Locality Sensitive Hashing

CompSci 590.04
Instructor: Ashwin Machanavajjhala

Problem: Finding Duplicate Elements

- Given a set of objects
- Find objects that are near duplicates of each other.

More formally,

- Let d(x,y) be a distance function defined over pair of objects.
- Group objects such that:

  objects within distance *d1* are both present in some group
  objects at distance > *d2* are never within the same group

Motivation: Entity Resolution

Problem of identifying and linking/grouping different manifestations of the same real world object.

Examples of manifestations and objects:

- Different ways of addressing (names, email addresses, FaceBook accounts) the same person in text.
- Web pages with differing descriptions of the same business.
- Different photos of the same object.
- …

Motivation: Document Clustering

Belinelli's late jumper gives **Popovich** his **1000**th career w…
Yahoo Sports (blog) - 4 hours ago
Gregg **Popovich** of the San Antonio Spurs has already established himself ... Nevertheless, it's pretty cool and rare any time a coach hits **1,000** ...

Spurs' Gregg **Popovich** becomes 9th NBA coach to win **1000** games
SI.com - 20 hours ago

SVG: **Popovich's 1000** wins 'a great accomplishment'
Detroit Free Press - 2 minutes ago

Gregg **Popovich** Wins **1000**th Game with Milestones Ahead & Other **...**
In-Depth - Bleacher Report - 18 hours ago

Six things to know about Gregg **Popovich's 1000**th win
Blog - Washington Post (blog) - 20 hours ago

Raptors Beat Spurs, Deny **Popovich 1000**th Win
In-Depth - ABC News - Feb 8, 2015

Distance Functions

- Jaccard Similarity
  - If each object x is a subset F_x from some universe (e.g., a document is a set of words)
  - Similarity between x and y is: $\frac{F_x \cap F_y}{F_x \cup F_y}$

- Hamming Distance
  - If each object x is in {0,1}^n (e.g., If n is the number of words in the vocabulary and a 0 or 1 in position i signifies whether or not ith word in the vocabulary appears in the document)
  - Similarity between x and y is: number of positions that x and y differ in

Duke UNIVERSITY

Distance Functions

- Cosine Similarity
  - Suppose each x is n dimensional vector of real numbers (e.g., the ith count represents the number of times the ith word in the vocabulary appears in a document)
  - Similarity between w = [w1, w2, …, wn] and y = [y1, y2, .., yn] is given by

Dot Product

$$d(D1, D2) = \frac{\sum_i w_i \cdot y_i}{\sqrt{\sum_i w_i^2}\sqrt{\sum_i y_i^2}}$$

L2 Norm

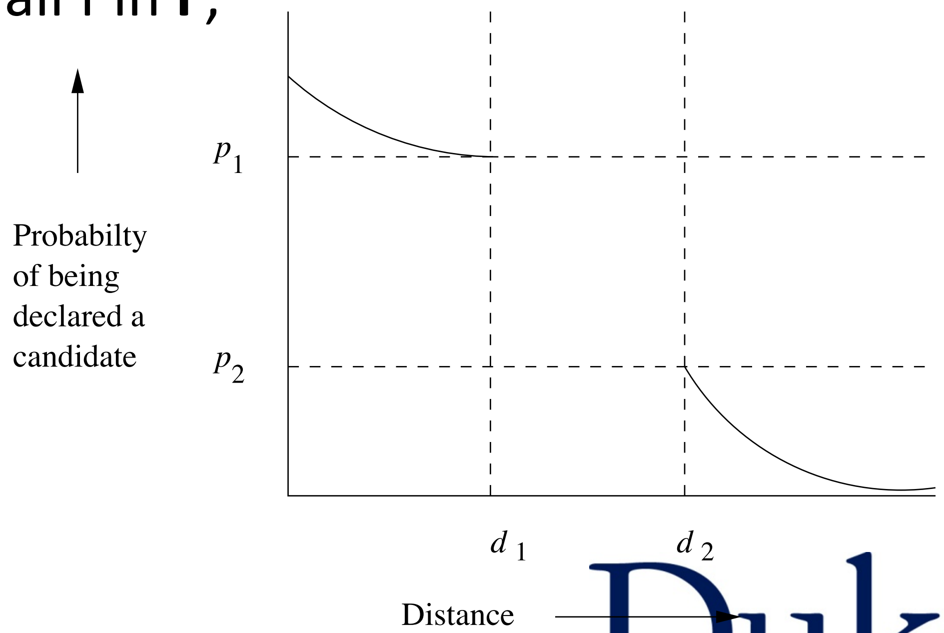
Locality Sensitive Hashing Idea

Construct a family of hash functions F.

