Big Data Computing

Master's Degree in Computer Science

2022-2023

Gabriele Tolomei

Department of Computer Science

Sapienza Università di Roma

tolomei@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

Recap from Last Lecture(s)

- Dealing with big data requires new computing tools and paradigms
- Hadoop/MapReduce → useful in all those situations where data need to be accessed sequentially
- Spark → general-purpose distributed scalable data processing engine which provides an ecosystem of services to work on (big) data

Let's Start Our Journey Into Big Data!

CLUSTERING

What is Clustering?

- A procedure to group a set of objects into classes of similar objects

What is Clustering?

- A procedure to group a set of objects into classes of similar objects
- A standard problem in many (big) data applications:
  - Categorizing documents by their topics
  - Grouping customers by their behaviors
  - …

What is Clustering?

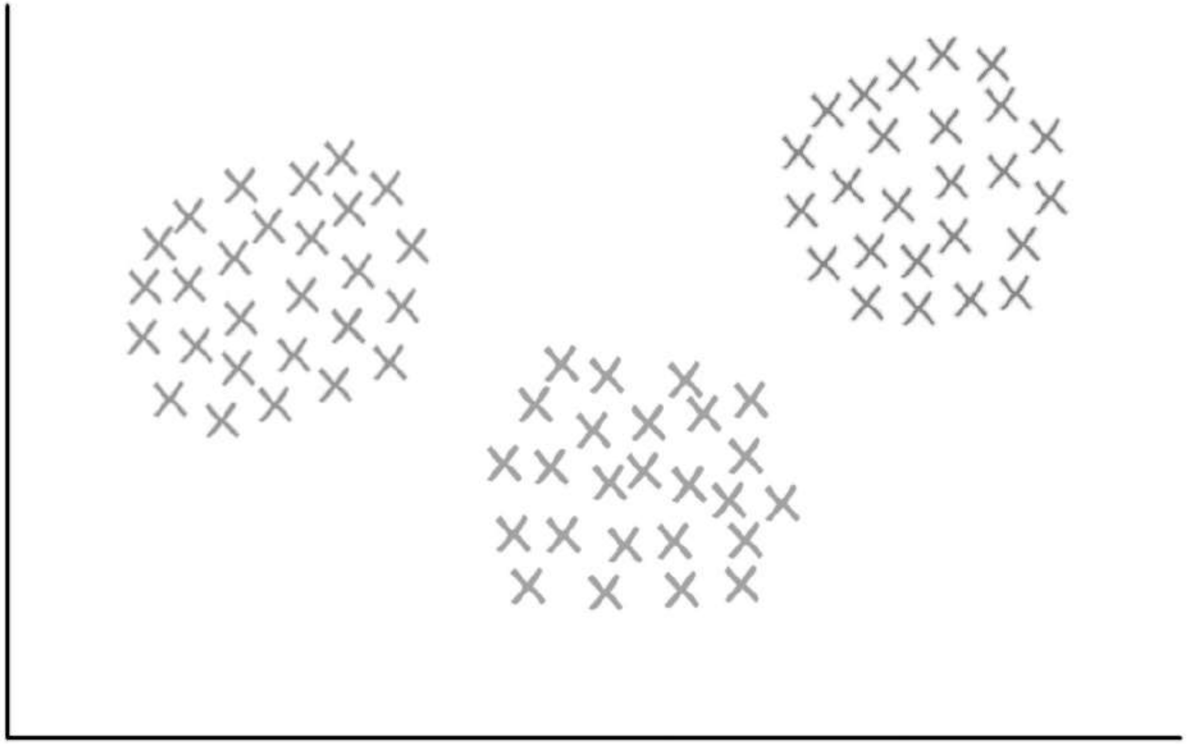
- A procedure to group a set of objects into classes of similar objects
- A standard problem in many (big) data applications:
  - Categorizing documents by their topics
  - Grouping customers by their behaviors
  - …
- A typical example of unsupervised learning technique

What is Clustering?

- A procedure to group a set of objects into classes of similar objects
- A standard problem in many (big) data applications:
  - Categorizing documents by their topics
  - Grouping customers by their behaviors
  - …
- A typical example of unsupervised learning technique
- A method of data exploration, i.e., a way of looking for patterns of interest in data

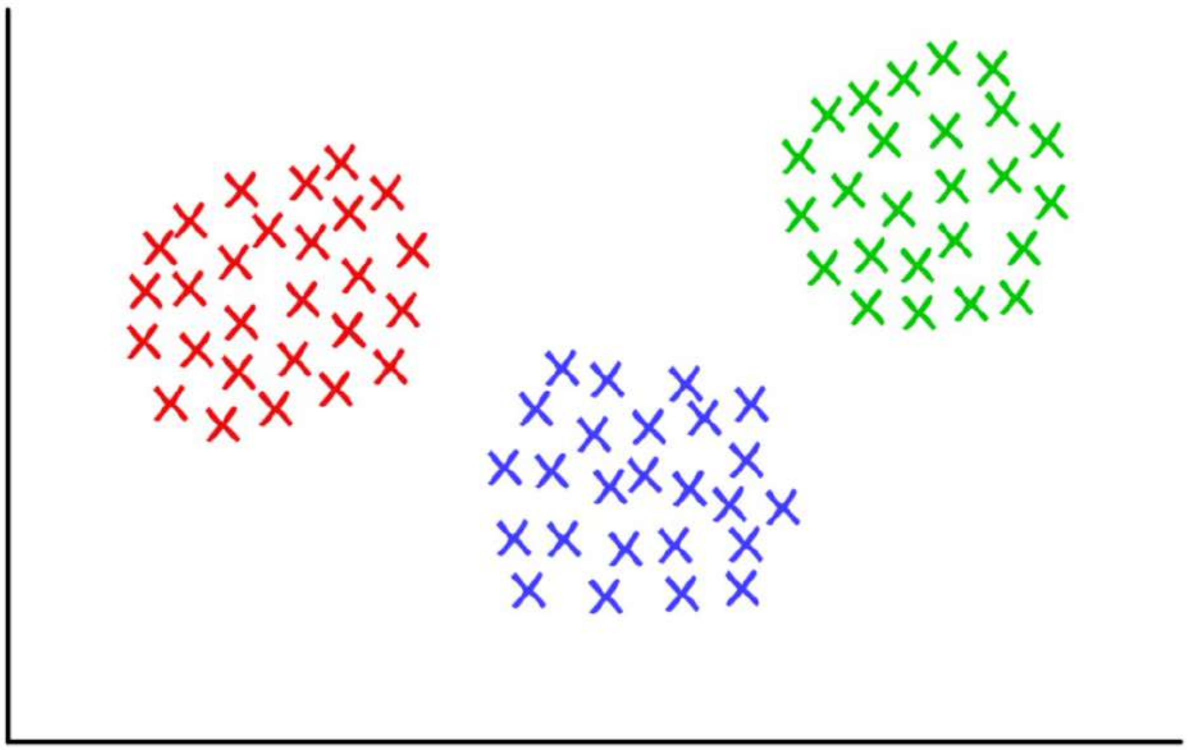
Clustering: Intuition

Given a set of 2-dimensional data points



Clustering: Intuition

We'd like to understand their "structure" in order to find groups of data points



Clustering: Formal Definition

- Given a set of data points and a notion of distance between those

Clustering: Formal Definition

- Given a set of data points and a notion of distance between those
- Group the data points into some number of clusters so that:
  - Members of a cluster are close/similar to each other (i.e., high intra-cluster similarity)
  - Members of different clusters are dissimilar (i.e., low inter-cluster similarity)

