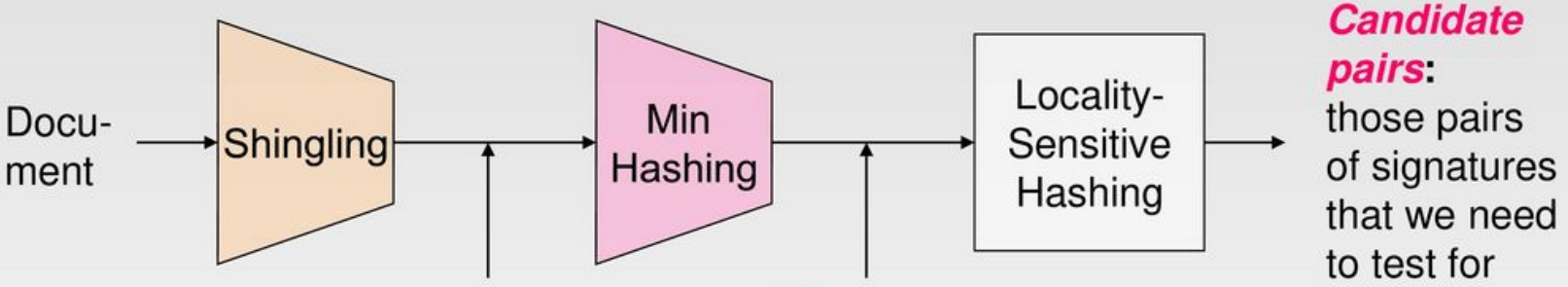
BIG DATA ANALYTICS

Finding Similar Items: Locality-Sensitive Hashing

The Big Picture



Step 3

Locality-Sensitive Hashing: Focus on pairs of signatures likely to be from similar documents

Motivation Locality-Sensitive Hashing

- Find near-duplicate documents among $N = 1$ million documents
- To compute pairwise Jaccard similarities for every pair of docs:
  - $N(N-1)/2 \approx 5 \times 10^{11}$ comparisons
  - At 10^5 secs/day and 10^6 comparisons/sec, it would take 5 days
- For $N = 10$ million, it takes more than a year...

LSH: First Cut

- **Goal**: Find documents with Jaccard similarity at least s
- LSH : Use a function $f(x, y)$ that tells whether x and y is a candidate pair
- For Min-Hash matrices:
  - Hash columns of signature matrix M to many buckets
  - 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:

