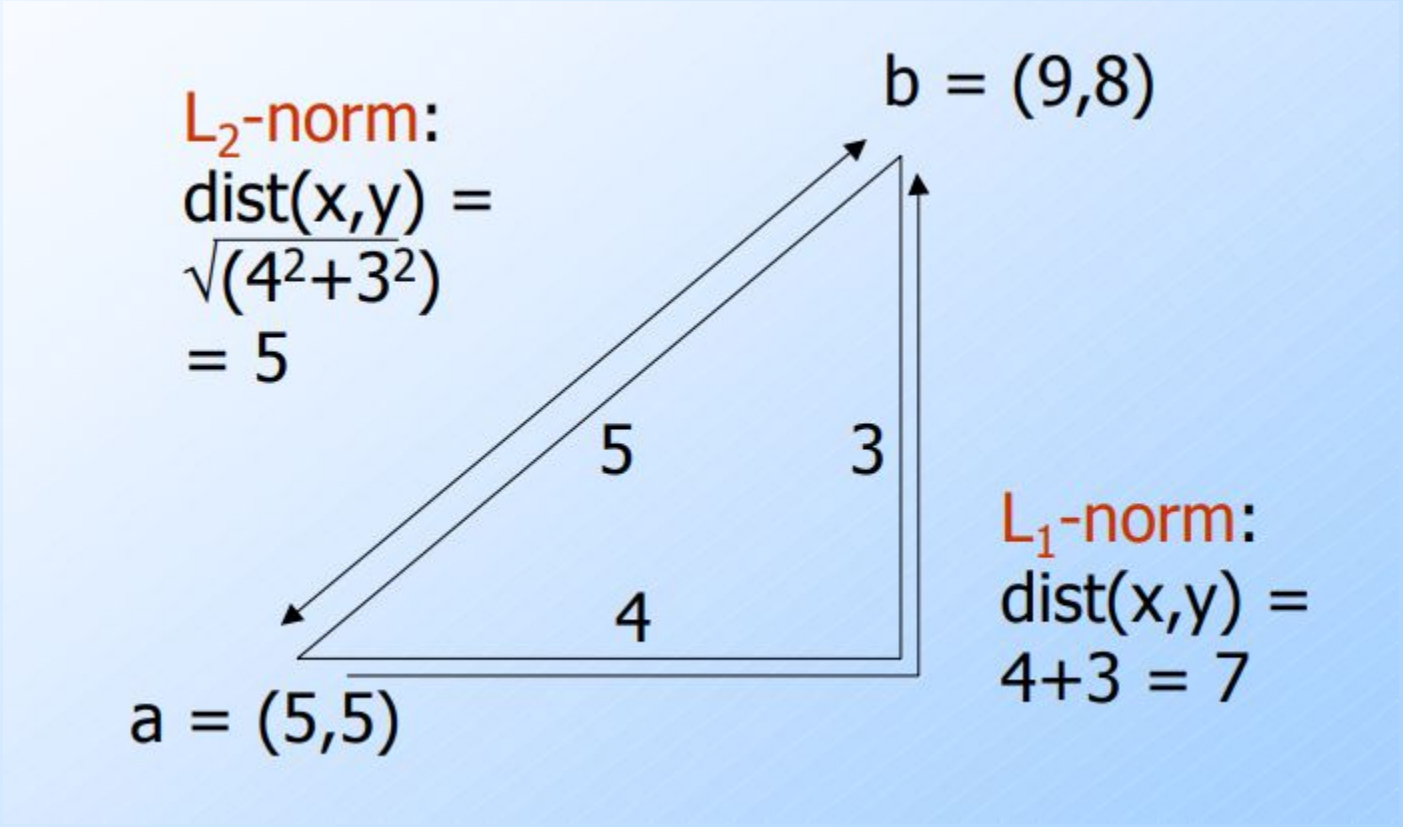
Measures of similarity

Kaustubh Kulkarni

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*).