Clustering: Practical Issues

- Object representation
  - Data points may be in very high-dimensional spaces

Clustering: Practical Issues

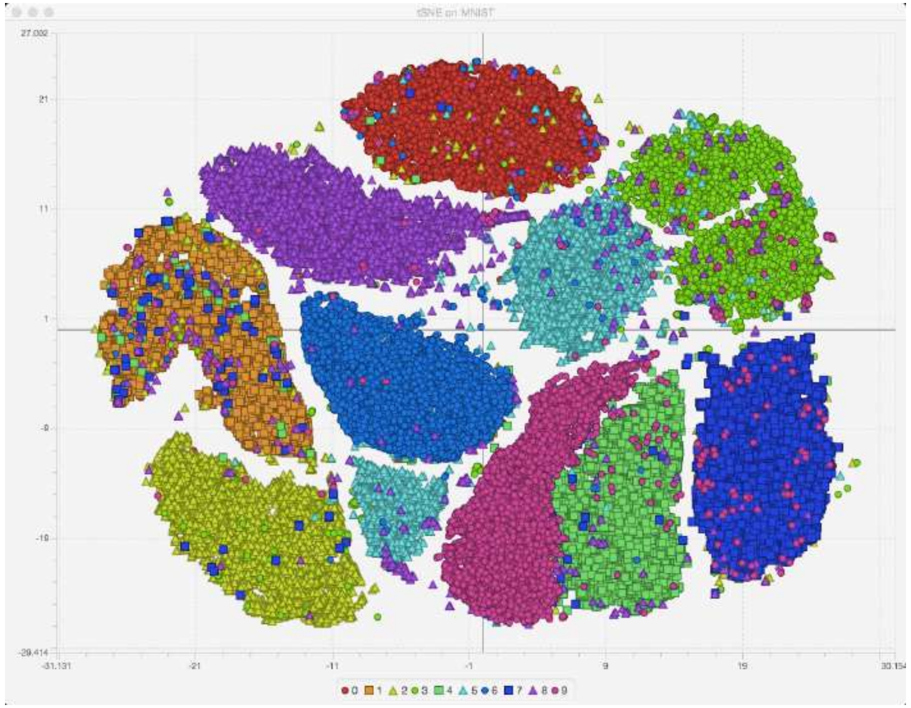
- Object representation
  - Data points may be in very high-dimensional spaces
- Notion of similarity between objects using a distance measure
  - Euclidean distance, Cosine similarity, Jaccard coefficient, etc.

Clustering: Practical Issues

- Object representation
  - Data points may be in very high-dimensional spaces
- Notion of similarity between objects using a distance measure
  - Euclidean distance, Cosine similarity, Jaccard coefficient, etc.
- Number of output clusters
  - Fixed apriori? Data-driven?

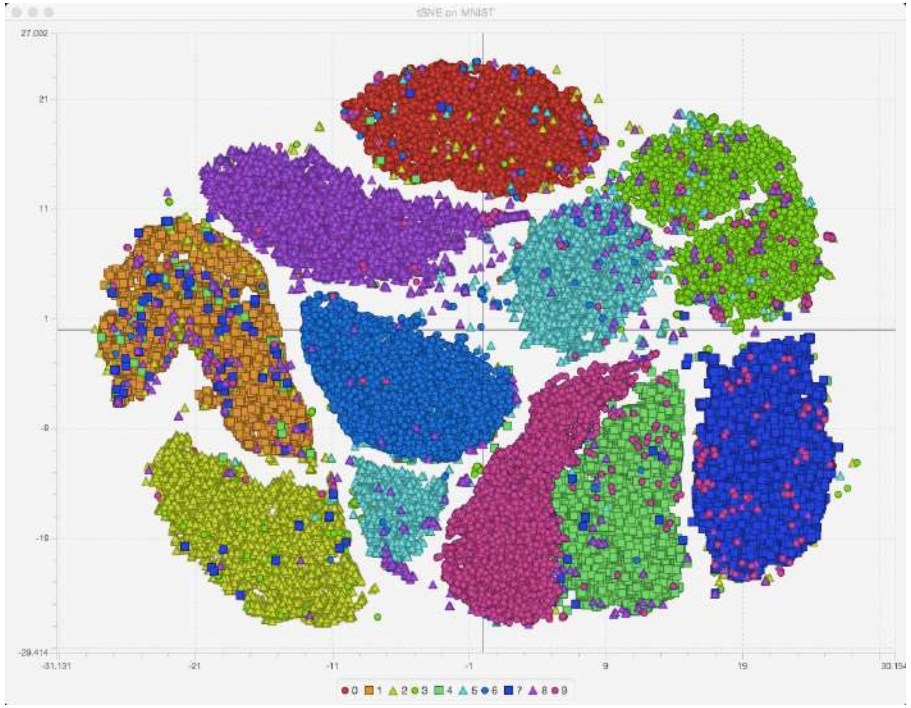
Clustering: A Hard Problem

Data points are not always easily and clearly separable



Clustering: A Hard Problem

Data points are not always easily and clearly separable



Finding a clear boundary between clusters may be hard in the real world

Clustering: A Hard Problem

- Clustering in 2 dimensions looks easy
- So does clustering of a small number of data points
- What does make things hard?

Clustering: A Hard Problem

- Clustering in 2 dimensions looks easy
- So does clustering of a small number of data points
- What does make things hard?

  Many real-world applications involve 10s, 100s, or 1,000s of dimensions

Clustering: A Hard Problem

- Clustering in 2 dimensions looks easy
- So does clustering of a small number of data points
- What does make things hard?

Many real-world applications involve 10s, 100s, or 1,000s of dimensions

In high-dimensional spaces almost all pairs of points are at the same (large) distance

High-Dimensional Spaces

- Data in a high-dimensional space tends to be sparser than in lower dimensions
    - Data points are more dissimilar to each other

High-Dimensional Spaces

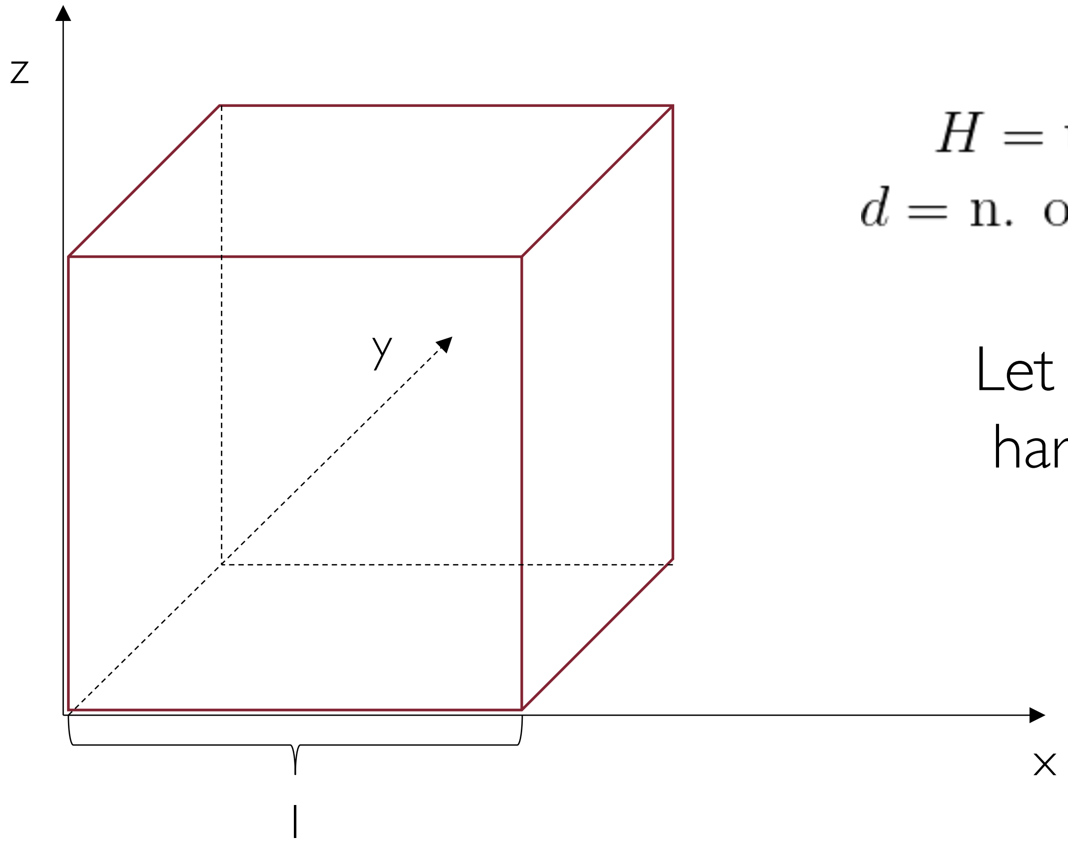
- Data in a high-dimensional space tends to be sparser than in lower dimensions
  - Data points are more dissimilar to each other
- In Euclidean space, the distance between two points is large as long as they are far apart along at least one dimension
  - The higher the number of dimensions the higher the chance this happens

