Nearest Neighbor and Locality-Sensitive Hashing

- Nearest Neighbor
- Set Similarity
- Locality-Sensitive Hashing
- Document Similarity

Philip Bille

Nearest Neighbor and Locality-Sensitive Hashing

- Nearest Neighbor
- Set Similarity
- Locality-Sensitive Hashing
- Document Similarity

Nearest Neighbor

- Nearest Neighbor.
- Preprocess a collection of high-dimensional vectors **V** = V_1, V_2, **...**, V_n to support
  - NN(S): return all $S_i \in$ **S** such that sim(S, S_i) ≥ threshold t
- Applications.
  - Classification
  - Search
  - Find similar items
  - Recommendation systems
  - ....

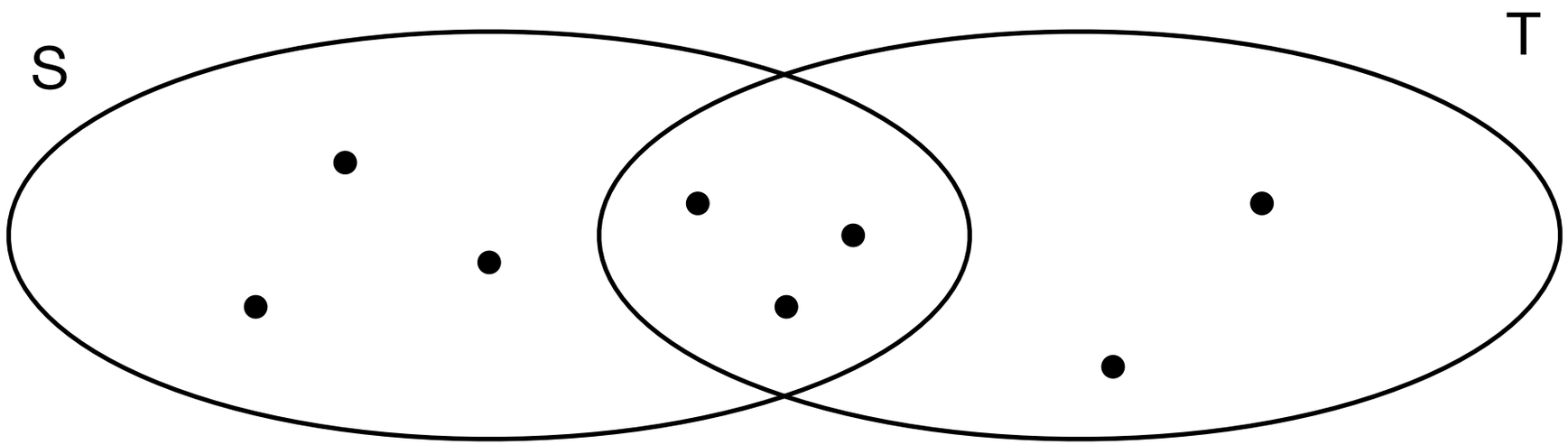
Nearest Neighbor

- Nearest Neighbor (Set version).
- Preprocess a collection of sets **S** = S_1, S_2, **...**, S_n to support
  - NN(S): return all $S_i \in$ **S** such that sim(S, S_i) ≥ t

Nearest Neighbor and Locality-Sensitive Hashing

- Nearest Neighbor
- Set Similarity
- Locality-Sensitive Hashing
- Document Similarity

