Data Mining and Analysis

Finding similar items

CSCE 676 :: Fall 2019
Texas A&M University
Department of Computer Science & Engineering
Prof. James Caverlee

Resources

MMDS Chapter 3 + slides

- http://i.stanford.edu/~ullman/mmds/ch3n.pdf
- http://www.mmds.org/mmds/v2.1/ch03-lsh.pdf

Carlos Castillo course on Data Mining [https://github.com/chatox/data-mining-course]

Example

- **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

Example: Suppose sim(C_1,C_2) = 0.8

- **Find pairs of** ≥ ***s***=0.8 similarity, set **b**=20, **r**=5
- 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** (we miss them)
  - **We would find 99.965% pairs of truly similar documents**

Example: Suppose sim(C_1,C_2) = 0.3

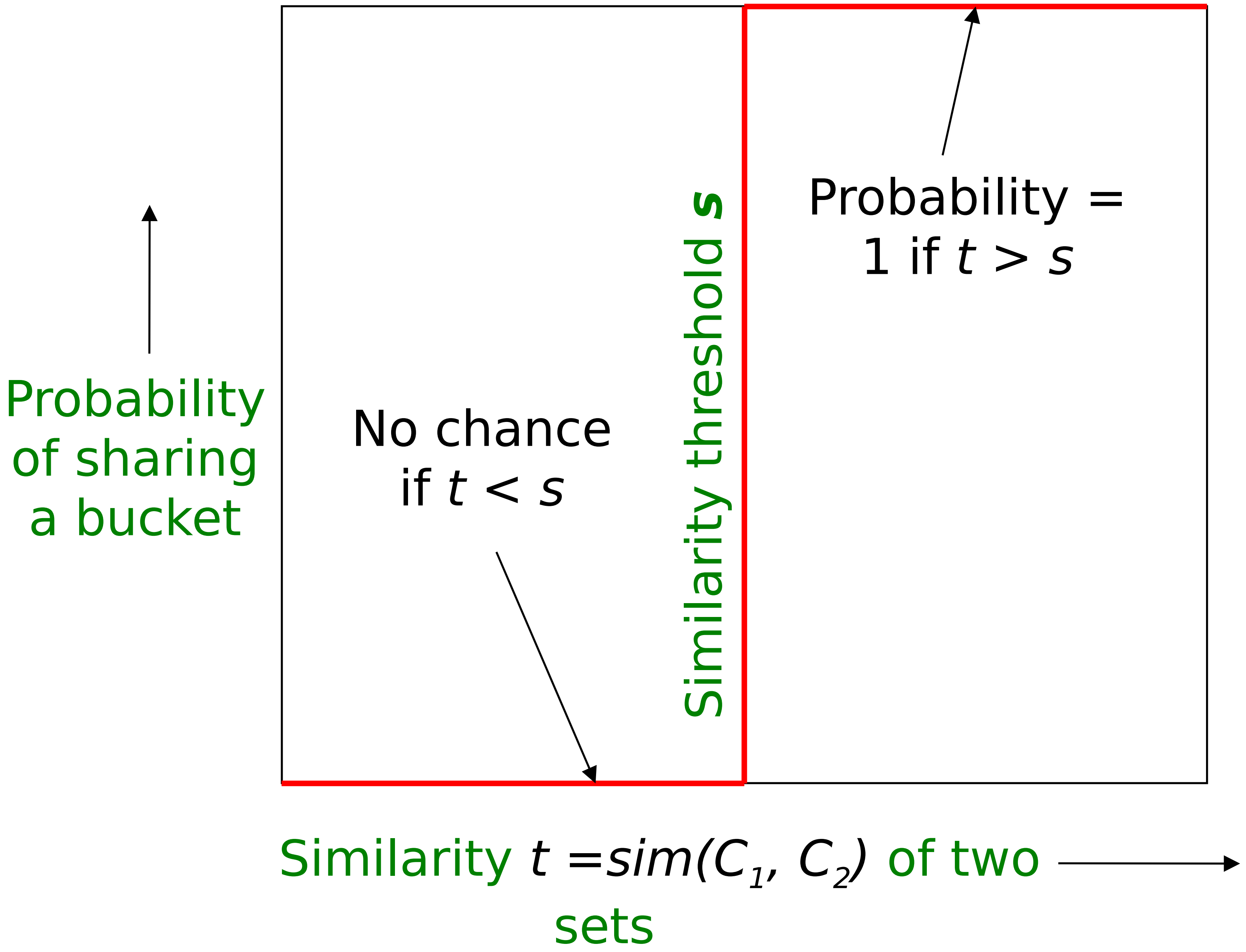
- **Find pairs of ≥ *s***=0.8 similarity, set **b**=20, **r**=5
- Since sim(C_1, C_2) < s we want C_1, C_2 to hash to NO common buckets (all bands should be different)
- Probability C_1, C_2 identical in one particular band: $(0.3)^5 = 0.00243$
- Probability C_1, C_2 identical in at least 1 of 20 bands: $1 - (1 - 0.00243)^{20} = 0.0474$
- In other words, approximately 4.74% pairs of docs with similarity 0.3% end up becoming **candidate pairs**
- They are **false positives** 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 trade-off

