SDSC3002
Finding Similar Items: Locality Sensitive Hashing

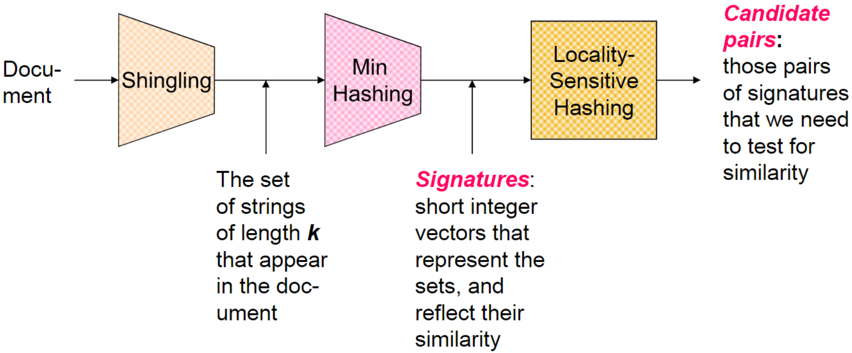
Yu Yang
yuyang@cityu.edu.hk

Recap: Min Hashing

- Document $\rightarrow$ Set of Shingles (by Shingling) $\rightarrow$ Signatures (via Min Hashing)
- To boost the estimation accuracy, we adopt r independent random permutations (hash functions)
  - The i-th random permutation/hashing results in $m_i(C)$ for each column C
  - $\hat{JS}(S_1, S_2) = \frac{1}{r} \sum_{i=1}^{r} \mathbf{I}(m_i(S_1) = m_i(S_2))$

But we still have not addressed the issue of $O(n^2)$ comparisons

The Big Picture



General Idea

- Goal: find docs with Jaccard similarity at least s
- General idea: process each doc and put them into some "buckets"
  - Matrix $\mathbf{M}$: each column represents the signatures of a doc
- Only pairs of docs in the same bucket are considered as candidate pairs

Outline

LSH for Min-Hash

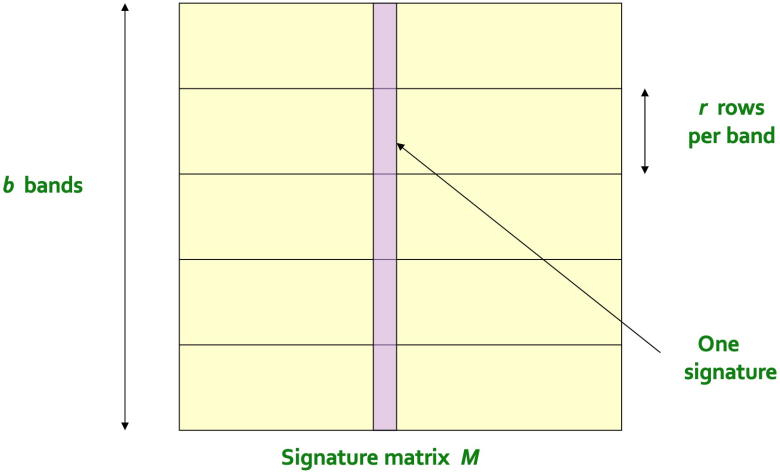
LSH Theory

LSH for Distance Measures

LSH for Min-Hash

- One single idea: (C_1, C_2) is a candidate pair only if their signatures are exactly the same
  - Scan the signature matrix $\mathbf{M}$ once, put the same columns together (how to implement this?)
- Maybe good for when the threshold s is close to 1
- A second thought: (C_1, C_2) is a candidate pair if they share many signatures
  - How to implement this?