Euclidean Distances



Euclidean Distances

- *L_1 norm* : sum of the differences in each dimension.
  - *Manhattan distance* = distance if you had to travel along coordinates only.
  - Used in k-NN
- *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."
  - Used in k-means

Minkowski Distance

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \quad for\ p \geq 1$$

p = 1, the Minkowski distance equation takes the same form as that of Manhattan distance

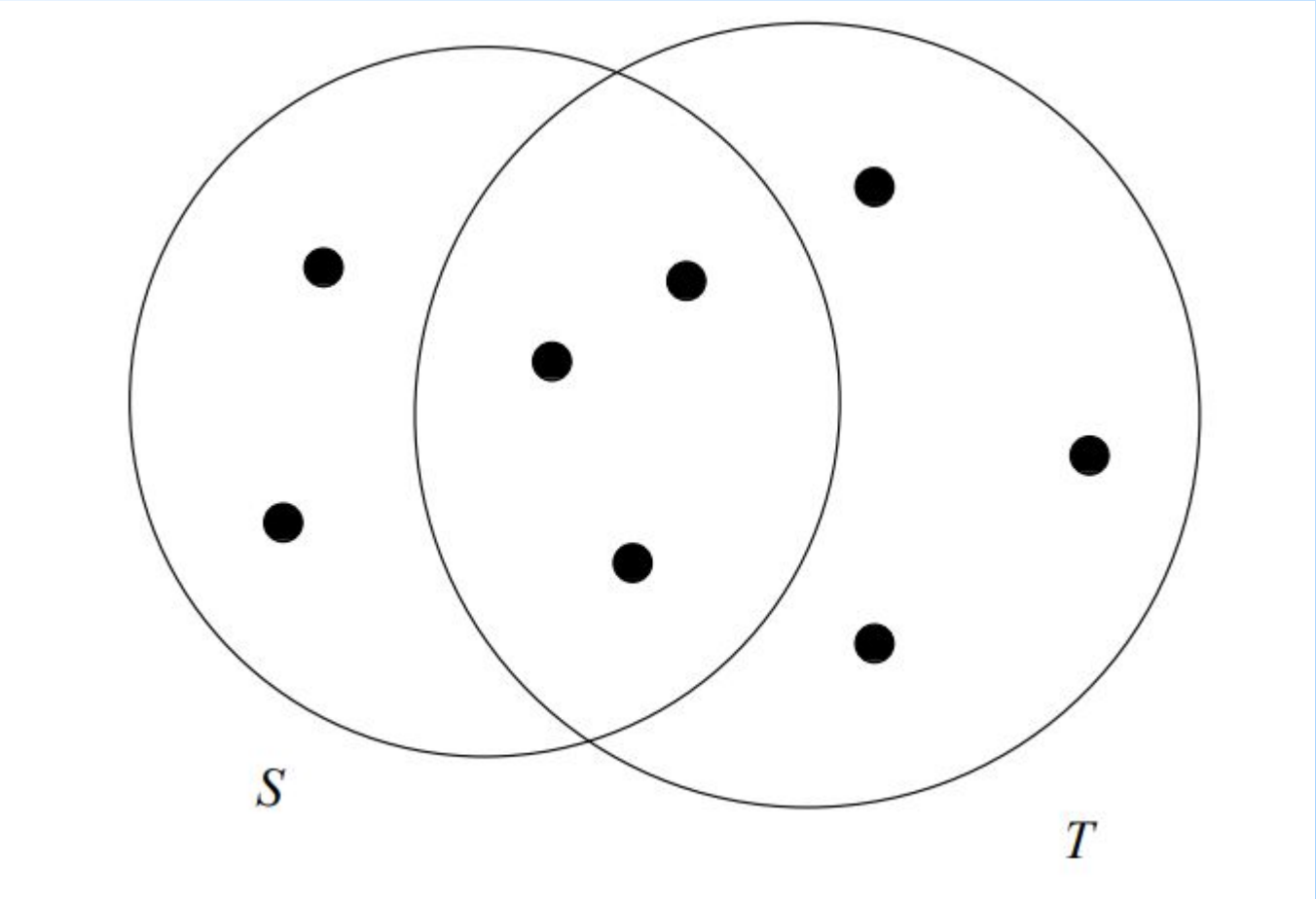
p = 2, the Minkowski distance is equivalent to the Euclidean distance

Jaccard Similarity (Index)