- **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/negatives

- **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

LSH: what we want



Probability of sharing a bucket

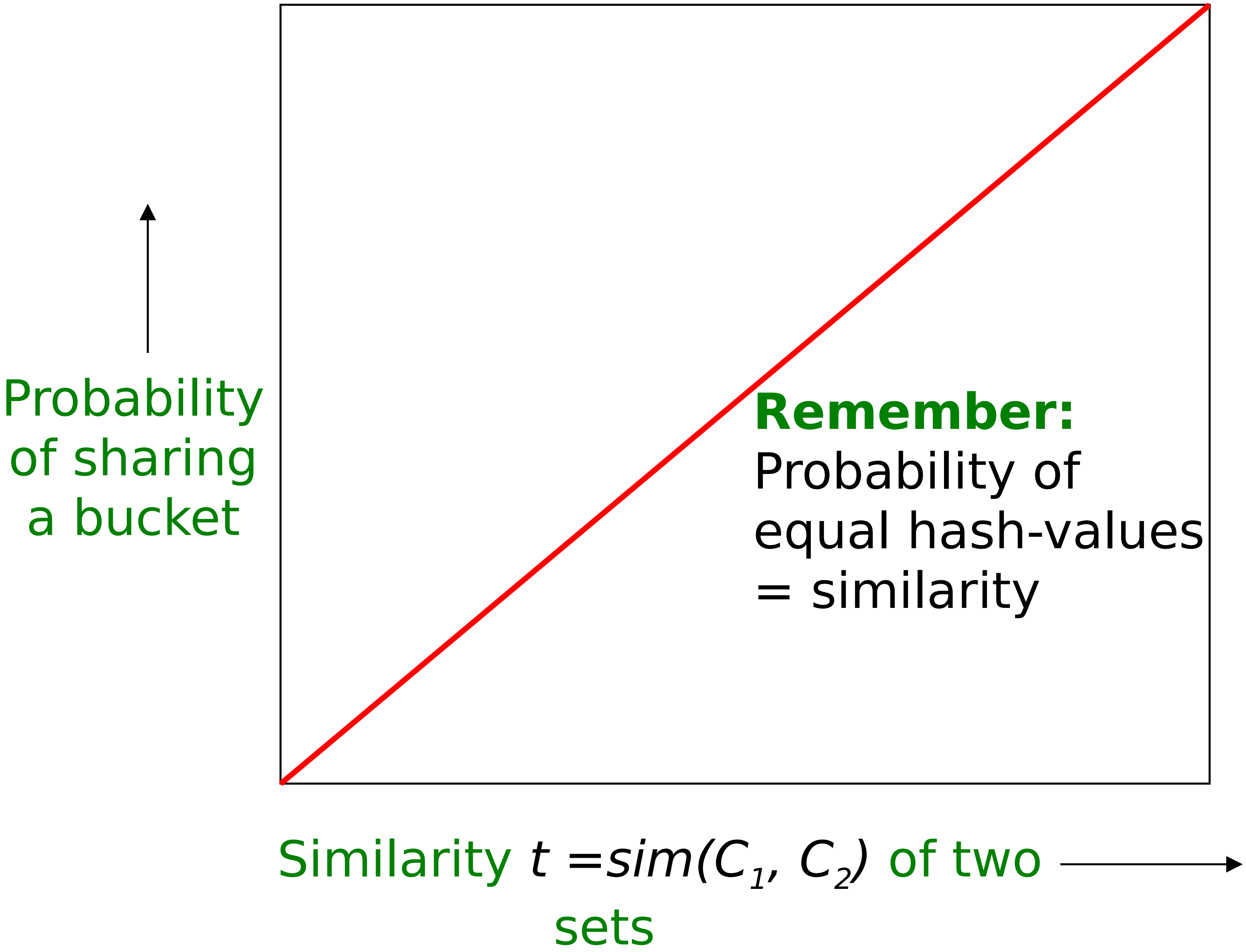
Similarity threshold s

Probability = 1 if $t > s$

No chance if $t < s$

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