Banding



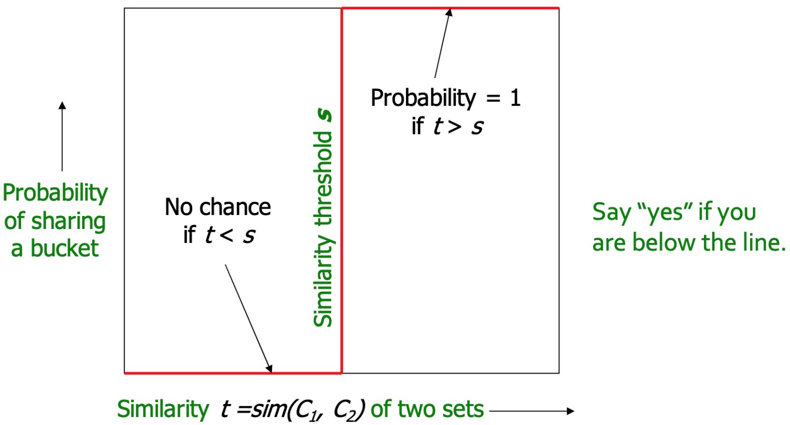
Partition M into Bands

- Divide **M** into b bands of r rows
- For each band, hash its portion of each sub-column into buckets (ideally, without collision, which is doable)
- Candidate column pairs are those having the same bucket for ≥ 1 band
- Tune b and r to make probabilities of similar pairs being in same buckets high, and probabilities of dissimilar pairs being in same buckets low

An Example

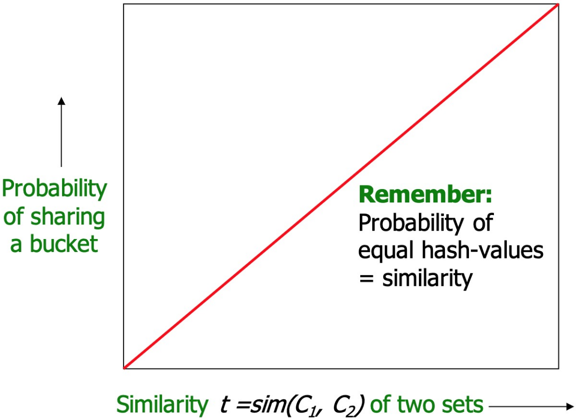
- Assume 100,000 docs, 100 signatures for each doc (**M** contains 100K columns and 100 rows)
- $b = 20$, $r = 5$, similarity threshold $s = 0.8$
- If $sim(C_1, C_2) \geq 0.8$
  - Prob. of sharing a bucket for one band: $\geq s^r = 0.8^5 = 0.328$
  - Prob. of being a candidate: $\geq 1 - (1 - s^r)^b = 1 - 0.00035$
- If $sim(C_1, C_2) \leq 0.3$
  - Prob. of sharing a bucket for one band: $\leq 0.3^r = 0.00243$
  - Prob. of being a candidate: $\leq 1 - (1 - 0.3^r)^b = 0.0474$
- Picking b and r is important

Ideal Case

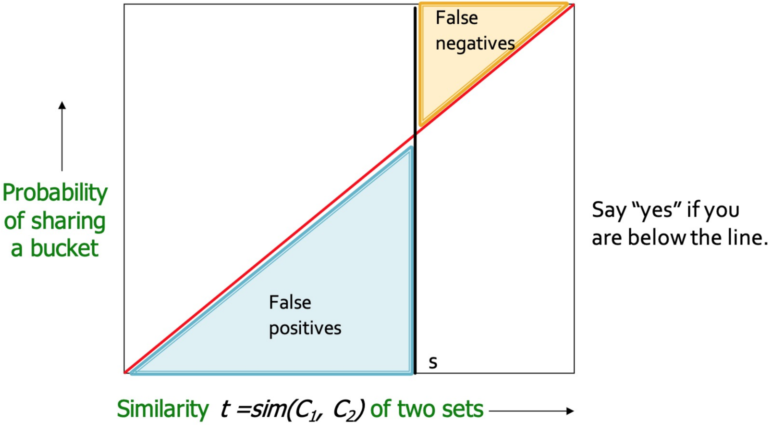


Probability of sharing a bucket
No chance if $t < s$
Similarity threshold s
Probability = 1 if $t > s$
Say "yes" if you are below the line.
Similarity $t = sim(C_1, C_2)$ of two sets

$b = r = 1$


Probability
of sharing
a bucket
Remember:
Probability of
equal hash-values
= similarity
Similarity $t = sim(C_1, C_2)$ of two sets