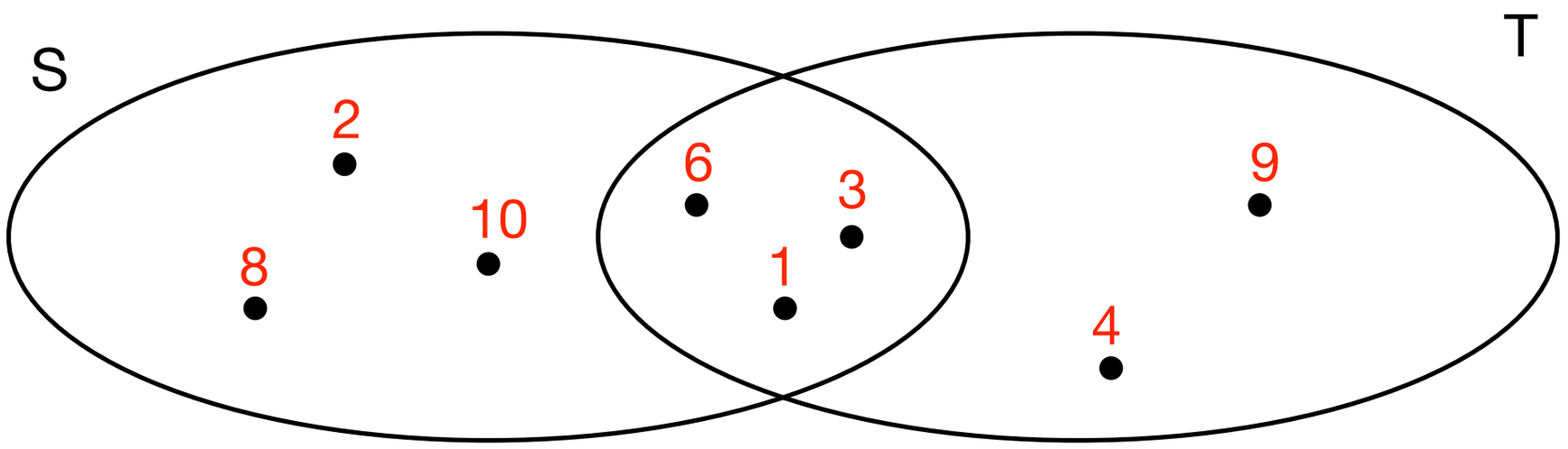
Jaccard Similarity



$$J(S,T) = \frac{|S \cap T|}{|S \cup T|}$$

Minhashing

- Pick a hash function f that maps elements to distinct integers.
- minhash h(S) = min hash on elements in S.



$$\Pr[h(S) = h(T)] = \frac{|S \cap T|}{|S \cup T|} = J(S, T)$$

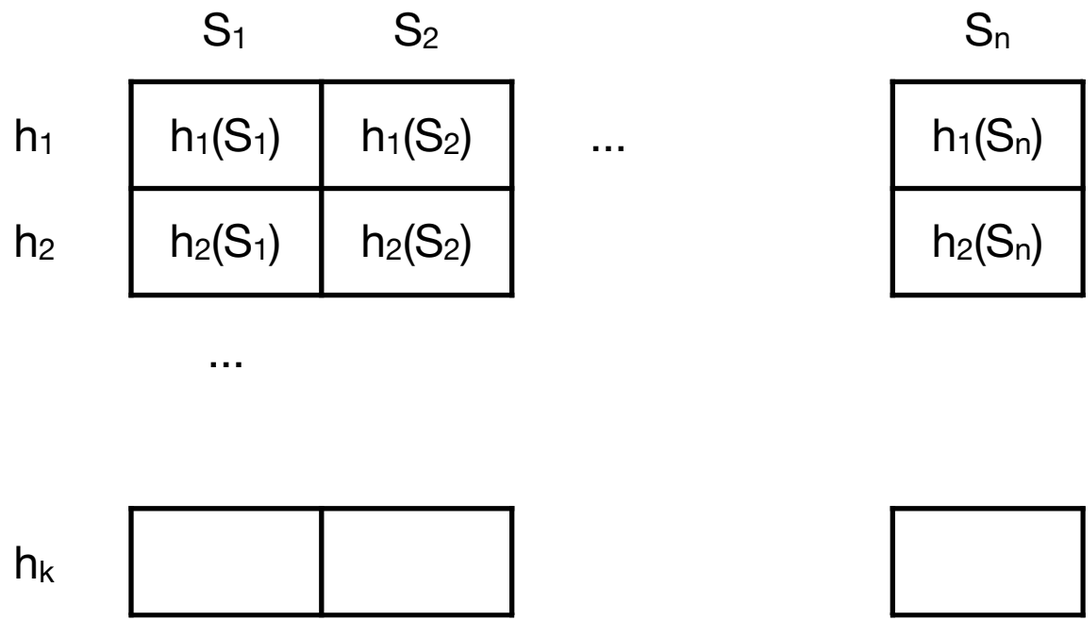
Set Signatures

- Set signature.
  - Pick k hash functions $f_1, f_2, \ldots, f_k$ independently
  - $\Rightarrow$ k minhashes $h_1, h_2, \ldots, h_k$
  - $sig(S) = [h_1(S), h_2(S), \ldots, h_k(S)]$

- Jaccard similarity estimation.
  - $J(S,T) \approx$ (#equal pairs in sig(S) and sig(T)) / k

Nearest Neighbor

- Data structure.
  - Signaturematrix M



- NN(S):
  - Compute sig(S).
  - Compare sig(S) with $sig(S_1)$,...,$sig(S_k)$ using Jaccard estimation. Return all sets with similarity estimation ≥ t.