What 1 band of 1 row gives you



Probability of sharing a bucket

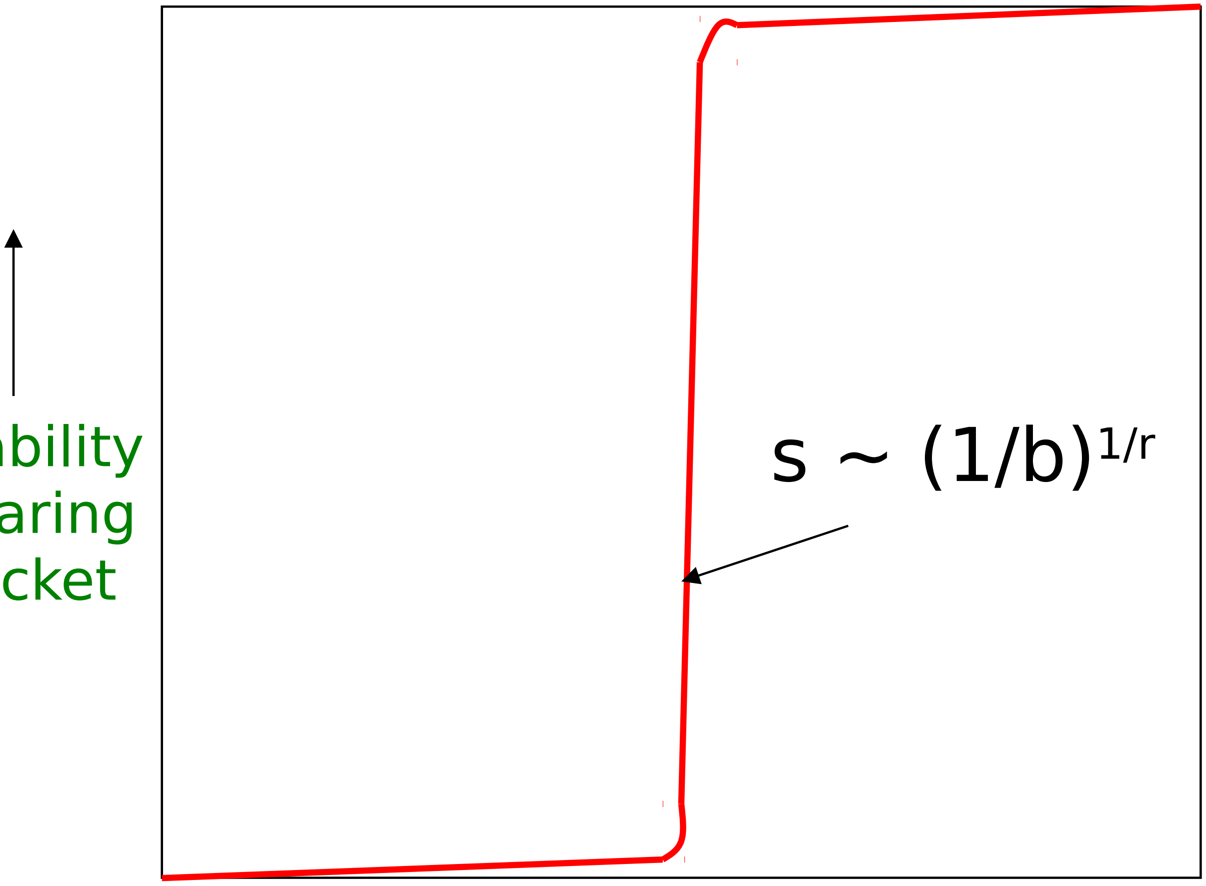
Remember: Probability of equal hash-values = similarity

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

b bands, r rows/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 =
  - $1 - (1 - t^r)^b$