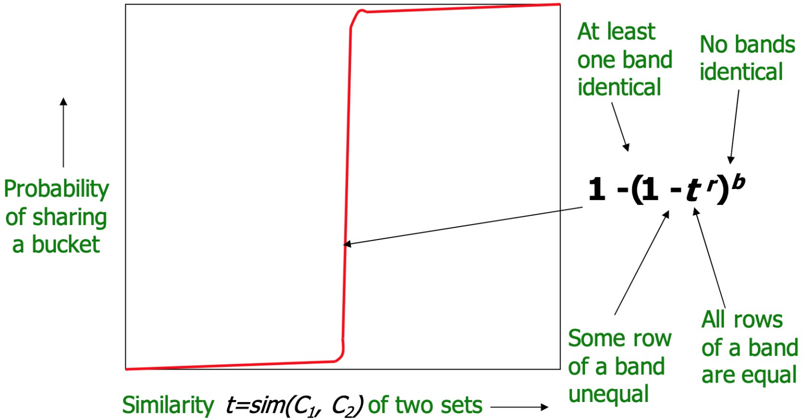
$b = r = 1$


Probability
of sharing
a bucket
False
negatives
False
positives
s
Similarity t =sim(C1, C2) of two sets
Say "yes" if you
are below the line.

b bands and r rows

- Suppose $sim(C_1, C_2) = t$
- Prob. of sharing a bucket for a band: t^r
- Prob. of being a candidate: $1 - (1 - t^r)^b$
  - Prob. of not being a candidate: $(1 - t^r)^b$

b bands and r rows



$b = 20$ and $r = 5$

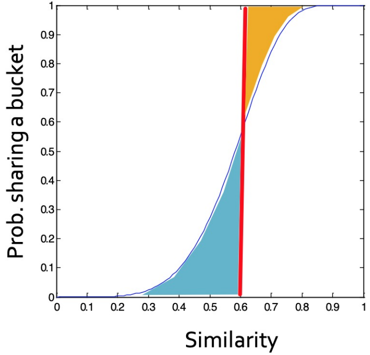
Prob. that at least 1 band is identical:

s	$1-(1-s^r)^b$
0.2	0.006
0.3	0.047
0.4	0.186
0.5	0.470
0.6	0.802
0.7	0.975
0.8	0.9996

The S-curve

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

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



Yellow area: False Negative rate
Blue area: False Positive rate

LSH Summary

- Pick b (#bands) and r (#rows/band)
- Hash bands of each doc (set of shingles) into buckets (without collision)
- (C_1, C_2) is a candidate pair if they ever share a bucket
- Check similarity (estimation by signatures) of candidate pairs of docs
- Return similar doc pairs

Outline

LSH for Min-Hash

LSH Theory

LSH for Distance Measures

Distance Measures

- LSH can be generalized to other distance measures
- $d(\cdot)$ a **distance measure** if it is a function from pairs of points x, y to real numbers such that
  - $d(x, y) = d(y, x) \geq 0$, $d(x, y) = 0 \leftrightarrow x = y$
  - $d(x, z) \leq d(x, y) + d(y, z)$ (triangle inequality)
- Examples of distance measures
  - Euclidean distance: $\|x - y\|$
  - Jaccard distance: $1 - JS(S_1, S_2)$
  - Cosine distance for vectors: angle between two vectors

Families of Hash Functions

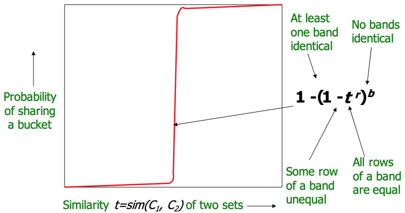
- Usage of hash functions in LSH: decide if C_1 and C_2 are equal (if $h(C_1) = h(C_2)$)
- A **family** of hash functions is a set of hash functions that we can **efficiently and randomly** pick one
  - Example: universal hashing, with a large prime M, $h_{cd}(a) = ca + d (\text{mod } M)$, $H = \{h_{cd} \mid c, d = 0, 1, \ldots, M-1\}$ is a family of hash functions