Nearest Neighbor and Locality-Sensitive Hashing

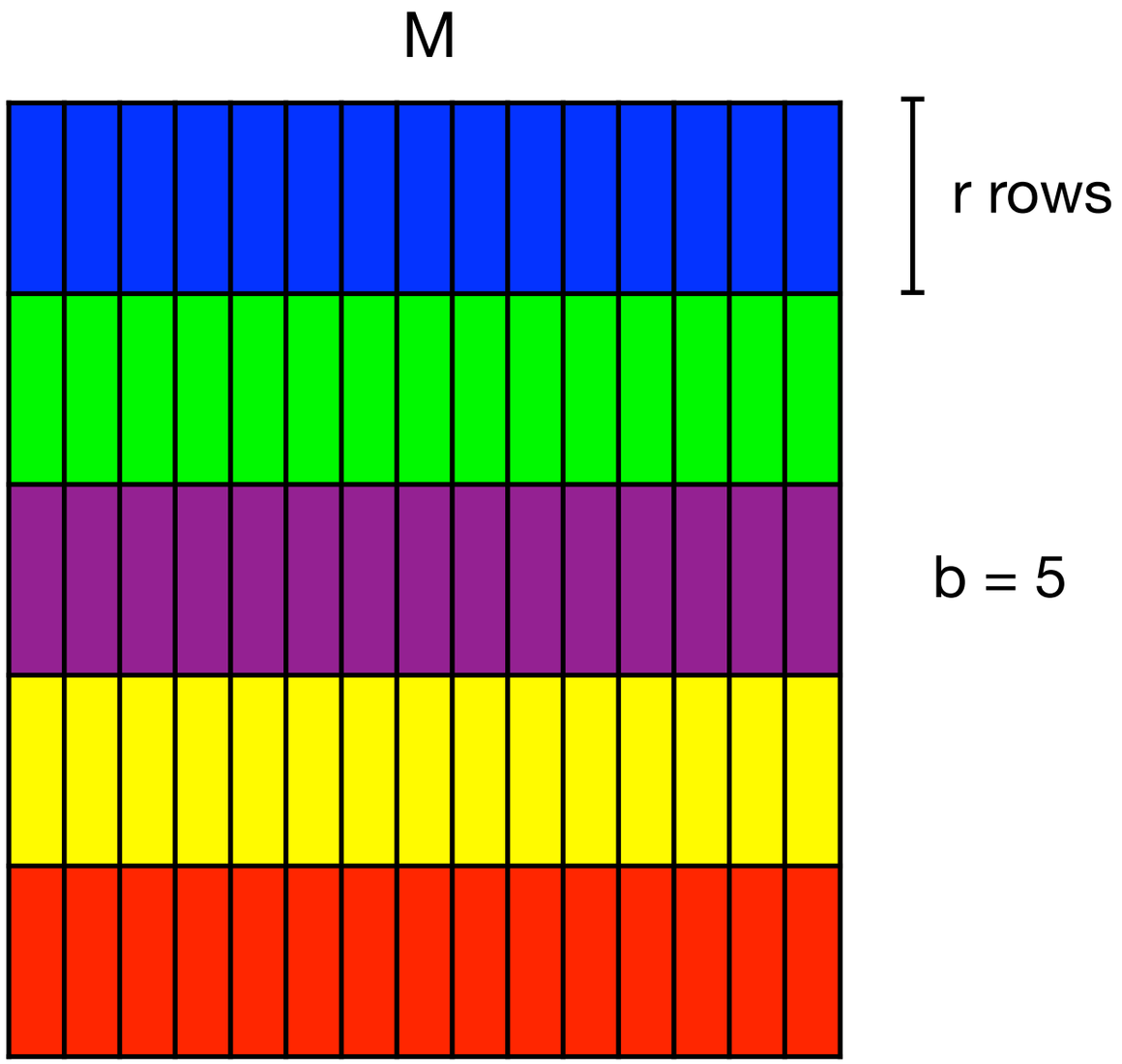
- Nearest Neighbor
- Set Similarity
- Locality-Sensitive Hashing
- Document Similarity

Locality-Sensitive Hashing

- Idea.
  - Filter all but a few candidates.
  - Check candidates using set signature similarity estimation.
    - (Optionally compute exact Jaccard similarity for candidates).