Call x and y similar if for a randomly chosen f in F, f(x) = f(y)

Let d1 and d2 be two distances. A family of functions **F** is said to be *(d1, d2, p1, p2)-sensitive* if for all f in **F**,

- If d(x,y) < d1,
  then P[f(x) = f(y)] > p1
- If d(x,y) > d2,
  then P[f(x) = f(y)] < p2

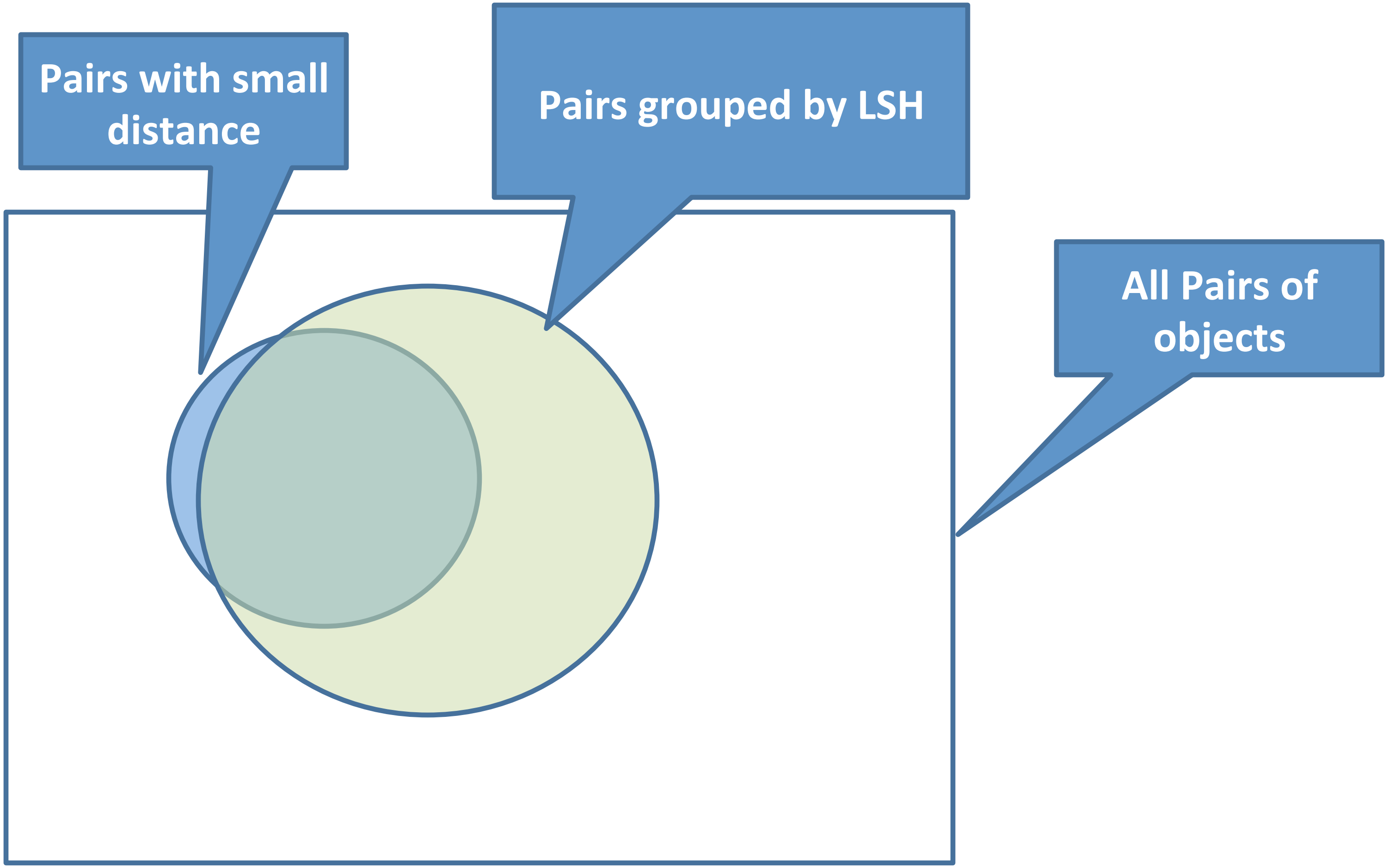


LSH: Motivation

- Naïve pairwise: $|S|^2$ pairwise comparisons
  - 1000 news articles each from 1,000 different topics
  - 1 trillion comparisons
  - 11.6 CPU days (if each comparison is 1 μs)

- Mentions from different topics are unlikely to have high similarity
  - Group by topic (can possibly miss some similar pairs, but very unlikely)
  - 1 billion comparisons
  - 16 CPU minutes (if each comparison is 1 μs)

LSH: Motivation



minHash (Minwise Independent Permutations)