High-Dimensional Spaces

- Data in a high-dimensional space tends to be sparser than in lower dimensions
  - Data points are more dissimilar to each other
- In Euclidean space, the distance between two points is large as long as they are far apart along at least one dimension
  - The higher the number of dimensions the higher the chance this happens

The Curse of Dimensionality

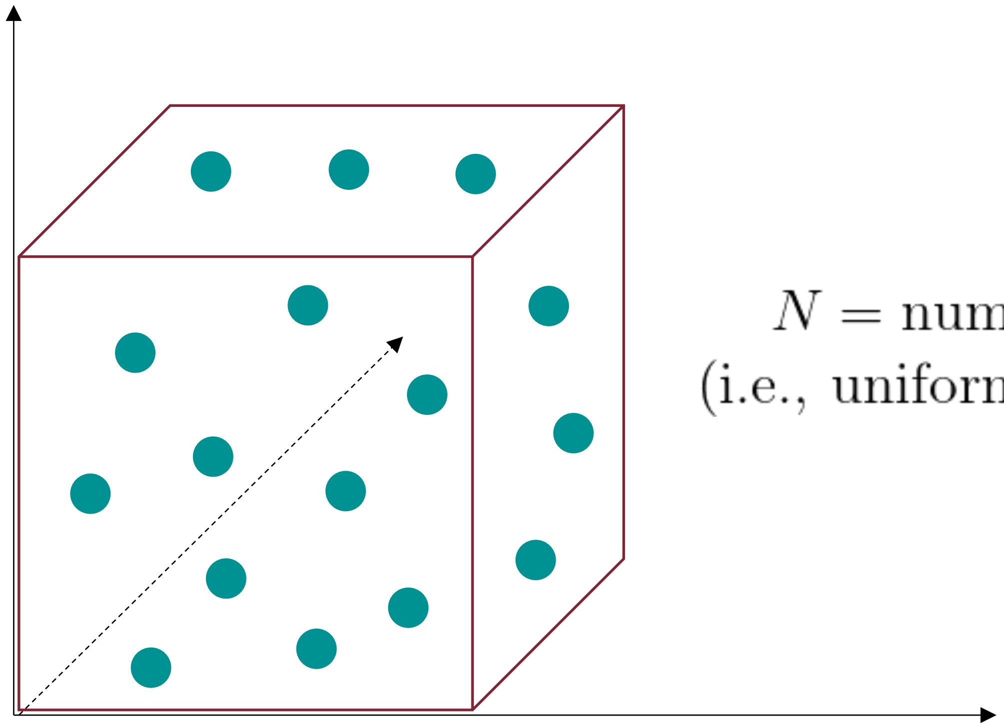
The Curse of Dimensionality



$$H = \text{unit-length hypercube in } \mathbb{R}^d$$
$$d = \text{n. of space dimensions}$$

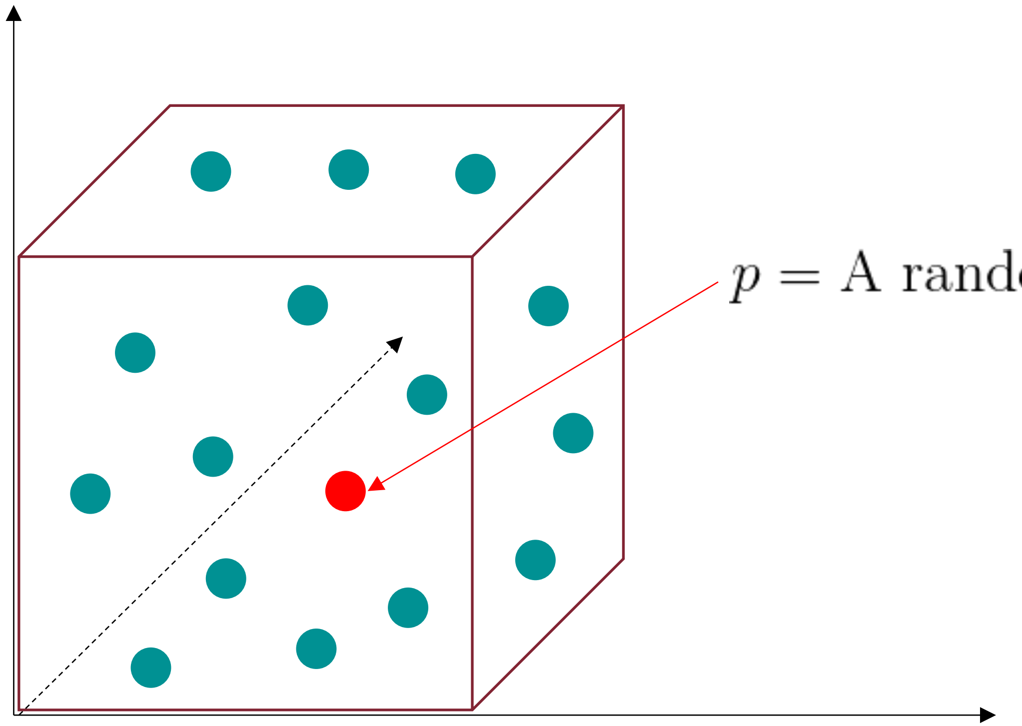
Let d=3 as beyond that it is hard to visualize the space

