

# Theory of LSH

Distance Measures

LS Families of Hash Functions

S-Curves

LSH for Different Distance Measures

Mining of Massive Datasets

Leskovec, Rajaraman, and Ullman

Stanford University



# Distance Measures

- Generalized LSH is based on some kind of “distance” between points.
  - Similar points are “close.”
- Jaccard similarity is not a distance;  $1 - \text{Jaccard similarity}$  is.
- Two major classes of distance measure:
  1. *Euclidean*
  2. *Non-Euclidean*

# Euclidean Vs. Non-Euclidean

- A *Euclidean space* has some number of real-valued dimensions and “dense” points.
  - There is a notion of “average” of two points.
  - A *Euclidean distance* is based on the locations of points in such a space.
- Any other space is *Non-Euclidean*.
  - Distance measures for non-Euclidean spaces are based on properties of points, but not their “location” in a space.

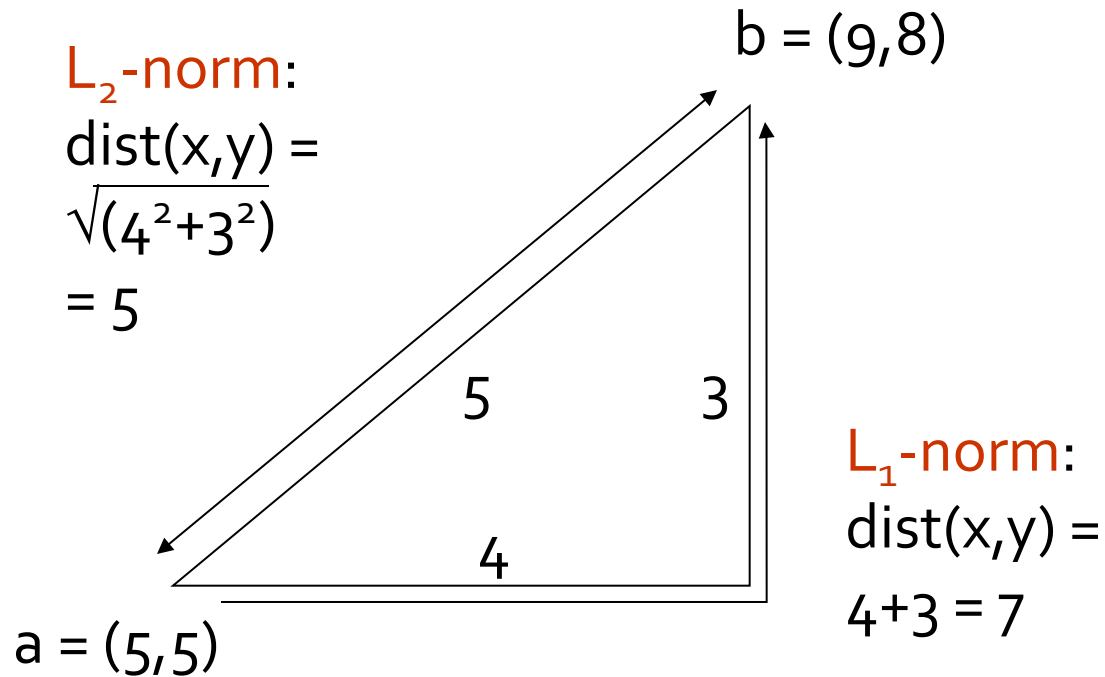
# Axioms of a Distance Measure

- $d$  is a *distance measure* if it is a function from pairs of points to real numbers such that:
  1.  $d(x,y) \geq 0$ .
  2.  $d(x,y) = 0$  iff  $x = y$ .
  3.  $d(x,y) = d(y,x)$ .
  4.  $d(x,y) \leq d(x,z) + d(z,y)$  (*triangle inequality*).

# Some Euclidean Distances

- $L_2$  norm:  $d(x,y)$  = square root of the sum of the squares of the differences between  $x$  and  $y$  in each dimension.
  - The most common notion of “distance.”
- $L_1$  norm: sum of the differences in each dimension.
  - *Manhattan distance* = distance if you had to travel along coordinates only.

