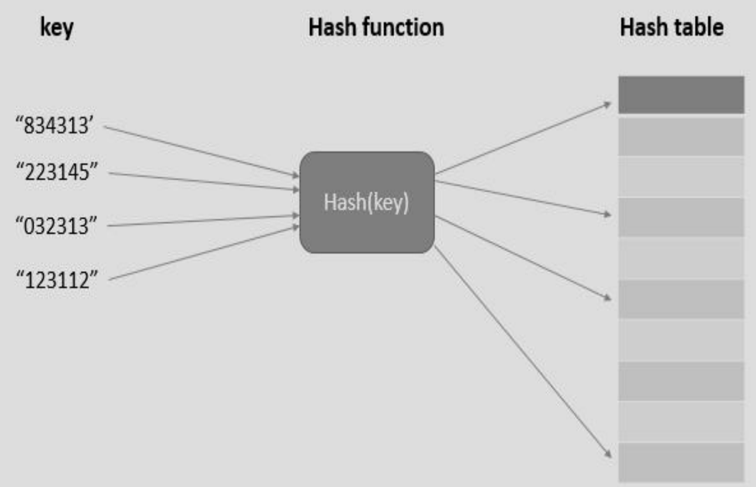


Section 4.4: Locality-Sensitive Hashing

Focus on pairs of signatures likely to be from similar documents

Content

4.4.1 LSH: First Cut

2	1	4	1
1	2	1	2
2	1	2	1

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

- **Locality-Sensitive Hashing(局部敏感哈希, 位置敏感哈希, LSH) (or called near-neighbor search, 近邻搜索),** general idea: Use a function $f(x, y)$ that tells whether x and y is a ***candidate pair*** (候选对)---a pair of elements whose similarity must be evaluated