The Curse of Dimensionality



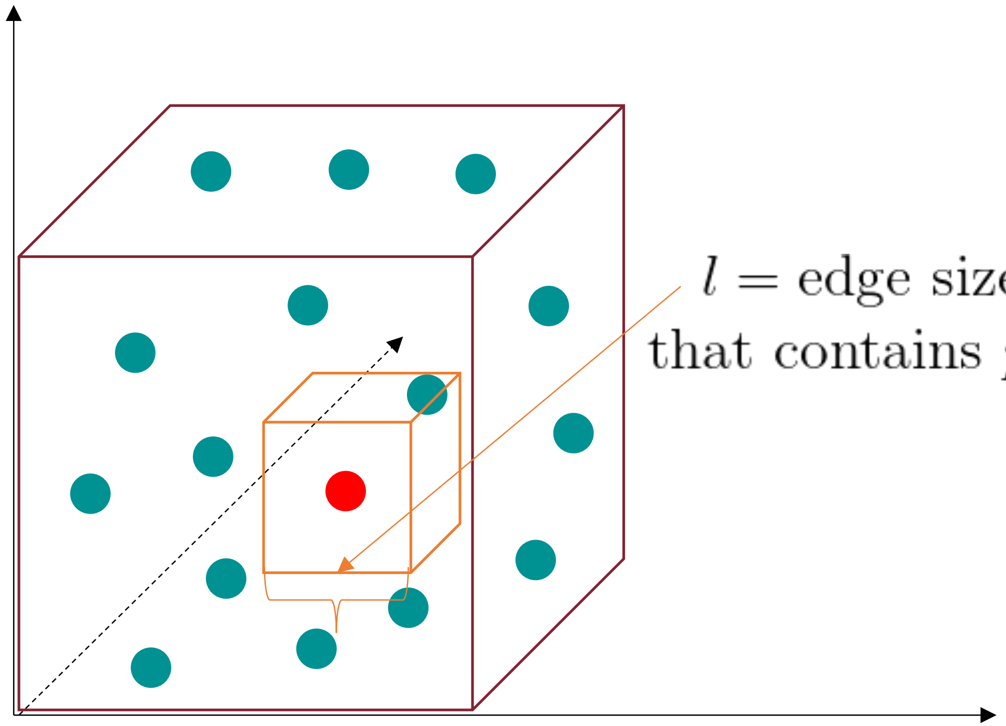
N = number of data points randomly (i.e., uniformly) distributed in H

The Curse of Dimensionality



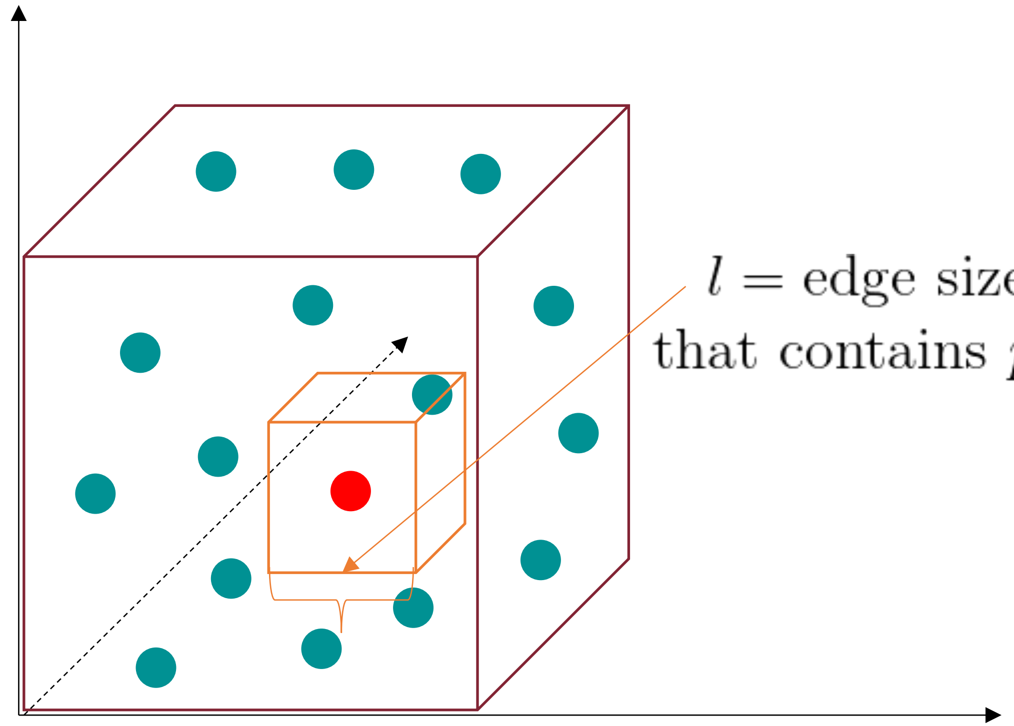
$p =$ A random point chosen among N

The Curse of Dimensionality



$l =$ edge size of the hypercube h inscribed in H that contains p along with its k nearest points

The Curse of Dimensionality

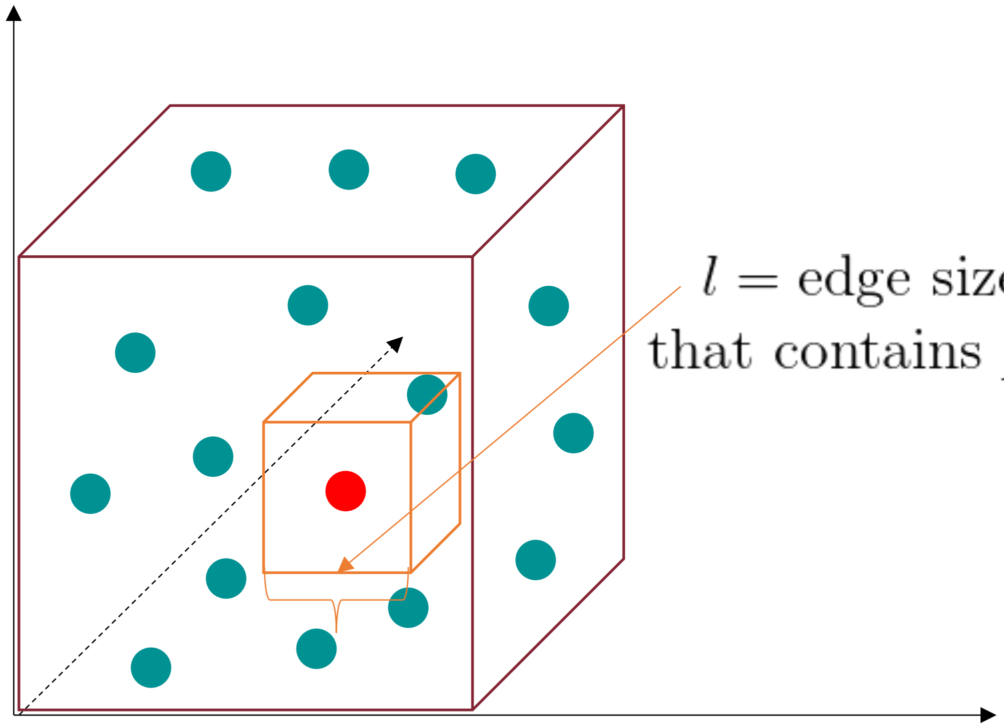


l = edge size of the hypercube h inscribed in H that contains p along with its k nearest points

We consider **edge points** whose distance from p is **at most** $\frac{l}{2}\sqrt{d}$