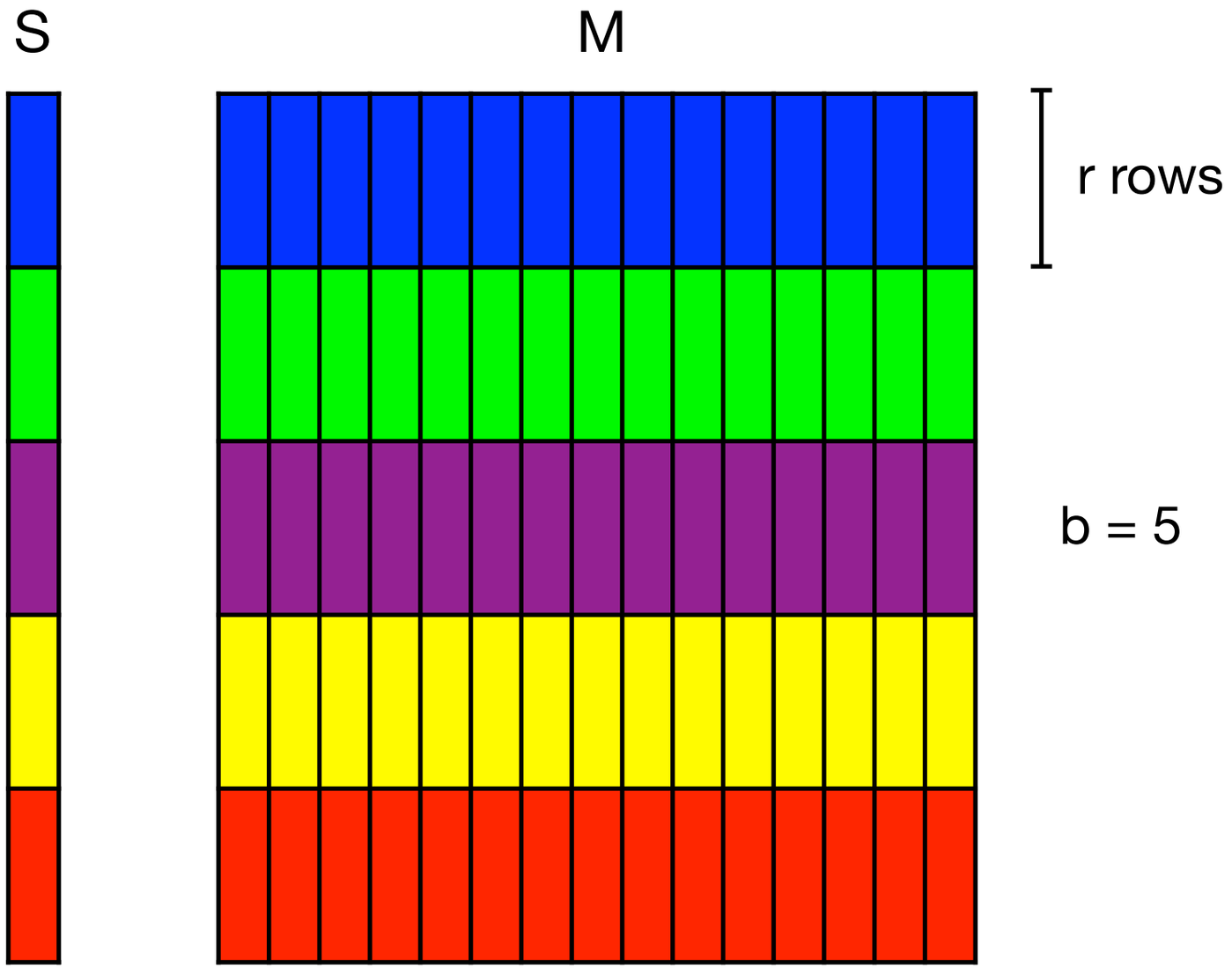
- Goal.
  - Balance false positives and false negatives
    - false positives = sets with similarity < t that become candidates
    - false negatives = sets with similarity > t that do not become candidates.