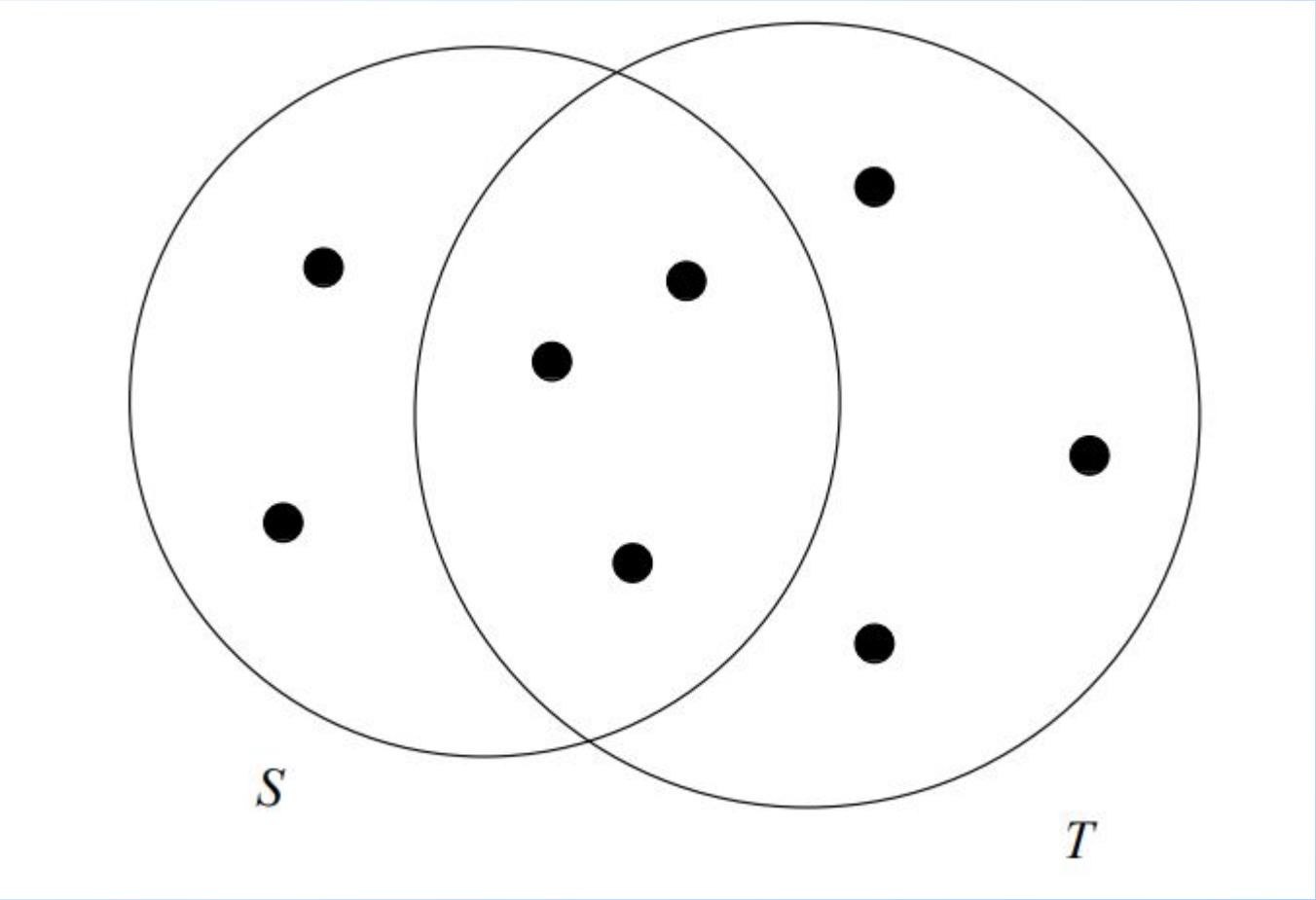
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}$$

Find the Jaccard Similarity



Ans : 3/8



Application : Text similarity

- An important class of problems that Jaccard similarity addresses well is that of finding textually similar documents in a large corpus such as the Web or a collection of news articles.
- The aspect of similarity we are looking at here is character-level similarity.

Application: Recommender Systems (Collaborative filtering)

Online Purchasing

- Amazon.com has millions of customers and sells millions of items.
- Its database records which items have been bought by which customers.
- We can say two customers are similar if their sets of purchased items have a high Jaccard similarity.
- Likewise, two items that have sets of purchasers with high Jaccard similarity will be deemed similar.
- Even a Jaccard similarity like 20% might be unusual enough to identify customers with similar tastes. The same observation holds for items; Jaccard similarities need not be very high to be significant.