# Examples of Euclidean Distances



# More Euclidean Distances

- $L_\infty$  norm:  $d(x,y)$  = the maximum of the differences between  $x$  and  $y$  in any dimension.
- **Note**: the maximum is the limit as  $r$  goes to  $\infty$  of the  $L_r$  norm: what you get by taking the  $r^{\text{th}}$  power of the differences, summing and taking the  $r^{\text{th}}$  root.

# Non-Euclidean Distances

- *Jaccard distance* for sets = 1 minus Jaccard similarity.
- *Cosine distance* for vectors = angle between the vectors.
- *Edit distance* for strings = number of inserts and deletes to change one string into another.
- *Hamming Distance* for bit vectors = the number of positions in which they differ.



# Example: Jaccard Distance

- Consider  $x = \{1,2,3,4\}$  and  $y = \{1,3,5\}$
- Size of intersection = 2; size of union = 5,  
Jaccard similarity (not distance) =  $2/5$ .
- $d(x,y) = 1 - (\text{Jaccard similarity}) = 3/5$ .

# Why J.D. Is a Distance Measure

- $d(x,y) \geq 0$  because  $|x \cap y| \leq |x \cup y|$ .
- $d(x,x) = 0$  because  $x \cap x = x \cup x$ .
  - And if  $x \neq y$ , then the size of  $x \cap y$  is strictly less than the size of  $x \cup y$ .
- $d(x,y) = d(y,x)$  because union and intersection are symmetric.
- $d(x,y) \leq d(x,z) + d(z,y)$  trickier – next slide.

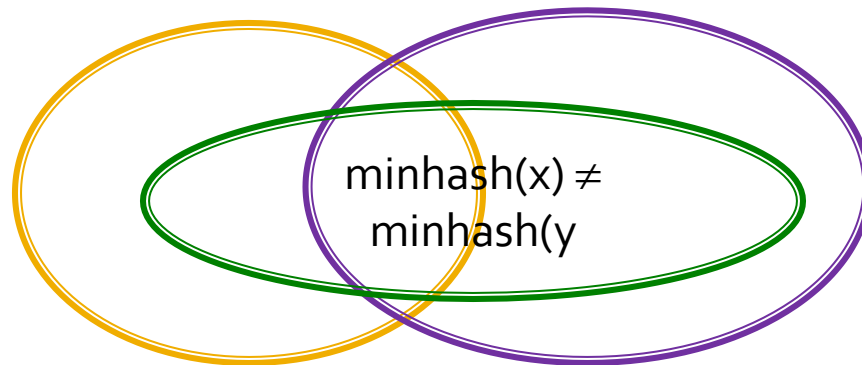
# Triangle Inequality for J.D.

$$1 - \frac{|x \cap z|}{|x \cup z|} + 1 - \frac{|y \cap z|}{|y \cup z|} \geq 1 - \frac{|x \cap y|}{|x \cup y|}$$

- **Remember:**  $|a \cap b| / |a \cup b|$  = probability that  $\text{minhash}(a) = \text{minhash}(b)$ .
- Thus,  $1 - |a \cap b| / |a \cup b|$  = probability that  $\text{minhash}(a) \neq \text{minhash}(b)$ .

# Triangle Inequality – (2)

- **Claim:**  $\text{prob}[\text{minhash}(x) \neq \text{minhash}(y)] \leq \text{prob}[\text{minhash}(x) \neq \text{minhash}(z)] + \text{prob}[\text{minhash}(z) \neq \text{minhash}(y)]$
- **Proof:** whenever  $\text{minhash}(x) \neq \text{minhash}(y)$ , at least one of  $\text{minhash}(x) \neq \text{minhash}(z)$  and  $\text{minhash}(z) \neq \text{minhash}(y)$  must be true.