Locality-Sensitive Hashing



- Banding.
  - Partition signature matrix M into b bands of r rows.
  - Store a dictionary for each band.

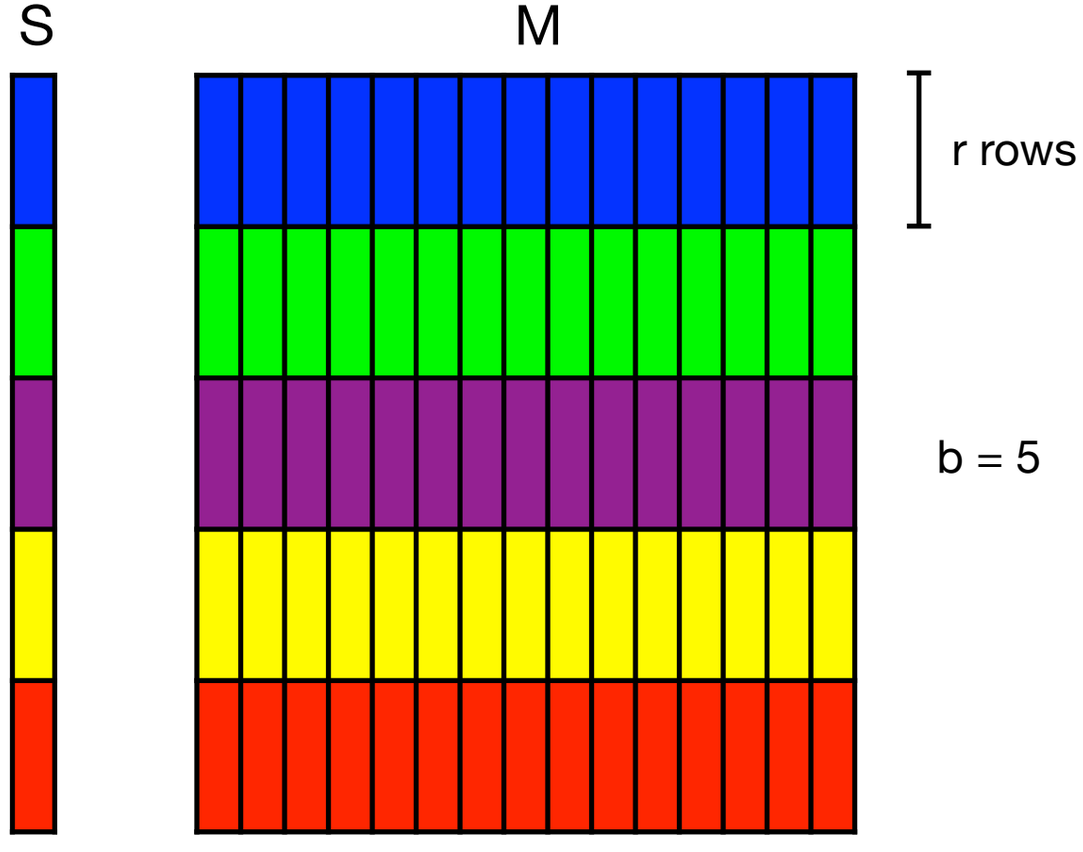
Locality-Sensitive Hashing



- NN(S):
  - Construct sig(S)
  - Partition sig(S) into bands and lookup in corresponding dictionary.
  - Make S_i a candidate if it matches on some band with S.