Locality-Sensitive Families

- Suppose we have a distance measure $d(x, y)$
- A family H of hash functions is (d_1, d_2, p_1, p_2)-sensitive if we randomly pick $h \in H$ such that for any x, y
  1. If $d(x, y) \leq d_1$, then $h(x) = h(y)$ with prob. at least p_1
  2. If $d(x, y) \geq d_2$, then $h(x) = h(y)$ with prob. at most p_2
- Min-Hash as an example
  - $d(x, y) = 1 - JS(x, y)$
  - $Pr\{h(x) = h(y)\} = 1 - d(x, y)$, $h(\cdot)$ is a random permutation
  - The set of all random permutations H is a $(d_1, d_2, (1 - d_1), (1 - d_2))$-sensitive LS family
- What we want: small $d_2 - d_1$, large $p_1 - p_2$
- We can amplify an LS family!

Amplifying an LS Family

- We want the S-curve
- Banding: amplify a given (d_1, d_2, p_1, p_2)-sensitive LS family
- Two constructions
  - **AND**: combine rows in a band
  - **OR**: combine bands



AND of Hash Functions

- Given H, construct H' where a hash function h consists of r random hash functions $[h_1, \dots, h_r]$ from H
- $h(x) = h(y)$ if $h_i(x) = h_i(y)$ for $i = 1, \dots, r$
  - Exactly what a band of r rows does

Theorem: If ***H*** is ***(d_1, d_2, p_1, p_2)***-sensitive,
then **H'** is ***(d_1,d_2, $(p_1)^r$, $(p_2)^r$)***-sensitive
Proof: Use the fact that ***h_i***'s are **independent**

Also lowers probability for small distances (Bad)

Lowers probability for large distances (Good)

OR of Hash Functions

- Given H, construct H' where a hash function h consists of b random hash functions $[h_1, \ldots, h_b]$ from H
- $h(x) = h(y)$ if $h_i(x) = h_i(y)$ for at least one $i \in [b]$
  - Similar to combining b bands

Theorem: If $\boldsymbol{H}$ is $(\boldsymbol{d_1}, \boldsymbol{d_2}, \boldsymbol{p_1}, \boldsymbol{p_2})$-sensitive, then $\boldsymbol{H'}$ is $(\boldsymbol{d_1}, \boldsymbol{d_2}, \mathbf{1-(1-}\boldsymbol{p_1}\mathbf{)^b}, \mathbf{1-(1-}\boldsymbol{p_2}\mathbf{)^b})$-sensitive

Proof: Use the fact that $\boldsymbol{h_i}$'s are **independent**

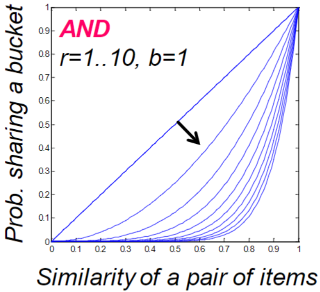
Raises probability for small distances (Good)

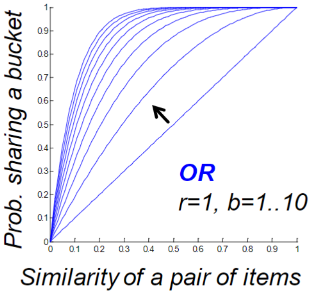
Raises probability for large distances (Bad)

Effect of AND and OR

AND makes all probs. **shrink**, but by choosing ***r*** correctly, we can make the lower prob. approach 0 while the higher does not

OR makes all probs. **grow**, but by choosing ***b*** correctly, we can make the higher prob. approach 1 while the lower does not

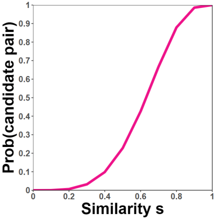




Combine AND and OR

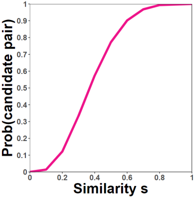
- Combine AND and OR to make p_1 bigger and p_2 smaller
  - AND-OR: r-way AND followed by b-way OR, LSH for Min-Hash, $1-(1-s^r)^b$
  - OR-AND: b-way OR followed by r-way AND, $(1-(1-s)^b)^r$
  - Or any sequence of AND's and OR's alternating

s	p=1-(1-s⁴)⁴
.2	**.0064**
.3	.0320
.4	.0985
.5	.2275
.6	.4260
.7	.6666
.8	**.8785**
.9	.9860



r = 4, b = 4 transforms a (.2,.8,.8,.2)-sensitive family into a (.2,.8,.8785,.0064)-sensitive family

s	p=(1-(1-s)⁴)⁴
.1	.0140
.2	**.1215**
.3	.3334
.4	.5740
.5	.7725
.6	.9015
.7	.9680
.8	**.9936**