Movie Ratings

- Netflix records which movies each of its customers rented, and also the ratings assigned to those movies by the customers.
- We can see movies as similar if they were rented or rated highly by many of the same customers, and see customers as similar if they rented or rated highly many of the same movies.
- When our data consists of ratings rather than binary decisions (bought/did not buy or liked/disliked), we cannot rely simply on sets as representations of customers or items.

1. Ignore low-rated customer/movie pairs; that is, treat these events as if the customer never watched the movie.
2. When comparing customers, imagine two set elements for each movie, "liked" and "hated."
    a. If a customer rated a movie highly, put the "liked" for that movie in the customer's set.
    b. If they gave a low rating to a movie, put "hated" for that movie in their set.
    c. Then, we can look for high Jaccard similarity among these sets.
    d. We can do a similar trick when comparing movies.
3. If ratings are 1-to-5-stars, put a movie in a customer's set n times if they rated the movie n-stars. Then, use Jaccard similarity for bags when measuring the similarity of customers.

Jaccard similarity for bags

- The is defined by counting an element n times in the **intersection** if n is the **minimum** of the number of times the element appears in both the bags.
- In the **union**, we count the element the **sum** of the number of times it appears in both the bags.
- {a, a, a, b} and {a, a, b, b, c}

- The intersection counts a twice and b once, so its size is 3.
- The size of the union of two bags is always the sum of the sizes of the two bags, or 9 in this case.
- The bag-similarity of bags {a, a, a, b} and {a, a, b, b, c} is 1/3.

Jaccard Distance for Sets

- Jaccard distance for sets = 1 minus Jaccard similarity.

$$\begin{aligned}\text{Jaccard Distance} J_D(A,B) &= 1 - \text{Jaccard Similarity} J(A,B) \\ &= 1 - \frac{|A \cap B|}{|A \cup B|} \\ &= \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \\ &= \frac{|A \triangle B|}{|A \cup B|}\end{aligned}$$

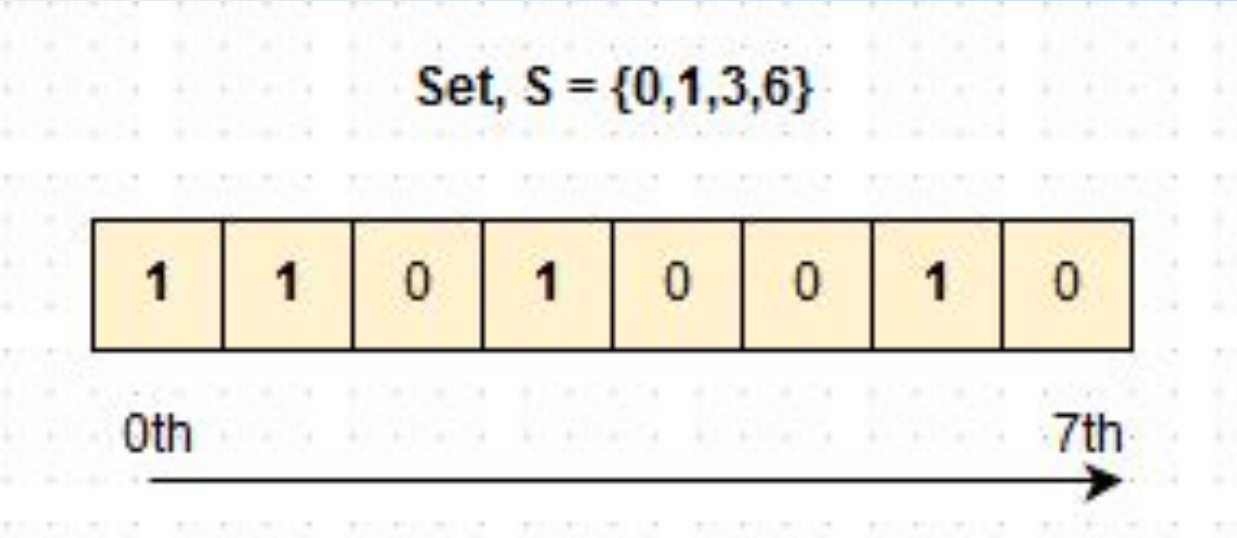
Why J.D. Is a Distance Measure

- $d(x,x) = 0$ because $x \cap x = x \cup x$.
- $d(x,y) = d(y,x)$ because union and intersection are symmetric.
- $d(x,y) \geq 0$ because $|x \cap y| \leq |x \cup y|$.
- $d(x,y) \leq d(x,z) + d(z,y)$ (see next slide.)