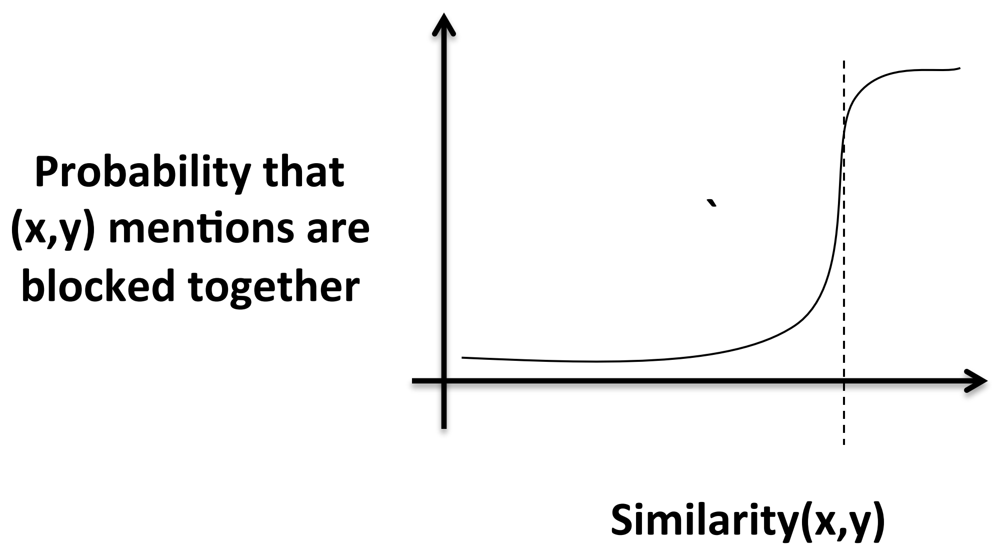
- Let F_x be a set representation of object x
  - Words in the document
  - character ngrams
  - Etc.
- Let π be a random permutation of features in F_x
  - E.g., order imposed by a random hash function
- *minHash(x)* = minimum element in F_x according to π

Why minHash works?

Surprising property: For a random permutation π,

$$P(minHash(x) = minhash(y)) = \frac{F_x \cap F_y}{F_x \cup F_y}$$

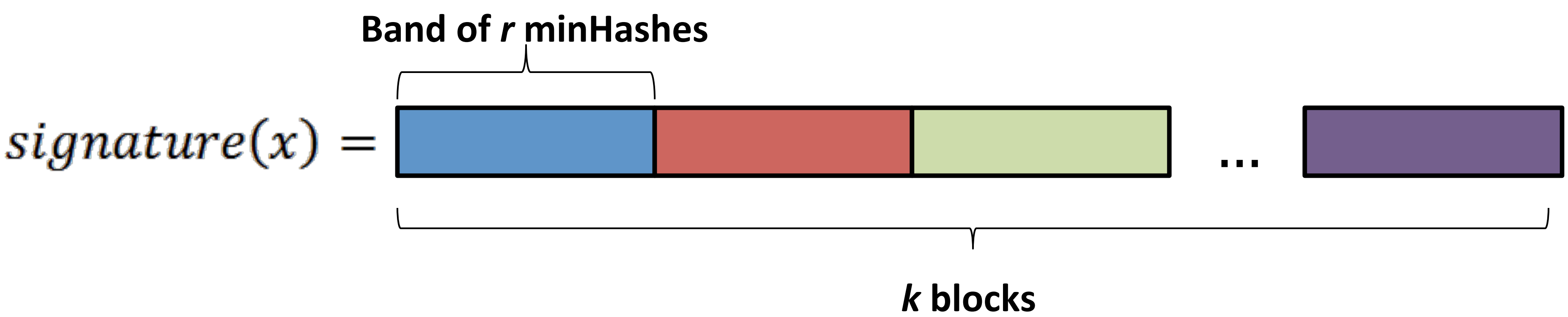
How to build a blocking scheme such that only pairs with Jacquard similarity > s fall in the same block (with high prob)?



Duke UNIVERSITY

Blocking using minHashes

- Compute minHashes using *r * k* permutations (hash functions)



- Signature's that match on ***1 out of k*** bands, go to the same block.