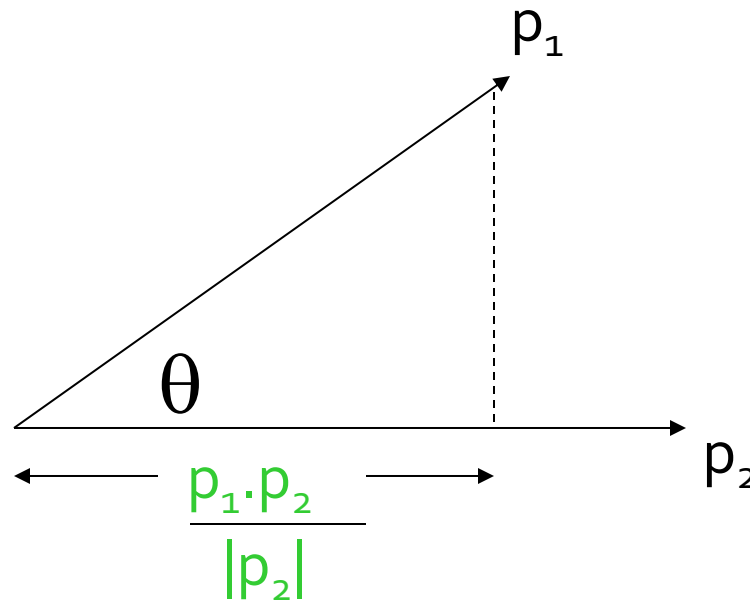
$\text{minhash}(x) \neq \text{minhash}(z)$

$\text{minhash}(z) \neq \text{minhash}(y)$

# Cosine Distance

- Think of a point as a vector from the origin  $(0,0,\dots,0)$  to its location.
- Two points' vectors make an angle, whose cosine is the normalized dot-product of the vectors:  $p_1 \cdot p_2 / |p_2| |p_1|$ .
  - **Example:**  $p_1 = 00111$ ;  $p_2 = 10011$ .
  - $p_1 \cdot p_2 = 2$ ;  $|p_1| = |p_2| = \sqrt{3}$ .
  - $\cos(\theta) = 2/3$ ;  $\theta$  is about 48 degrees.

# Cosine-Measure Diagram



$$d(p_1, p_2) = \theta = \arccos\left(\frac{p_1 \cdot p_2}{|p_2| |p_1|}\right)$$

# Why C.D. Is a Distance Measure

- $d(x,x) = 0$  because  $\arccos(1) = 0$ .
- $d(x,y) \geq 0$  because any two intersecting vectors make an angle in the range 0 to 180 degrees.
- $d(x,y) = d(y,x)$  by symmetry.
- **Triangle inequality**: physical reasoning. If I rotate an angle from  $x$  to  $z$  and then from  $z$  to  $y$ , I can't rotate less than from  $x$  to  $y$ .

# Edit Distance

- The *edit distance* of two strings is the number of inserts and deletes of characters needed to turn one into the other.
- An equivalent definition:  $d(x,y) = |x| + |y| - 2|LCS(x,y)|$ .
  - LCS = *longest common subsequence* = any longest string obtained both by deleting from  $x$  and deleting from  $y$ .



# Example

- $x = abcde$  ;  $y = bcduve$ .
- Turn  $x$  into  $y$  by deleting  $a$ , then inserting  $u$  and  $v$  after  $d$ .
  - Edit distance = 3.
- Or,  $LCS(x,y) = bcde$ .
- **Note:**  $|x| + |y| - 2|LCS(x,y)| = 5 + 6 - 2*4 = 3$   
= edit distance.

# Why Edit Distance Is a Distance Measure

- $d(x,x) = 0$  because 0 edits suffice.
- $d(x,y) \geq 0$ : no notion of negative edits.
- $d(x,y) = d(y,x)$  because insert/delete are inverses of each other.
- **Triangle inequality**: changing  $x$  to  $z$  and then to  $y$  is one way to change  $x$  to  $y$ .

# Hamming Distance

- *Hamming distance* is the number of positions in which bit-vectors differ.
- **Example:**  $p_1 = 10101$ ;  $p_2 = 10011$ .
- $d(p_1, p_2) = 2$  because the bit-vectors differ in the 3<sup>rd</sup> and 4<sup>th</sup> positions.

# Why Hamming Distance Is a Distance Measure

- $d(x,x) = 0$  since no positions differ.
- $d(x,y) \geq 0$  since strings cannot differ in a negative number of positions.
- $d(x,y) = d(y,x)$  by symmetry of “different from.”
- **Triangle inequality**: changing  $x$  to  $z$  and then to  $y$  is one way to change  $x$  to  $y$ .