What b bands of r rows gives you



$s \sim (1/b)^{1/r}$

Probability of sharing a bucket

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

At least one band identical

No bands identical

$$\mathbf{1 - (1 - \mathit{t}^{\,r})^{b}}$$

Some row of a band unequal

All rows of a band are equal

Example: b=20, r=5

Similarity threshold s

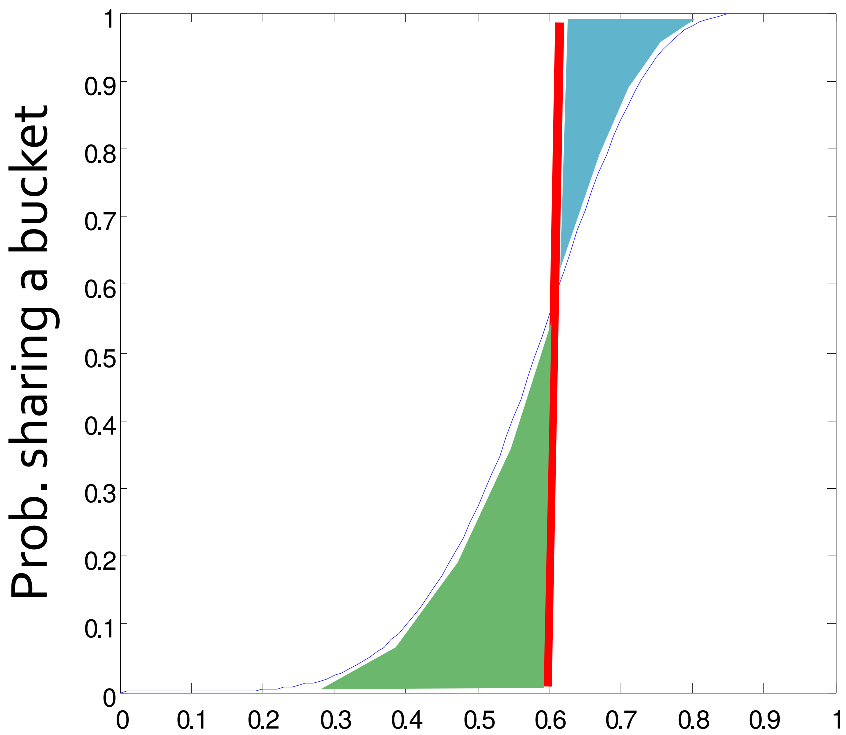
Probability that at least 1 band is identical:

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)

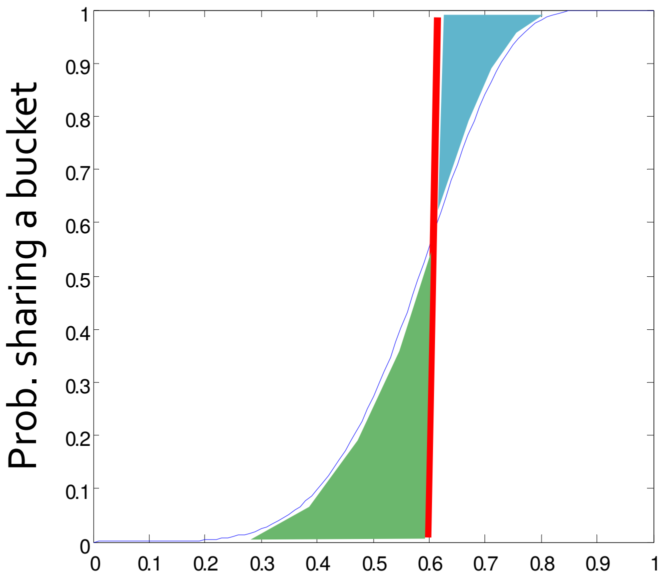


Blue area: False Negative rate

Green area: False Positive rate

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

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

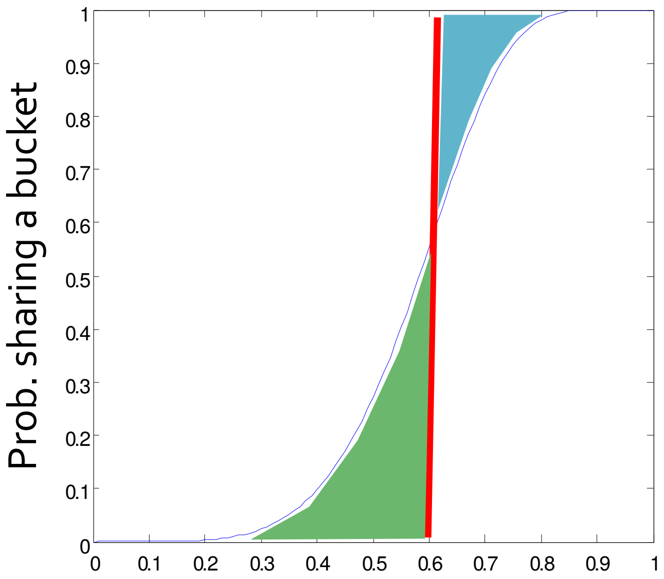


Blue area: False Negative rate
Green area: False Positive rate

Blue area X: False Negative rate These are pairs with sim > s but the X fraction won’t share a band and then will never become candidates. This means we will never consider these pairs for (slow/ exact) similarity calculation!