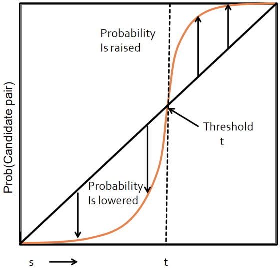
The example transforms a (.2,.8,.8,.2)-sensitive family into a (.2,.8,.9936,.1215)-sensitive family

Cascading Constructions

- Example: apply $(4,4)$ OR-AND followed by $(4,4)$ AND-OR
  - $\left(1-\left(1-\left(1-(1-s)^4\right)^4\right)^4\right)^4$
- Boost a $(0.2, 0.8, 0.8, 0.2)$-sensitive family to a $(0.2, 0.8, 0.9999996, 0.0008715)$-sensitive family
- $4 * 4 * 4 * 4 = 256$ hash functions are used

General Use of S-Curves

- For an AND-OR S-curve $(1-(1-s)^r)^b$, there is a threshold t such that $(1-(1-t)^r)^b = t$ (OR-AND is similar)
  - Raise probabilities above t and lower probabilities below t
- To amplify (d_1, d_2, p_1, p_2)-sensitive family to be (d_1, d_2, p_1', p_2')-sensitive, the closer p_1' to 1 and p_2' to 0, the more hash functions are needed



Outline

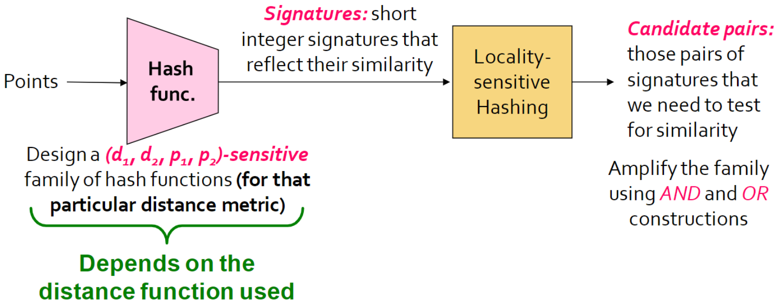
LSH for Min-Hash

LSH Theory

LSH for Distance Measures

The Big Picture

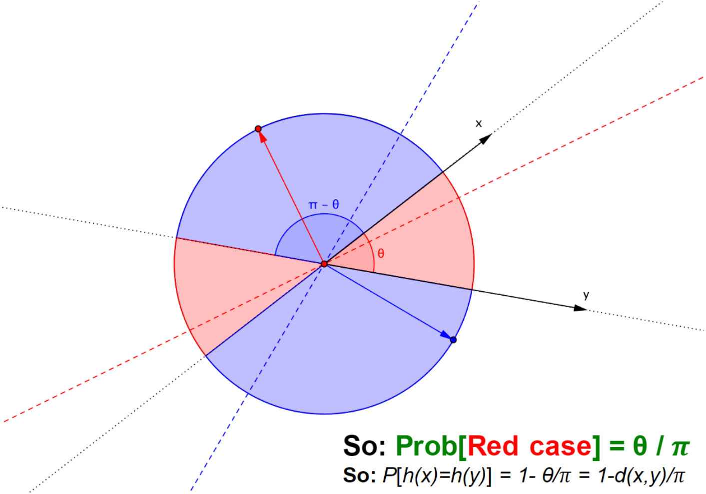
As long as $\Pr\{h(x) = h(y)\} = sim(x, y)$



LSH for Cosine Distance

- Cosine distance: angle between two vectors
  - $d(A, B) = \theta = \arccos(\frac{A \cdot B}{\|A\| \|B\|}) \in [0, \pi]$
  - Cosine similarity: $1 - d(A, B)/\pi$, or $\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$
- Use **random hyperplanes** to hash
  - A random vector v determines a hash function h_v of two buckets
  - $h_v(x) = 1$ if $v \cdot x \geq 0$, $h_v(x) = -1$ if $v \cdot x < 0$
  - $(d_1, d_2, (1 - d_1/\pi), (1 - d_2/\pi))$-sensitive if one random hyperplane is used