$$\boxed{\frac{l}{2}\sqrt{3}}$$

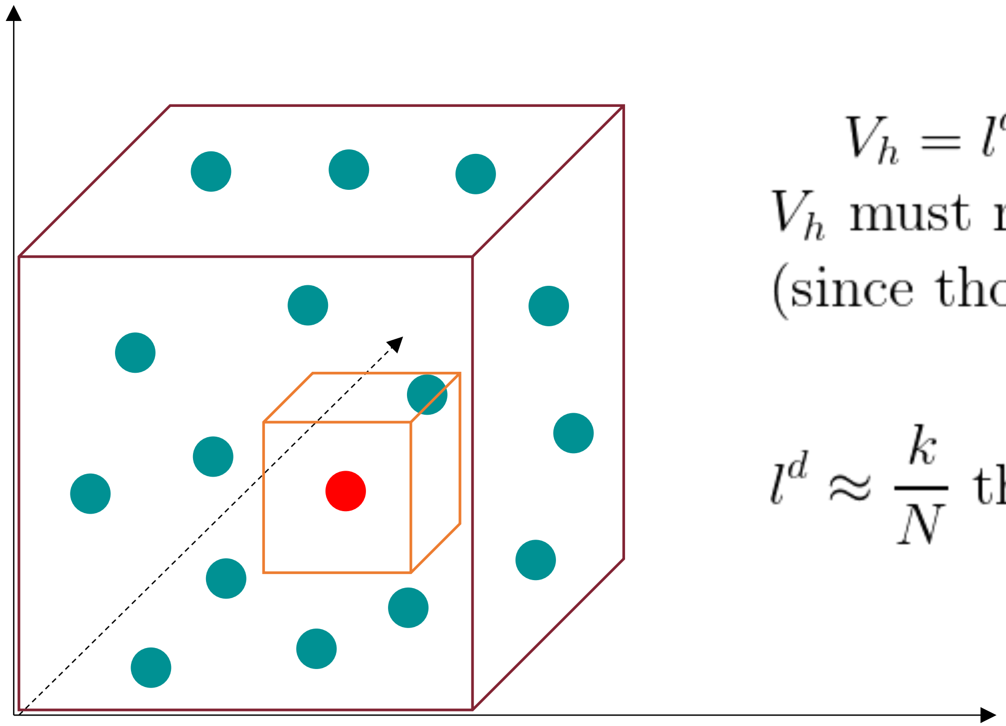
The Curse of Dimensionality



l = edge size of the hypercube h inscribed in H that contains p along with its k nearest points

The same question can be formulated in terms of the radius l of an inscribed hypersphere

The Curse of Dimensionality



$V_h = l^d$ volume of the hypercube h

V_h must roughly contain k/N points

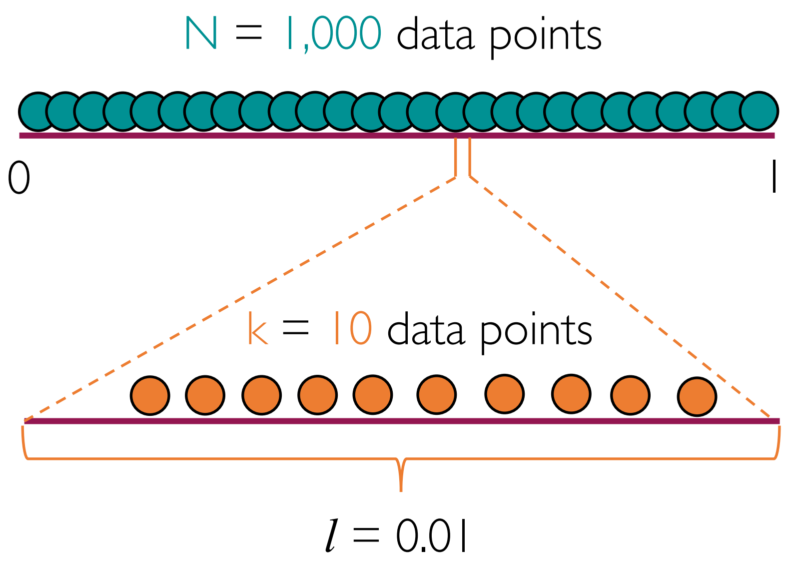
(since those are randomly distributed)

$$l^d \approx \frac{k}{N} \text{ therefore } l \approx \left(\frac{k}{N}\right)^{1/d}$$

The Curse of Dimensionality

A few numbers… $N = 1,000; k = 10 \quad l \approx \left(\frac{10}{1000}\right)^{1/d} = \left(\frac{1}{100}\right)^{1/d}$

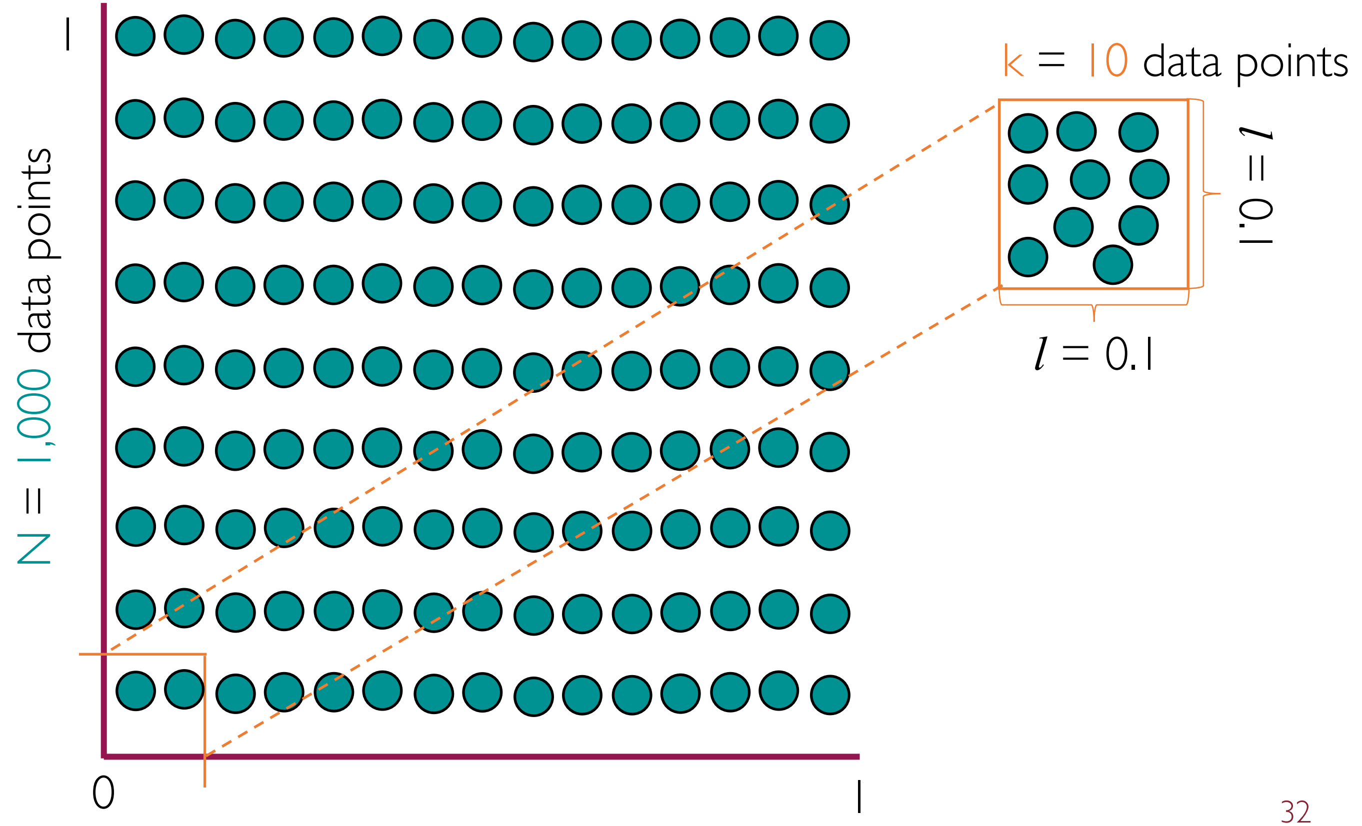
d	l
1	0.01



The Curse of Dimensionality

A few numbers… $N = 1,000; k = 10 \quad l \approx \left(\frac{10}{1000}\right)^{1/d} = \left(\frac{1}{100}\right)^{1/d}$

d	l
1	0.01
2	0.1



The Curse of Dimensionality

A few numbers… $N = 1,000; k = 10 \quad l \approx \left(\frac{10}{1000}\right)^{1/d} = \left(\frac{1}{100}\right)^{1/d}$

d	l
1	0.01
2	0.1
3	0.215
…	…
10	0.631

When *d* is equal 10 the length of the edge of the inscribed hypercube is already about 63% of the largest hypercube

The Curse of Dimensionality

A few numbers… $N = 1,000; k = 10 \quad l \approx \left(\frac{10}{1000}\right)^{1/d} = \left(\frac{1}{100}\right)^{1/d}$

d	l
1	0.01
2	0.1
3	0.215
…	…
10	0.631
…	…
1000	0.995

When d is equal 1,000 there is basically no difference between the two hypercubes!