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

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

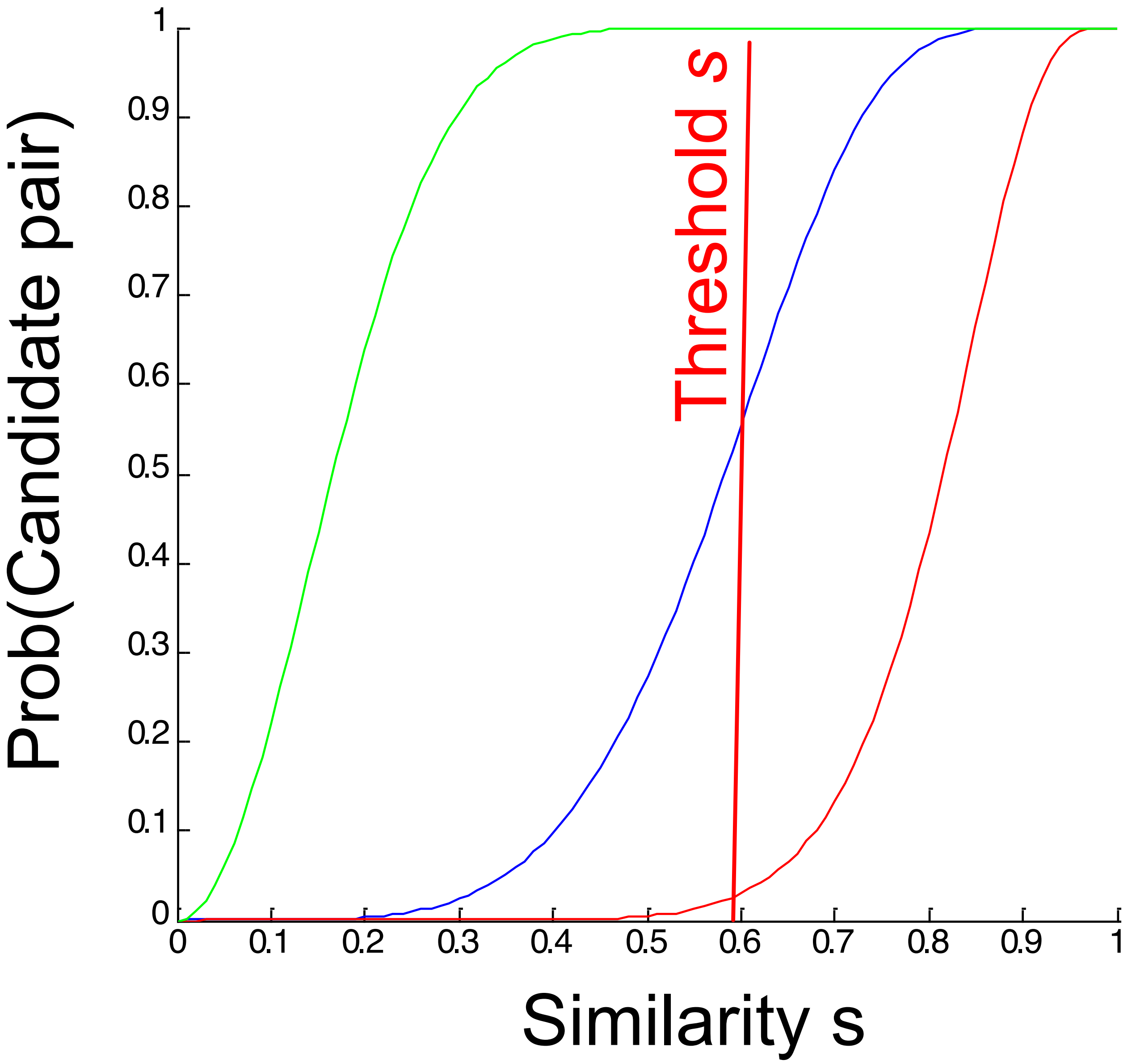


Blue area: False Negative rate
Green area: False Positive rate

Green area Y: False positive rate.

These are pairs with sim < s but we will consider them as candidates. This is not too bad, we will consider them for (slow/exact) similarity computation and discard them

Suppose we have 50 hash functions (r * b = 50)



r=2, b=25

r=5, b=10

r=10, b=5

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