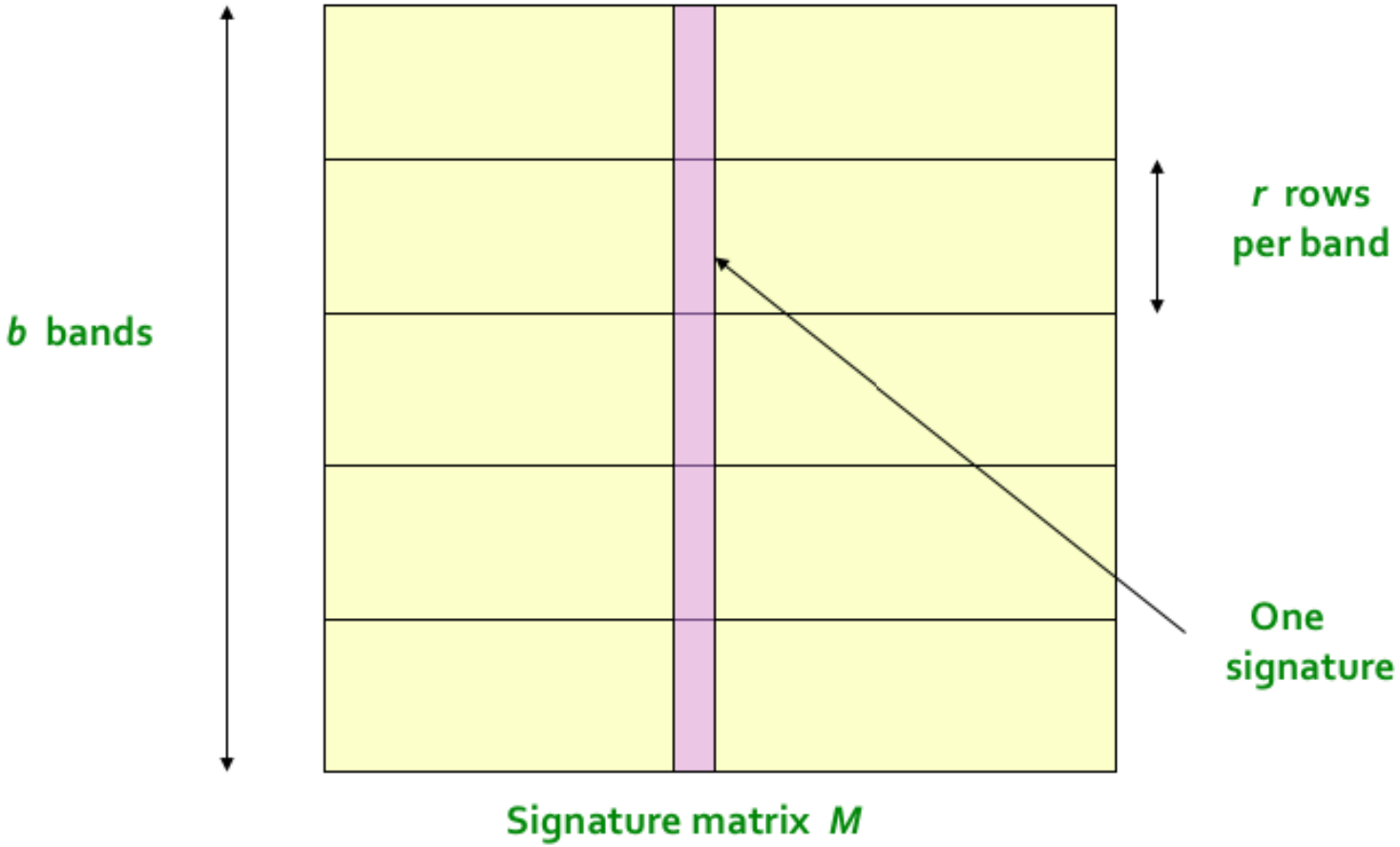
$$M(i, x) = M(i, y) \quad \text{for at least frac. } s \text{ values of } i$$

- We expect documents x and y to have the same (Jaccard) similarity as their signatures
- Check in main memory that candidate pairs really do have similar signatures

LSH from min-hash

- Big idea: Hash columns of signature matrix M several times
- 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
- Check in main memory that candidate pairs really do have similar signatures

Partition of M into b bands



Partition of M into b bands

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

Assumption:

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

LSH involves a trade off

- To balance false positives/negatives pick:
  - The number of Min-Hashes (rows of M)
  - The number of bands b
  - The number of rows r per band

Example: $r = 5$; $b = 20$

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

LSH Summary

- Tune M, b, r:
  - to get almost all pairs with similar signatures
  - eliminate most pairs that do not have similar signatures
- Check in main memory that candidate pairs have similar signatures
- Optional: check that the remaining candidate pairs represent similar documents