The Curse of Dimensionality: Why Bother?

- Points are more likely to be located at the edges of the region

The Curse of Dimensionality: Why Bother?

- Points are more likely to be located at the edges of the region
- Nearest points are not close at all!

The Curse of Dimensionality: Why Bother?

- Points are more likely to be located at the edges of the region
- Nearest points are not close at all!
- Distance between points indistinguishable (distance concentration)
  - Hard to separate between nearest and furthest data points
  - Hard to find clusters among so many pairs that are all at approximately the same distance

The Curse of Dimensionality: The Edge

Let ε define the **edge** (i.e., border) of our space

The Curse of Dimensionality: The Edge

Let ε define the edge (i.e., border) of our space

See how the probability of picking a data point that is not located at the edge changes as the number of dimensions grow

The Curse of Dimensionality: The Edge

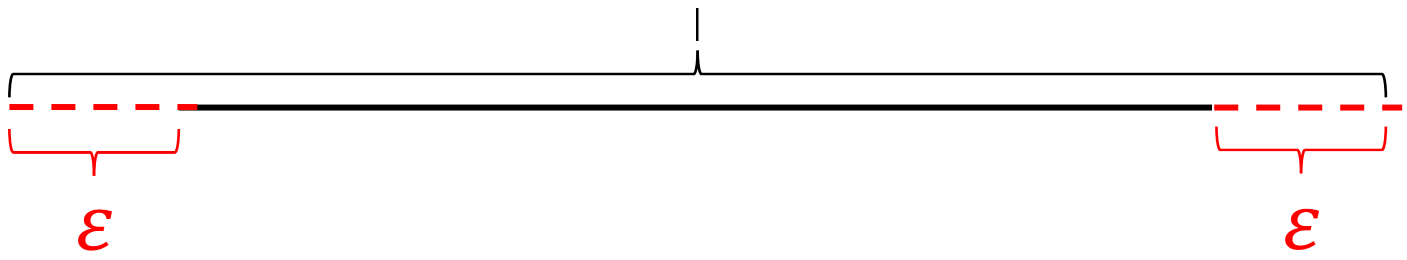
Let ε define the edge (i.e., border) of our space

See how the probability of picking a data point that is not located at the edge changes as the number of dimensions grow

Remember:
We assume data points are uniformly distributed at random on the space

The Curse of Dimensionality: The Edge

$d = 1$

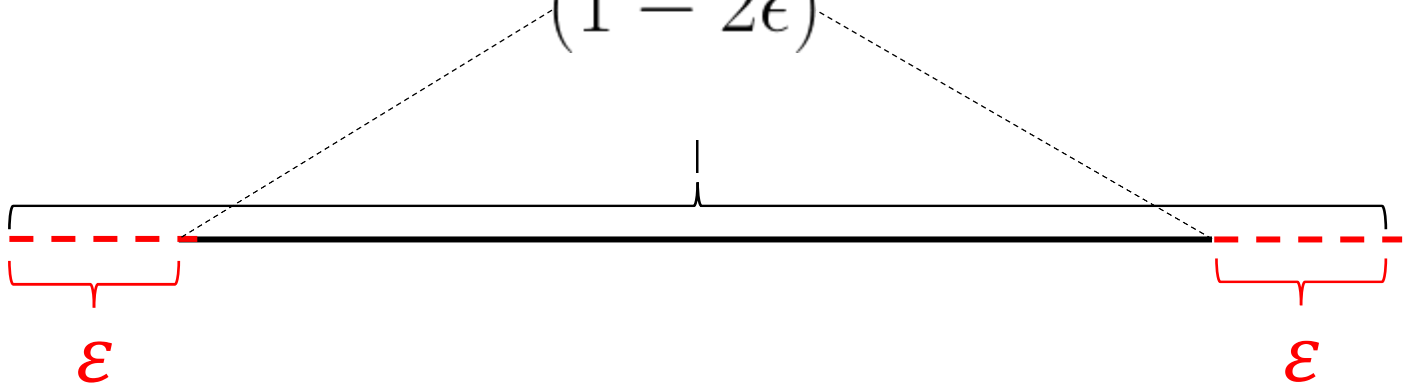


The Curse of Dimensionality: The Edge

$d = 1$

The probability of being **not** at the edge is just

$$(1 - 2\epsilon)$$



The Curse of Dimensionality: The Edge

$d > 1$

The probability of being **not** at the edge is the probability of being not at the edge on **every single dimension**

The Curse of Dimensionality: The Edge

$d > 1$

The probability of being **not** at the edge is the probability of being not at the edge on **every single dimension**

$$(1 - 2\epsilon)^d$$

assuming each dimension is independent from each other

The Curse of Dimensionality: The Edge

$d > 1$

The probability of being not at the edge is the probability of being not at the edge on every single dimension

$$(1 - 2\epsilon)^d$$

assuming each dimension is independent from each other

$$\lim_{d \to \infty} (1 - 2\epsilon)^d = 0$$

The Curse of Dimensionality

A Notebook where the Curse of Dimensionality is (visually) explained is available at the following link:

https://github.com/gtolomei/big-data-computing/blob/master/notebooks/The_Curse_Of_Dimensionality.ipynb

So What Can We Do?

- If data are really uniformly distributed in a high-dimensional space…

  nothing!

So What Can We Do?

- If data are really uniformly distributed in a high-dimensional space… nothing!
- Luckily, though, real-world (interesting) data have patterns underneath (i.e., they are not random!)

So What Can We Do?

- If data are really uniformly distributed in a high-dimensional space… nothing!
- Luckily, though, real-world (interesting) data have patterns underneath (i.e., they are not random!)
- Lower intrinsic dimensionality
  - The Manifold Hypothesis: High dimensional data (e.g., images) lie on low-dimensional manifolds (i.e., sub-space) embedded in the high-dimensional space

So What Can We Do?

- If data are really uniformly distributed in a high-dimensional space… nothing!
- Luckily, though, real-world (interesting) data have patterns underneath (i.e., they are not random!)
- Lower intrinsic dimensionality
  - The Manifold Hypothesis: High dimensional data (e.g., images) lie on low-dimensional manifolds (i.e., sub-space) embedded in the high-dimensional space
  - Dimensionality reduction techniques (more on this later…)

A Digression on Similarity Measures

- What does "similar" mean?

A Digression on Similarity Measures

- What does "similar" mean?
- No single answer! It depends on what we want to find or emphasize in the data

A Digression on Similarity Measures

- What does "similar" mean?
- No single answer! It depends on what we want to find or emphasize in the data
- Domain and representation specific

A Digression on Similarity Measures

- What does "similar" mean?
- No single answer! It depends on what we want to find or emphasize in the data
- Domain and representation specific
- The similarity measure is often more important than the clustering algorithm used itself!