4.4.1 Candidates from Min-Hash

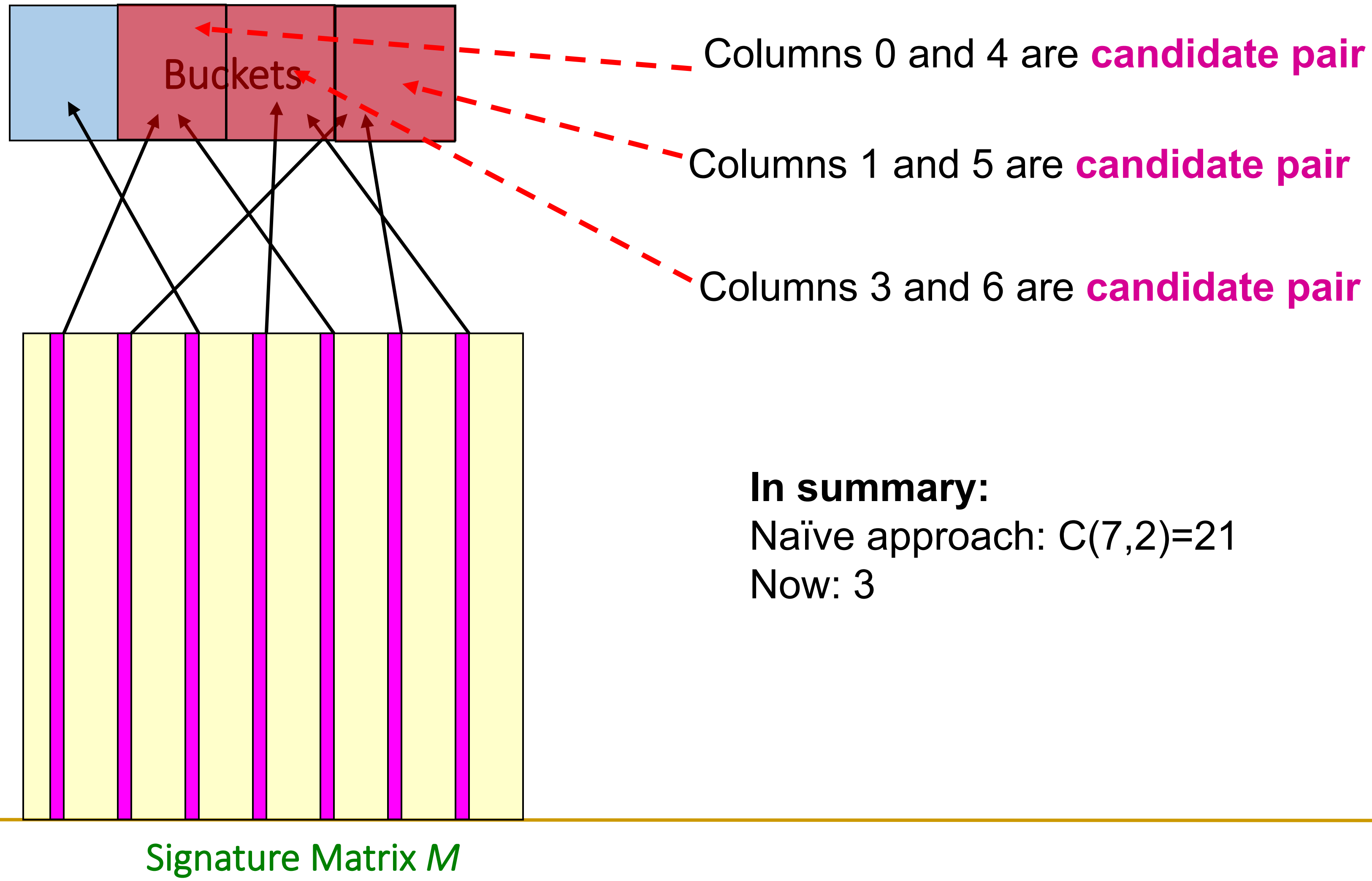
2	1	4	1
1	2	1	2
2	1	2	1

For Min-Hash matrices:

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