Summary: 3 steps

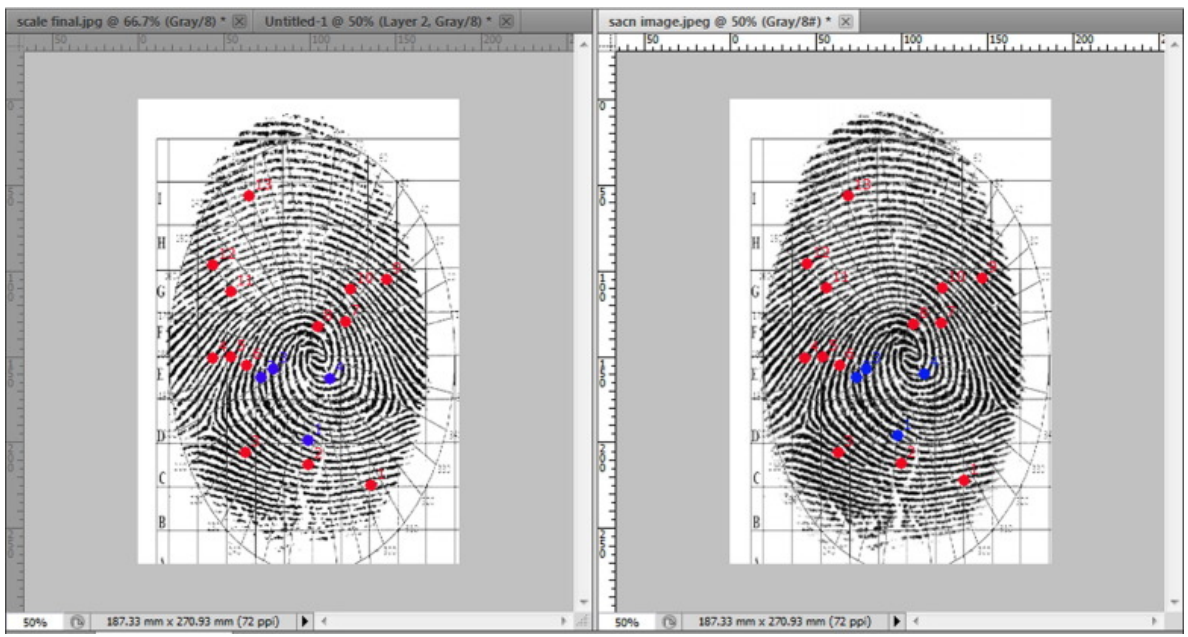
- **Shingling**: Convert documents to sets
  - We used hashing to assign each shingle an ID
- **Min-Hashing**: Convert large sets to short signatures, while preserving similarity
  - We used similarity preserving hashing
  - We used hashing to get around generating random permutations
- **Locality-Sensitive Hashing**: Focus on pairs of signatures likely to be from similar documents
  - We used hashing to find candidate pairs of similarity $\geq s$

Applications of Locality-Sensitive Hashing

- **Matching Fingerprints**
- **Matching Newspaper Articles**

Matching Fingerprints I

- Representation: a set of locations in which *minutiae* are located
- A minutia is a place where something unusual happens: two ridges merging or a ridge ending
- We can represent the fingerprint by the set of grid squares in which minutiae are located:



Matching Fingerprints II

- **The many-one problem:**
  - a fingerprint has been found
  - we want to compare it with all the fingerprints in a large database
- **The many-many version:**
  - see if there are any pairs that represent the same individual

Similar News Articles

- A large repository of on-line news articles that were derived from the same basic text
- Each newspaper surrounds it by information that is special to that newspaper
- $\Longrightarrow$ the same news article can appear quite different at the Web sites of different newspapers

Similar News Articles

- Shingling treats all parts of a document equally
- Ignore ads or the headlines of other articles to which the newspaper added a link
- **Difference between text in prose and text in ads or headlines**
- Prose has a much greater frequency of **stop words**
  - the most frequent words: such as "the" or "and"

Shingles Built from Words

- Define a shingle to be a stop word followed by the next two words
- $\Longrightarrow$ more shingles come from the news article
- Bias the set of shingles in favor of the article

Similar News Articles

- Two Web pages: half news text and half ads
  - The news text is the same but the surrounding material is different:
    - a large fraction of the shingles would be the same
    - a Jaccard similarity of 75%
  - the surrounding material is the same but the news content is different:
    - the number of common shingles would be small
    - a Jaccard similarity of 25%
- Conventional shingling: the two documents share half their shingles (i.e., a Jaccard similarity of 1/3)

References

- J. Leskovec, A. Rajaraman and J. D. Ullman *Mining of Massive Datasets* (2014), Chapter 3

- A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions", Comm. ACM 51:1, pp. 117 - 122, 2008.