Random Hyperplane Partition



So: Prob[Red case] = θ / π

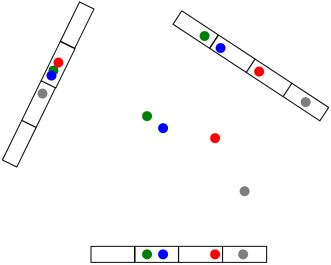
So: $P[h(x)=h(y)] = 1- \theta/\pi = 1-d(x,y)/\pi$

Signatures for Cosine Distance

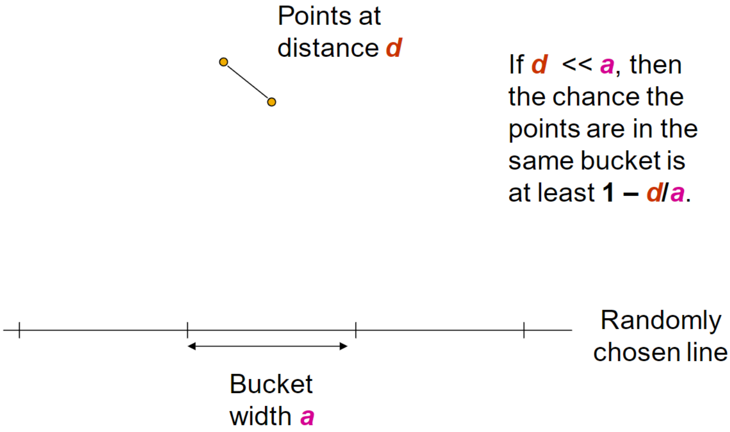
- Pick some random vectors as hyperplanes
  - Implementation: randomly set v_i as 1 or -1 to construct $v = (v_1, \ldots, v_d)$
- Signatures of a data point is a vector of $+1$'s and -1's
- Amplify using AND/OR constructions

LSH for Euclidean Distance

- Hash functions correspond to random lines
- Partition each random line into buckets of width a
- Project each data point onto a line to find the corresponding bucket



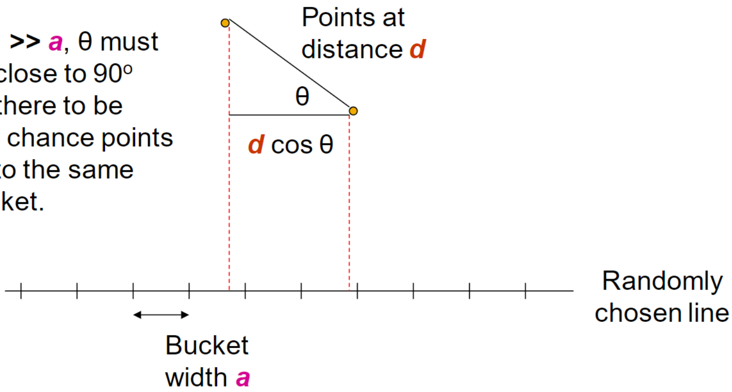
Projection of Points



If *d* << *a*, then the chance the points are in the same bucket is at least **1 – *d*/*a***.

Projection of Points

If ***d*** **>>** ***a***, θ must be close to 90º for there to be any chance points go to the same bucket.



Sensitivity Analysis

- If $d(x,y) \leq a/2$, with prob. at least $1 - d(x,y)/a \geq 1/2$ their projections are within the same bucket
- If $d(x,y) \geq 2a$, the projections are within the same bucket only if $d(x,y)\cos\theta \leq a$
  - $\cos\theta \leq 1/2 \rightarrow 60^\circ \leq \theta \leq 90^\circ \rightarrow$ prob. at most $1/3$
- $(a/2, 2a, 1/2, 1/3)$-sensitive family

Readings

- Chapter 3 of the required book

Acknowledgement

- Some of the contents originate from Jure Leskovec's slides for CS246 at Stanford