4.4.1 LSH example

- Assume one hash function here



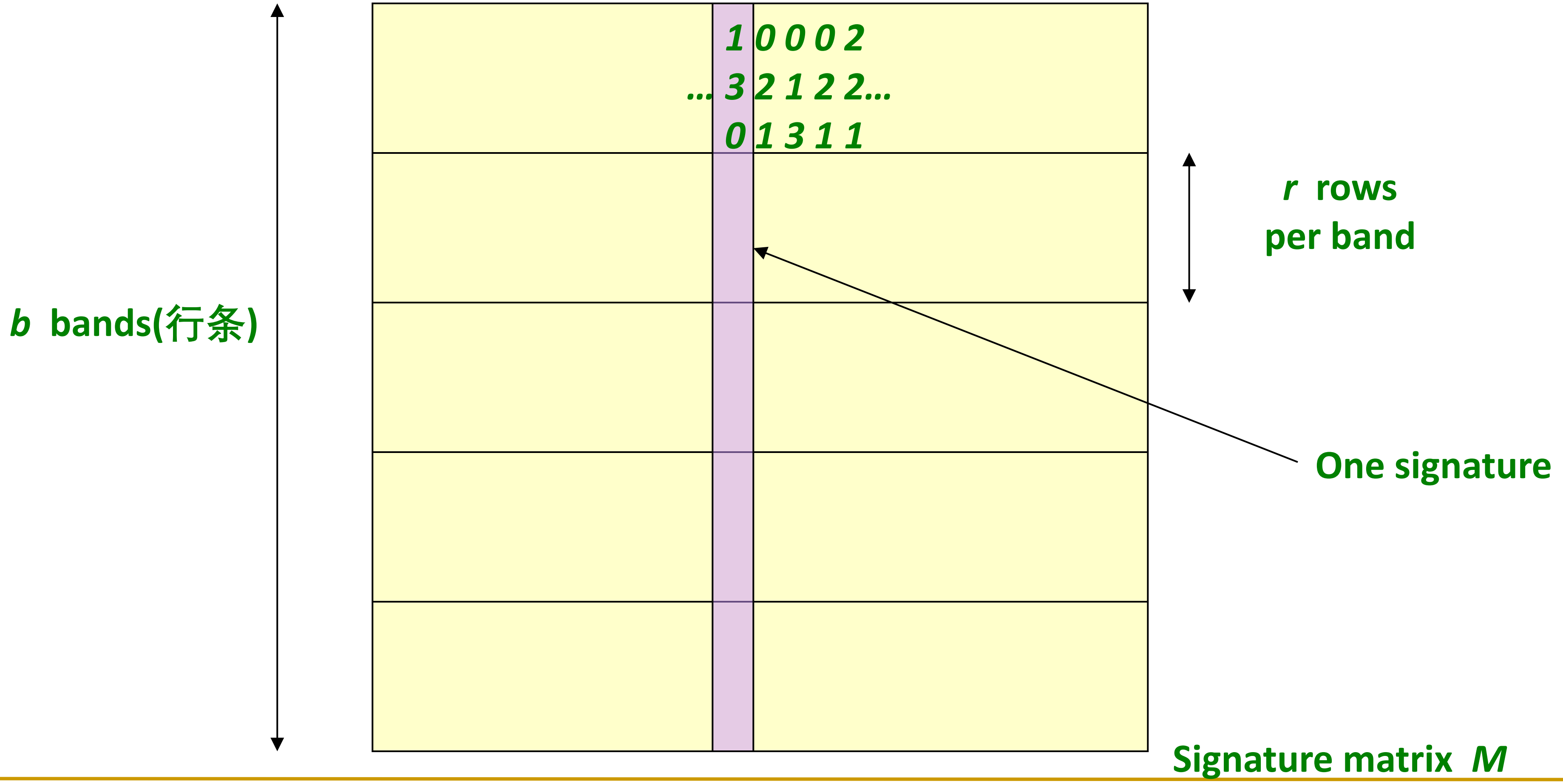
In summary:
Naïve approach: C(7,2)=21
Now: 3