Notion of Similarity

- So far, we haven't really talked about the similarity between objects

Notion of Similarity

- So far, we haven't really talked about the similarity between objects
- In fact, we implicitly assumed:
  - Data live in a *d*-dimensional Euclidean space
  - Similarity between data is computed using Euclidean metric (i.e., distance)

Notion of Similarity

- So far, we haven't really talked about the similarity between objects
- In fact, we implicitly assumed:
  - Data live in a *d*-dimensional Euclidean space
  - Similarity between data is computed using Euclidean metric (i.e., distance)
- Other metrics can be used depending on the domain
  - Cosine similarity
  - Jaccard coefficient

Metric and Metric Space

X is a set

δ is a function $\delta : X \times X \rightarrow [0, \infty)$, where:

1. $\delta(x, y) \geq 0$ (**non-negativity**)
2. $\delta(x, y) = 0 \Leftrightarrow x = y$ (**identity** of indiscernibles)
3. $\delta(x, y) = \delta(y, x)$ (**symmetry**)
4. $\delta(x, y) \leq \delta(x, z) + \delta(z, y)$ (**triangle inequality**)

Then δ is called a **metric** (or distance function) and X a **metric space**

Euclidean Metric (Distance) & Euclidean Space

$$X = \mathbb{R}^d$$

$$\delta : \mathbb{R}^d \times \mathbb{R}^d \to [0, \infty)$$

$\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d)$ are 2 points in $\mathbb{R}^d$

$$\delta(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + \ldots + (x_d - y_d)^2} = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2}$$

Euclidean Norm (L^2-Norm)

- The position of a point in a Euclidean d-space is a Euclidean vector

Euclidean Norm (L^2-Norm)

- The position of a point in a Euclidean *d*-space is a Euclidean vector
- The Euclidean norm of a vector measures its length (from the origin)

Euclidean Norm (L^2-Norm)

- The position of a point in a Euclidean *d*-space is a Euclidean vector
- The Euclidean norm of a vector measures its length (from the origin)

$$||\mathbf{x}||_2 = \sqrt{x_1^2 + \ldots + x_d^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

where $\cdot$ indicates the **dot product**

Euclidean Norm (L^2-Norm)

- The position of a point in a Euclidean *d*-space is a Euclidean vector
- The Euclidean norm of a vector measures its length (from the origin)

$$||\mathbf{x}||_2 = \sqrt{x_1^2 + \ldots + x_d^2} = \sqrt{\mathbf{x} \cdot \mathbf{x}}$$

where $\cdot$ indicates the **dot product**

This can be just seen as the Euclidean distance between vector's tail and tip

Euclidean Norm & Euclidean Metric

Let $\mathbf{x}-\mathbf{y} = (x_1-y_1, \ldots, x_d-y_d)$ the **displacement vector** between $\mathbf{x}$ and $\mathbf{y}$

Euclidean Norm & Euclidean Metric

Let $\mathbf{x}-\mathbf{y} = (x_1-y_1, \ldots, x_d-y_d)$ the **displacement vector** between $\mathbf{x}$ and $\mathbf{y}$

The Euclidean distance between x and y is just the Euclidean norm of the displacement vector

Euclidean Norm & Euclidean Metric

Let $\mathbf{x} - \mathbf{y} = (x_1 - y_1, \ldots, x_d - y_d)$ the **displacement vector** between $\mathbf{x}$ and $\mathbf{y}$

The Euclidean distance between x and y is just the Euclidean norm of the displacement vector

$$\delta(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2 = \sqrt{(\mathbf{x} - \mathbf{y}) \cdot (\mathbf{x} - \mathbf{y})}$$

Euclidean Distance: 1-dimensional Case

$$d = 1$$

$$\mathbf{x}, \mathbf{y} \in \mathbb{R}^d = \mathbb{R}$$

$$\mathbf{x} = x, \mathbf{y} = y \text{ both } \mathbf{x} \text{ and } \mathbf{y} \text{ are scalars}$$

$$\delta(\mathbf{x}, \mathbf{y}) = \delta(x, y) = \sqrt{(x - y)^2} = |x - y|$$

Euclidean Distance: 1-dimensional Case

$$d = 1$$

$$\mathbf{x}, \mathbf{y} \in \mathbb{R}^d = \mathbb{R}$$

$$\mathbf{x} = x, \mathbf{y} = y \text{ both } \mathbf{x} \text{ and } \mathbf{y} \text{ are scalars}$$

$$\delta(\mathbf{x}, \mathbf{y}) = \delta(x, y) = \sqrt{(x - y)^2} = |x - y|$$

The Euclidean distance between any two 1-*d* points on the real line is the absolute value of the numerical difference of their coordinates

Euclidean Distance: 2-dimensional Case

$$d = 2$$

$$\mathbf{x}, \mathbf{y} \in \mathbb{R}^d = \mathbb{R}^2$$

$$\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$$

$$\delta(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = ||\mathbf{x} - \mathbf{y}||_2$$

Euclidean Distance: 2-dimensional Case

$$d = 2$$
$$\mathbf{x}, \mathbf{y} \in \mathbb{R}^d = \mathbb{R}^2$$
$$\mathbf{x} = (x_1, x_2), \mathbf{y} = (y_1, y_2)$$

$$\delta(\mathbf{x}, \mathbf{y}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} = ||\mathbf{x} - \mathbf{y}||_2$$

The Euclidean distance between any two 2-*d* points on the Euclidean plane equals to the Pythagorean theorem

Minkowski Distance (L^p-Norm)

Generalization of the Euclidean distance

$$\mathbf{x} = (x_1, \ldots, x_d) \text{ and } \mathbf{y} = (y_1, \ldots, y_d) \in \mathbb{R}^d$$

$$\delta_p(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Minkowski Distance (L^p-Norm): p=1

L^1-Norm or Manhattan Distance

$$\delta_1(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^1 \right)^{\frac{1}{1}} = \sum_{i=1}^{d} |x_i - y_i|$$

Minkowski Distance (L^p-Norm): p=2

L^2-Norm or Euclidean Distance

$$\delta_2(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{d} |x_i - y_i|^2 \right)^{\frac{1}{2}} = \sqrt{\sum_{i=1}^{d} |x_i - y_i|^2}$$

Minkowski Distance (L^p-Norm): p=∞

L^∞-Norm or Chebyshev Distance

$$\delta_\infty(\mathbf{x}, \mathbf{y}) = \lim_{p\to\infty} \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}} =$$

$$= \max\{|x_1 - y_1|, |x_2 - y_2|, \ldots, |x_d - y_d|\}$$

Cosine Similarity

- A measure of similarity between two non-zero vectors of an inner product space

Cosine Similarity

- A measure of similarity between two non-zero vectors of an inner product space
- Measures the cosine of the angle between vectors

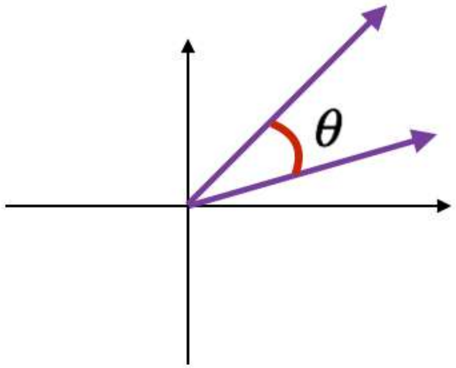
Cosine Similarity

- A measure of similarity between two non-zero vectors of an inner product space
- Measures the cosine of the angle between vectors
- It ranges between [-1,1]

Cosine Similarity

- A measure of similarity between two non-zero vectors of an inner product space
- Measures the cosine of the angle between vectors
- It ranges between [-1,1]
- It captures the orientation and not the magnitude