Triangle Inequality for J.D.

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

Representing sets using bit vectors



- $a_j = 1$ **if** $j \in A$

  `01101001` **{ 0, 3, 5, 6 }**

  `76543210`

  `01010101` **{ 0, 2, 4, 6 }**

  `76543210`

Operations

- **&** **Intersection** `01000001` { 0, 6 }
- **|** **Union** `01111101` { 0, 2, 3, 4, 5, 6 }
- **^** **Symmetric difference** `00111100` { 2, 3, 4, 5 }
- **~** **Complement** `10101010` { 1, 3, 5, 7 }

- Example: p_1 = 10111; p_2 = 10011.
- Size of intersection = 3;
- size of union = 4,
- Jaccard similarity = 3/4.
- d(x,y) = 1 – (Jaccard similarity) = 1/4.

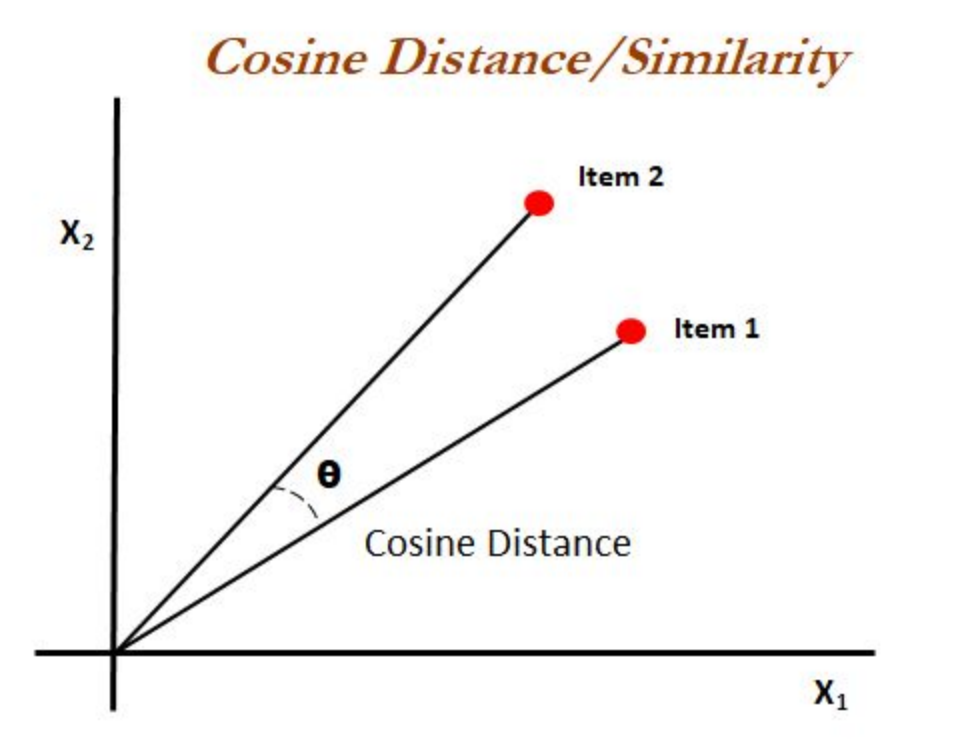
Cosine Distance

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

Cosine Similarity between A & B

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Item 1 = A, Item2 = B



Why C.D. Is a Distance Measure

- d(x,x) = 0 because arccos(1) = 0.
- d(x,y) = d(y,x) by symmetry.
- d(x,y) $\geq$ 0 because angles are chosen to be in the range 0 to 180 degrees.
- 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. Equivalently:
- 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: LCS

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

Variant Edit Distances

- Allow insert, delete, and *mutate*.
  - Change one character into another.
- Minimum number of inserts, deletes, and mutates also forms a distance measure.
- Ditto for any set of operations on strings.
  - Example: substring reversal OK for DNA sequences

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 3rd and 4th positions.

Why Hamming Distance Is a Distance Measure

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