4.4.1 LSH for Min-Hash

2	1	4	1
1	2	1	2
2	1	2	1

- **Big idea: Hash columns of signature matrix *M* several times** (Note this is a general approach)
- 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**

- An effective way to choose the hashing is to: **divide matrix M into b bands of r rows**

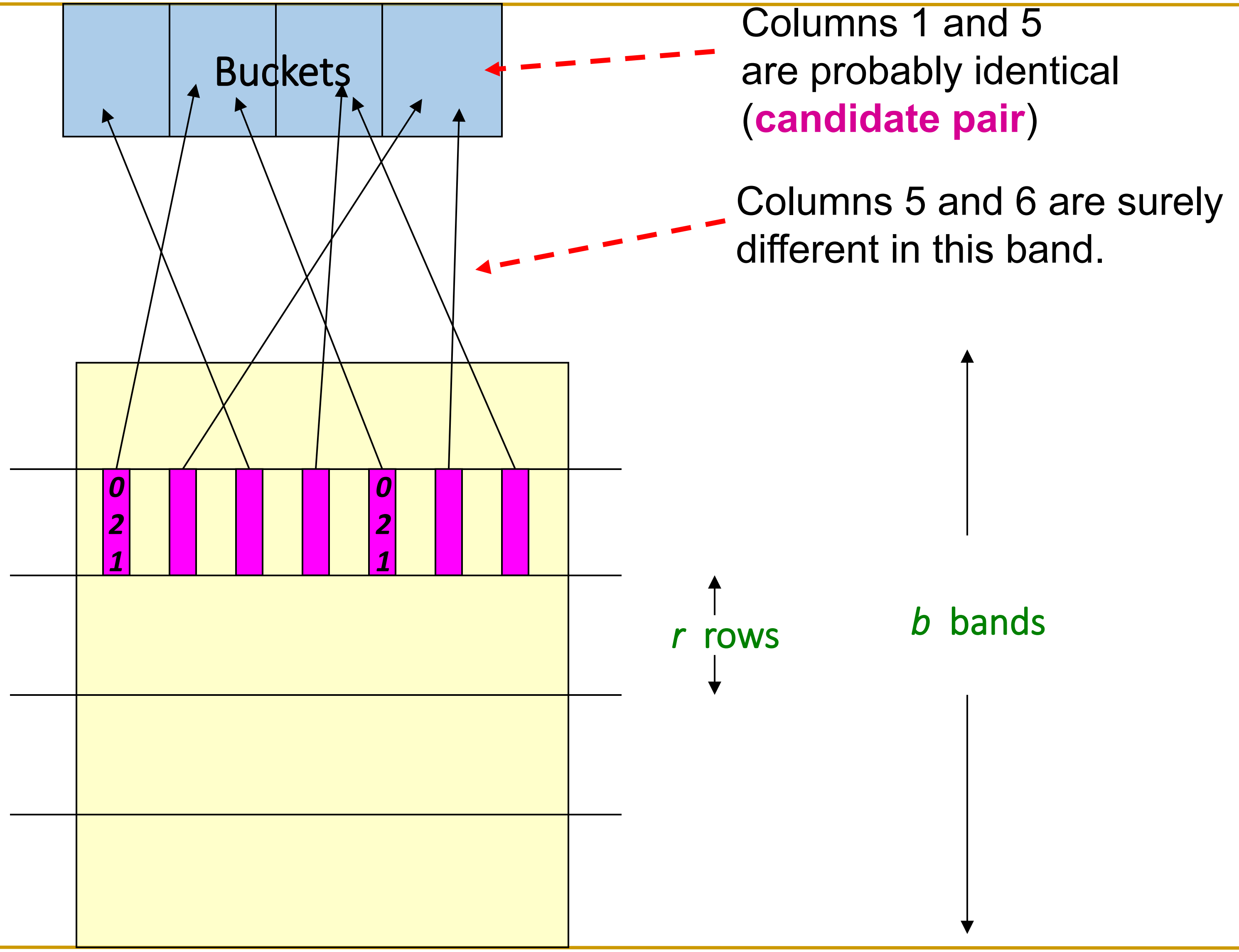


4.4.2 Partition M into Bands

- For each band, hash its portion of each column to a hash table with k buckets
  - Make k as large as possible
- We can use different hash functions for each band; Or, we can use the same hash function for all band, but use a separate bucket array for each band
  - Columns with same vector in same band will hash to the same bucket.
  - Columns with same vector in different bands, not hash to the same bucket.
- **Candidate 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

Note: ***b*** **bands,** ***r*** **rows per band**

4.4.2 Hashing Bands Example



Signature matrix M

4.4.3 Analysis of LSH

- **Simplifying Assumption**: 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
- **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

4.4.3 C_1, C_2 are 80% Similar

2	1	4	1
1	2	1	2
2	1	2	1

- **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 is identical)
- **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 20 bands: $(1-0.328)^{20} = 0.00035$
  - i.e., about 1/3000th of the 80%-similar column pairs are **false negatives** (伪反例, we miss them)
  - **We would find 99.965% pairs of truly similar documents**

False positive(伪正例):某些文档对不是相似的, 但它被认为是相似的

4.4.3 C_1, C_2 are 30% Similar

2	1	4	1
1	2	1	2
2	1	2	1

- **Find pairs of ≥ *s*=0.8 similarity, set b=20, r=5**
- **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)
- **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 20 bands: $1 - (1 - 0.00243)^{20} = 0.0474$
  - In other words, approximately 4.74% pairs of docs with similarity 0.3% end up becoming **candidate pairs**
    - They are **false positives** (伪正例) since we will have to examine them (they are candidate pairs) but then it will turn out their similarity is below threshold **s**