Locality-Sensitive Hashing

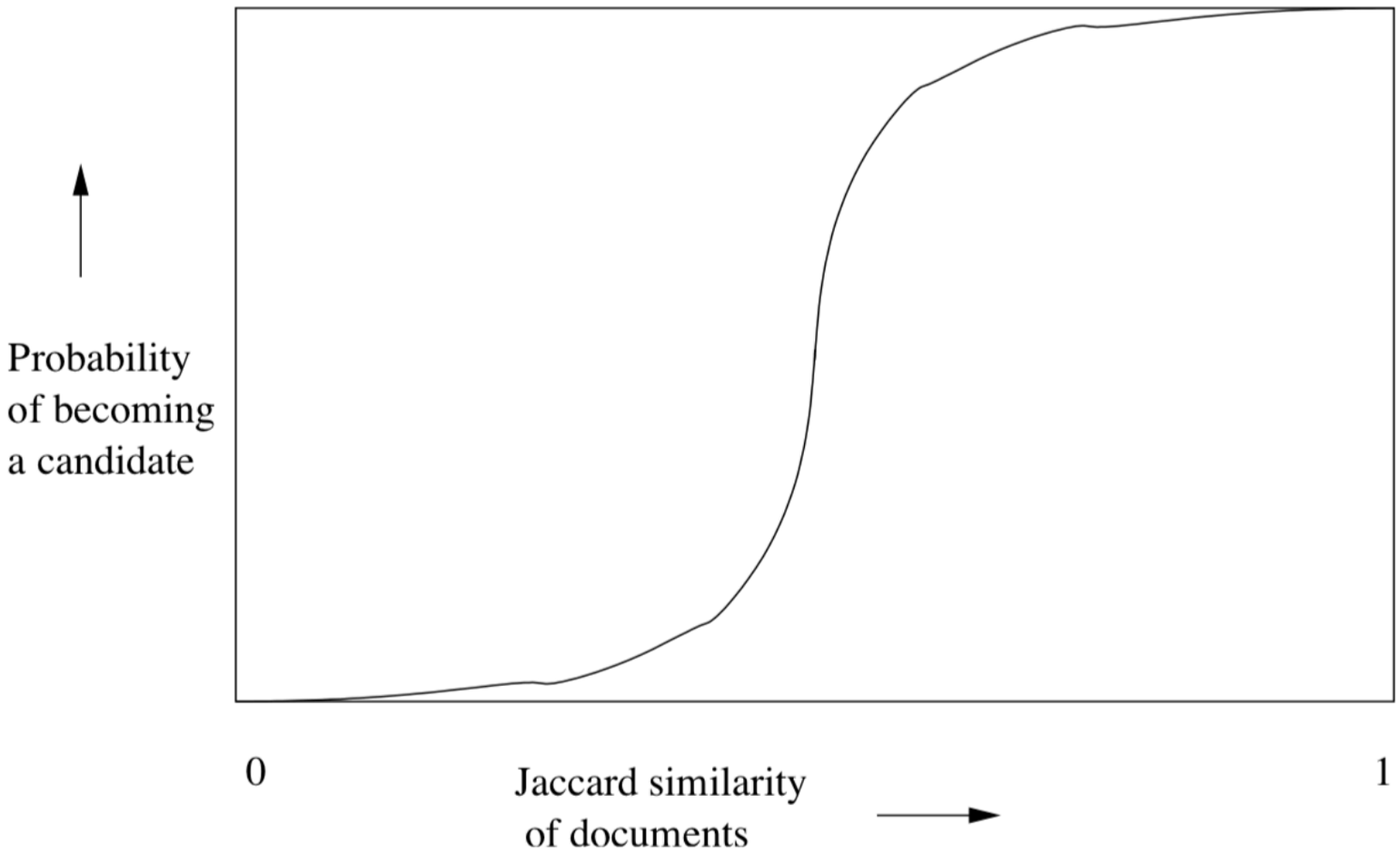
- Analysis of banding. Suppose S and S_i have similarity s. What is probability that S_i becomes a candidate?
  - Probability identical on 1 row = s
  - Probability identical on 1 band = s^r
  - Probability at least 1 row in a band is not identical = $1 - s^r$
  - Probability no band is identical = $(1-s^r)^b$
  - Probability at least 1 band is identical = $1 - (1-s^r)^b$



Locality-Sensitive Hashing

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

b = 20, r = 5, n = br = 100



- Choosing b and r.
  - Threshold: similarity where probability of becoming a candidate is > 1/2
  - Threshold ≈ $(1/b)^{1/r}$

Nearest Neighbor and Locality-Sensitive Hashing

- Nearest Neighbor
- Set Similarity
- Locality-Sensitive Hashing
- Document Similarity

Documents as Sets

- Shingles.
  - "I used to think I was indecisive, but now I'm not too sure."
  - ["I", "used", "to"], ["used", "to", "think"], ["think", "I", "was"]

- Document = set of shingles.