Duke UNIVERSITY

minHash Analysis

False Negatives: (missing matches)

P(pair x,y not in the same block with Jacquard sim = s) $= (1 - s^r)^k$

should be very low for high similarity pairs

False Positives: (blocking non-matches)

P(pair x,y in the same block with Jacquard sim = s) $= k \times s^r$

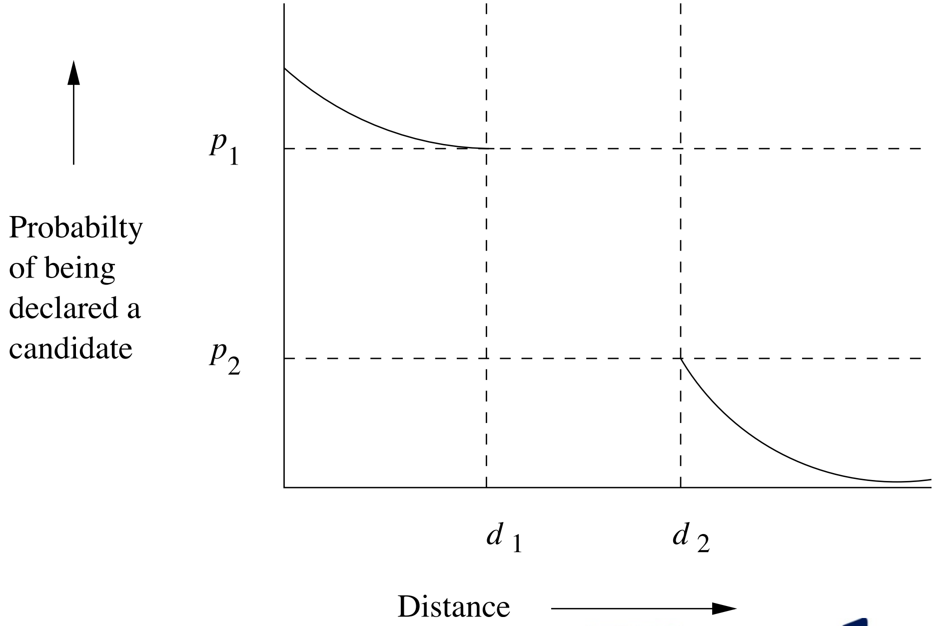
$r = 5, k = 20$

Sim(s)	P(not same block)
0.9	10^{-8}
0.8	0.00035
0.7	0.025
0.6	0.2
0.5	0.52
0.4	0.81
0.3	0.95
0.2	0.994
0.1	0.9998

Locality Sensitive Hashing Functions

Let d1 and d2 be two distances. A family of functions **F** is said to be *(d1, d2, p1, p2)-sensitive* if for all f in **F**,

- If d(x,y) < d1,
  then P[f(x) = f(y)] > p1
- If d(x,y) > d2,
  then P[f(x) = f(y)] < p2



Duke UNIVERSITY

Locality sensitive family for Jaccard distance

- minHash is one example of locality sensitive family that can strongly distinguish pairs that are close from pairs that are far.

- The family of minHash functions is a (d1, d2, 1-d1, 1-d2)-sensitive family for any d1, d2.

Amplifying a Locality-sensitive family

- AND construction:
  - Construct a new family F' consisting of r members of F
  - f in F' = {f1, f2, …, fr}
  - f(x) = f(y) iff for all i, fi(x) = fi(y)
  - If F is (d1, d2, p1, p2)-sensitive, then F' is (d1, d2, $p1^r$, $p2^r$)-sensitive

- OR construction:
  - Construct a new family F' consisting of b members of F
  - f in F' = {f1, f2, …, fb}
  - f(x) = f(y) iff there exists i, fi(x) = fi(y)
  - If F is (d1, d2, p1, p2)-sensitive,
    then F' is (d1, d2, $1-(1-p1)^b$, $1-(1-p2)^b$)-sensitive

Example

- Suppose F is (0.2, 0.6, 0.8, 0.4)-sensitive.
- We use AND-construction with r= 4 to create F1
- We use OR-construction with b=4 to create F2

- F2 is (0.2, 0.6, $1-(1-0.8^4)^4$, $1-(1-0.4^4)^4$)
  = (0.2, 0.6, 0.875, 0.0985)-sensitive

LSH for Hamming distance

- Given two vectors x, y
- Hamming distance h(x,y) = number of positions where x and y are different

- minHash: (d1, d2, 1-d1/d, 1-d2/d)-sensitive

LSH for Cosine Distance

- Cosine Distance: angle between two vectors
- Locality sensitive function **F**:
  Pick a random vector vf.
  f(x) = f(y) is x.vf and y.vf have the same sign.
- **F** is (d1, d2, (180-d1)/180, d2/180)-sensitive
- Another method:
  Generate v in $\{-1, +1\}^d$ (d is the dimensionality of x)
  f(x) = f(y) is x.vf and y.vf have the same sign.

Duke UNIVERSITY

Summary of Locality Sensitive Hashing

- Locality sensitive hashing functions can strongly distinguish pairs that are close from pairs that are far.

- AND and OR construction help amplify the distinguishing capability of locality sensitive functions.

- Used in almost all production systems that require efficient similarity computation.