False negative(伪反例):某些文档对是相似的, 但它却认为是不相似的

4.4.3 LSH Involves a Tradeoff

2	1	4	1
1	2	1	2
2	1	2	1

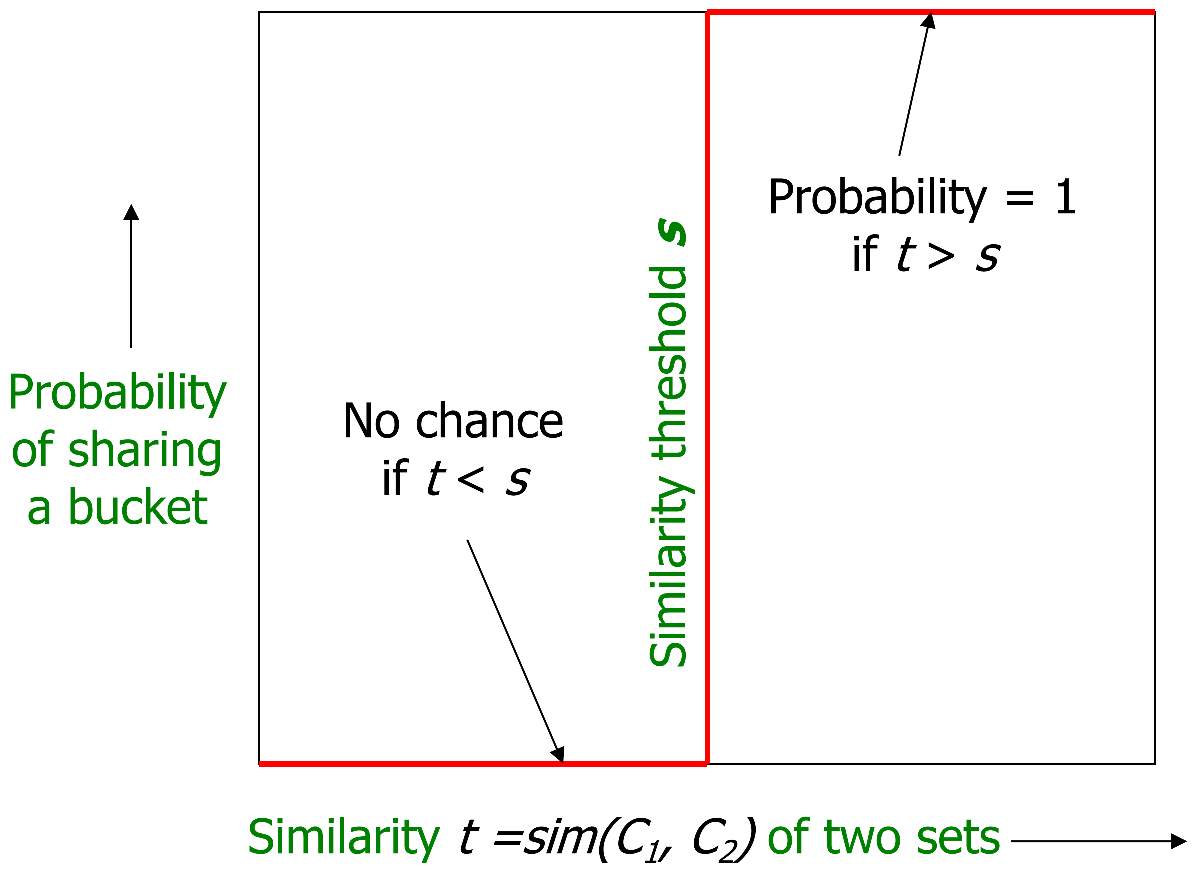
- **Pick:**
  - The number of Min-Hashes (rows of ***M***)
  - The number of bands ***b***, and
  - The number of rows ***r*** per band

  to balance false positives(伪正例)/negatives(伪反例)

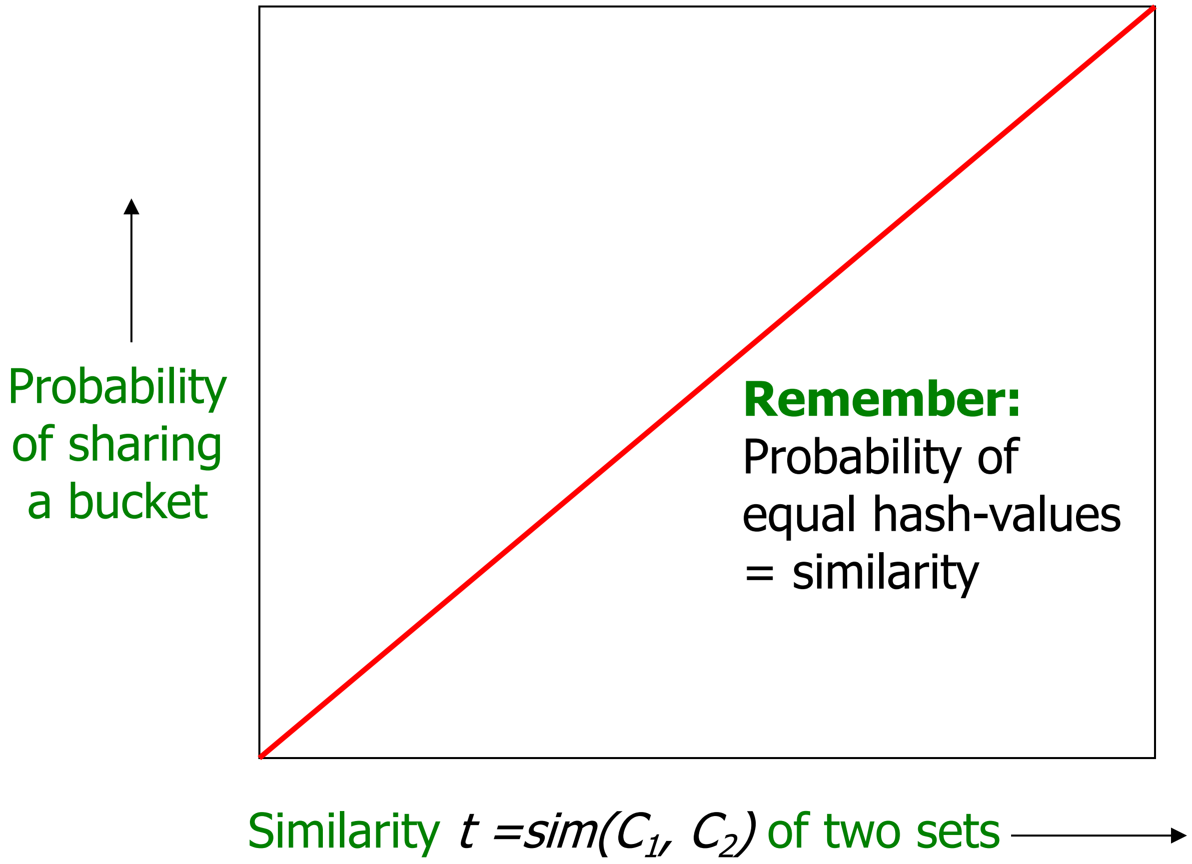
- **Example:** If we had only 15 bands of 5 rows, the number of false positives would go down, but the number of false negatives would go up

