of a similar pair to share a bucket

$$b = 20; r = 5$$

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 ($r=5, b=10$)



Red area: False Negative rate
Purple area: False Positive rate

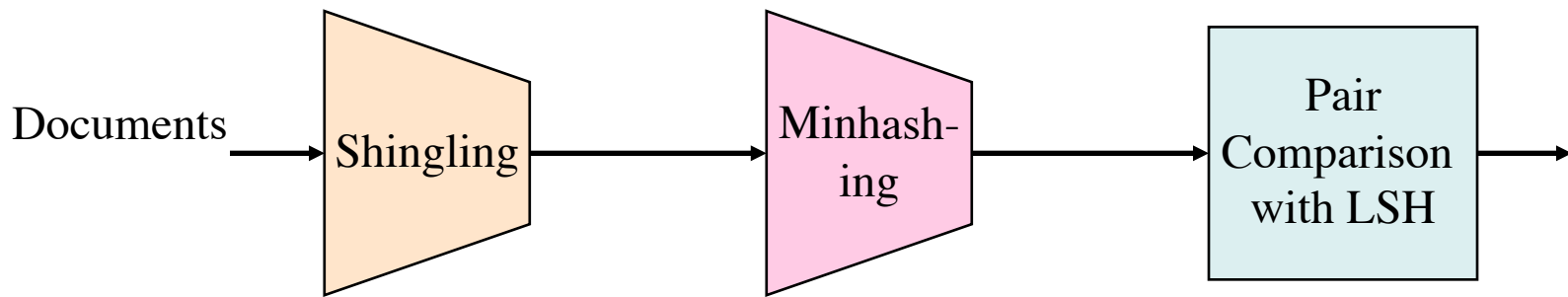
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, <http://www.mmds.org>

57

Choose $b=15$ bands of $r=5$ rows, false positives would go down, but false negatives would go up.

Shingling, MIN hashing, & LSH Summary

- **Get almost all pairs with similar signatures, but eliminate most pairs that do not have similar signatures.**
 - Check that candidate pairs really do have similar signatures.
- **LSH involves tradeoff**
 - Pick the number of minhashes, the number of bands, and the number of rows per band to balance false positives/negatives.
 - Small rounds → low false positives go down, but lower recall (false negatives would go up)



Summary

- **Shingling for duplicate comparison**
 - Signature generation with n-grams
 - Jaccard similarity to measure resemblance
- **Minhashing**
 - Reduce the number of signatures
- **LSH**
 - Reduce the complexity of similarity comparison