False positive(伪正例):某些文档对不是相似的, 但它被认为是相似的
False negative(伪反例):某些文档对是相似的, 但它却认为是不相似的

4.4.3 Analysis of LSH – What We Want



4.4.3 What 1 Band of 1 Row Gives You



4.4.3 *b* bands, *r* rows/band

Note: ***b*** **bands,** ***r*** **rows per band**

- Columns C_1 and C_2 have similarity t
- Pick any band (r rows)
  - Prob. that all rows in band equal = t^r
  - Prob. that some row in band unequal = $1 - t^r$
- Prob. that no band identical = $(1 - t^r)^b$
- Prob. that at least 1 band identical (签名在最少一个行条中全部相等的概率, 也就是成为候选对的概率) = $1 - (1 - t^r)^b$

4.4.3 What b Bands of r Rows Gives You

成为候选对的概率:

At least one band identical

No bands identical

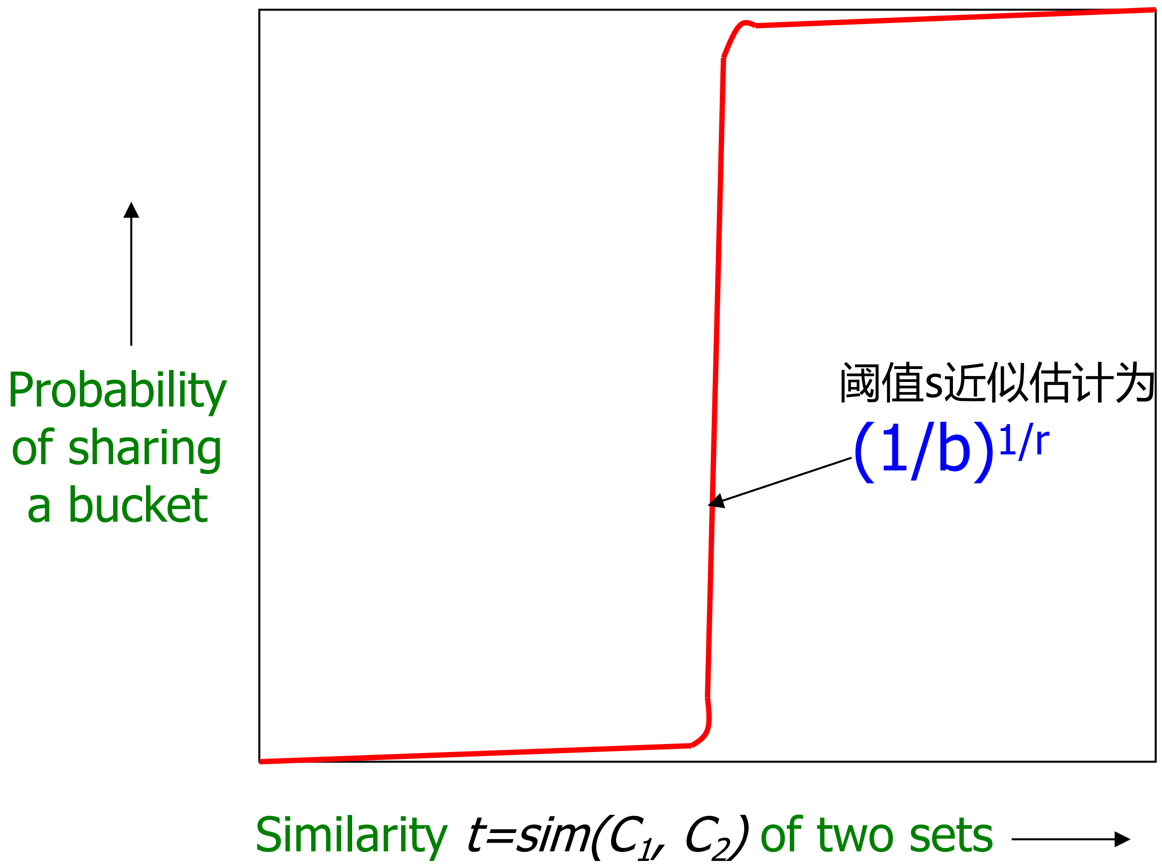
$$1 - (1 - t^r)^b$$

Some row of a band unequal

All rows of a band are equal

$b = 20$; $r = 5$

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



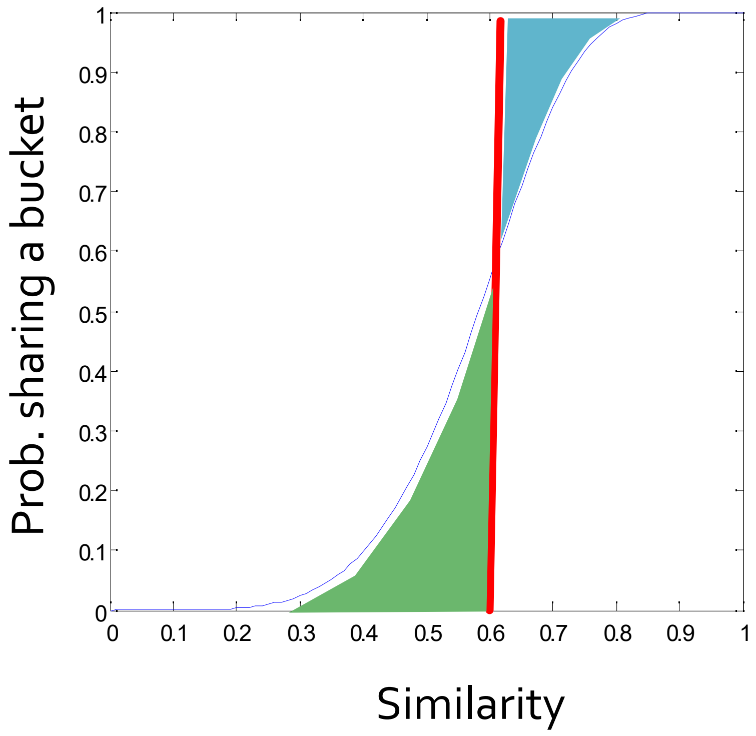
S-curve (S 曲线)

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

❑**Picking *r* and *b* to get the best S-curve**

阈值s近似估计为 $(1/b)^{1/r}$

- 50 hash-functions (r=5, b=10)



- 如果避免伪反例的产生很重要, 蓝色区域要降低(红色线条右移), 选择合适的b和r以产生小于s的阈值;
- 如果速度很重要且限制伪正例的数量, 绿色区域要减少(红色线条左移), 选择合适的b和r以获得更高的阈值

Green area: False Positive(伪正例) rate
Blue area: False Negative(伪反例) rate

False positive(伪正例):某些文档对不是相似的, 但它被认为是相似的
False negative(伪反例):某些文档对是相似的, 但它却认为是不相似的

4.4.3 LSH小结

- 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

Chapter 4 总结

- 1、**Shingling:** Convert documents to sets
  - We used hashing to assign each shingle an ID
- 2、**Min-Hashing:** Convert large sets to short signatures, while preserving similarity
  - We used **similarity preserving hashing** to generate signatures with property $\mathbf{Pr}[h_\pi(C_1) = h_\pi(C_2)] = sim(C_1, C_2)$
  - We used hashing to get around generating random permutations
- 3、**Locality-Sensitive Hashing:** Focus on pairs of signatures likely to be from similar documents
  - We used hashing to find **candidate pairs** of similarity $\geq$ **s**