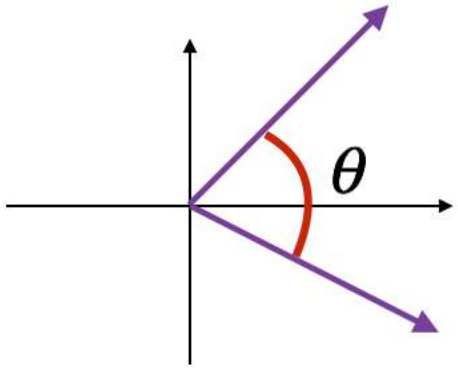
Cosine Similarity



θ is close to 0°

$\cos(\theta) \approx 1$

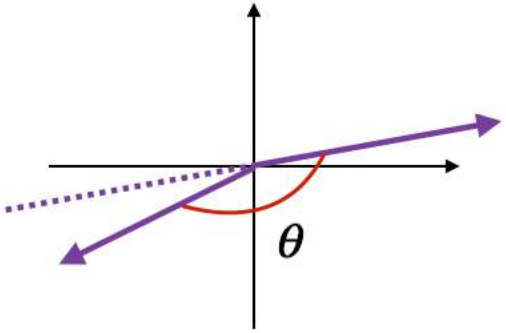
similar vectors



θ is close to 90°

$\cos(\theta) \approx 0$

orthogonal vectors

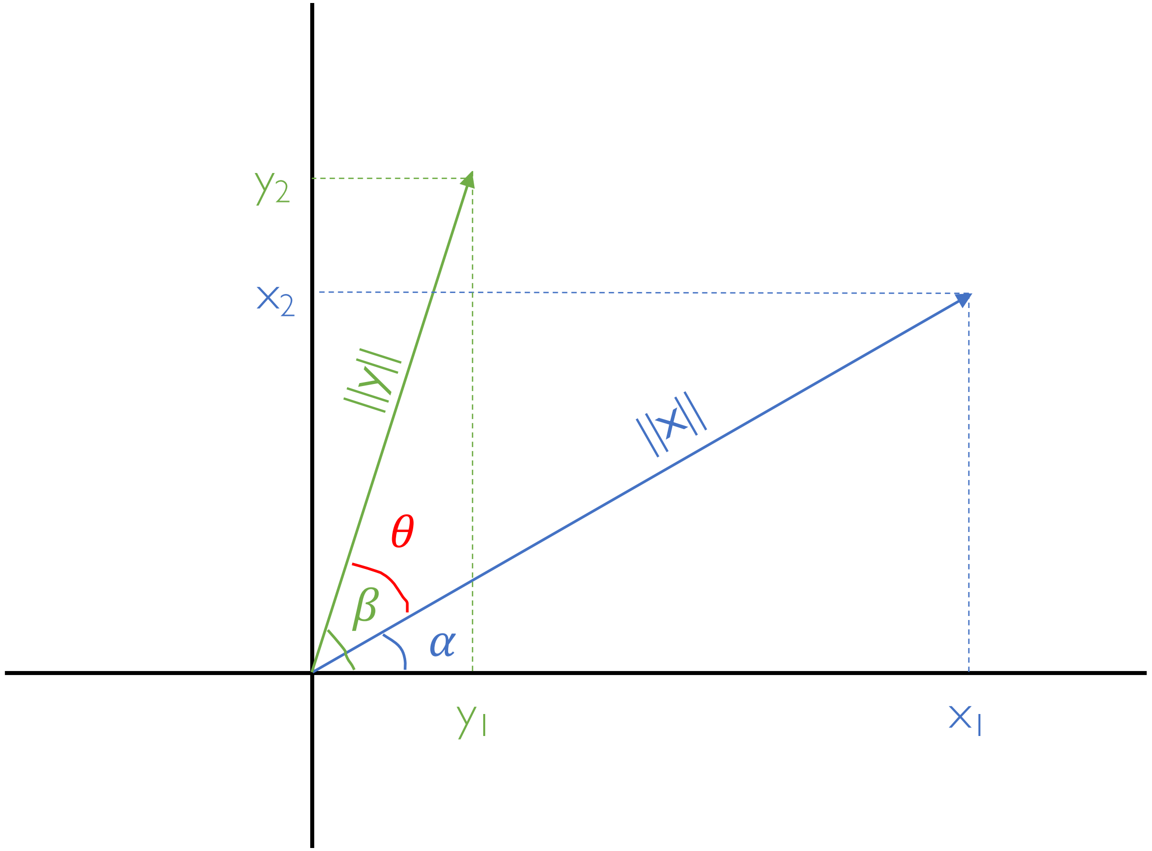


θ is close to 180°

$\cos(\theta) \approx -1$

opposite vectors

Cosine Similarity: 2-dimensional Case

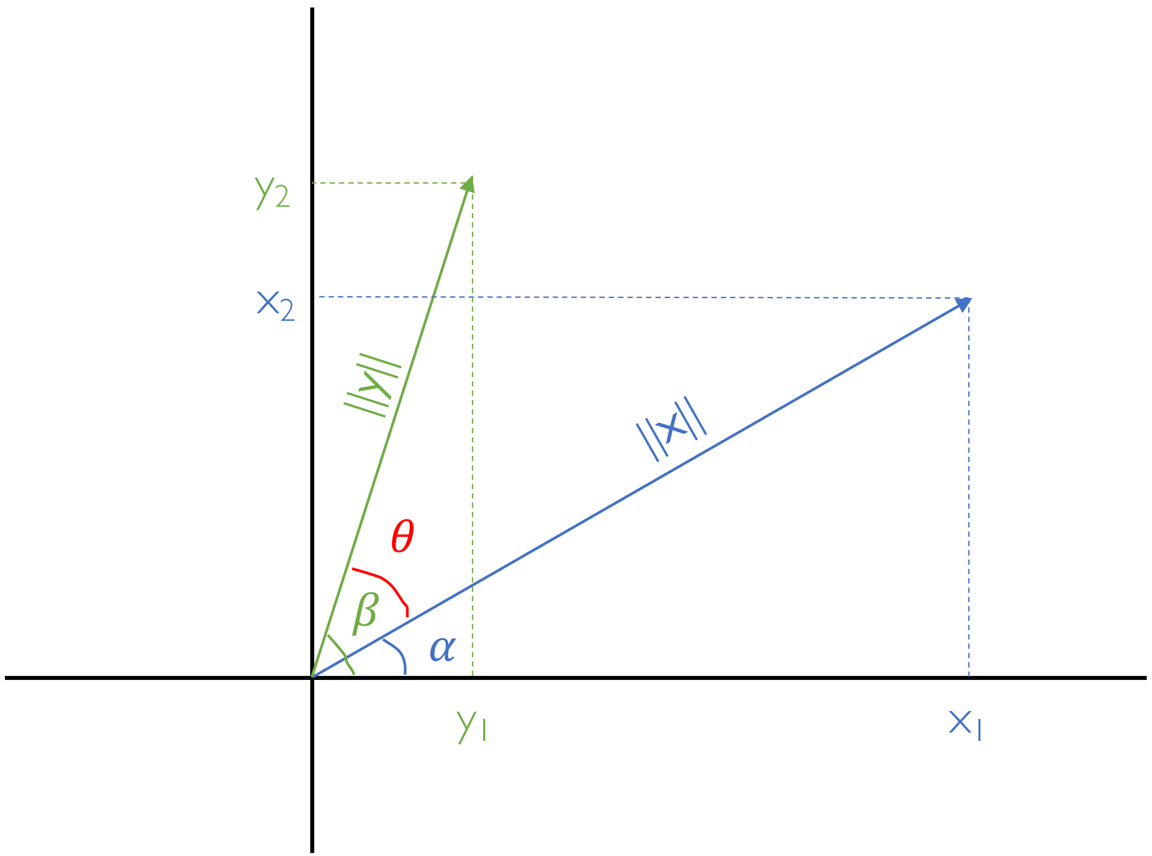


$$\theta = \beta - \alpha$$

$$x = (\underbrace{||x||\cos\alpha}_{x_1}, \underbrace{||x||\sin\alpha}_{x_2})$$

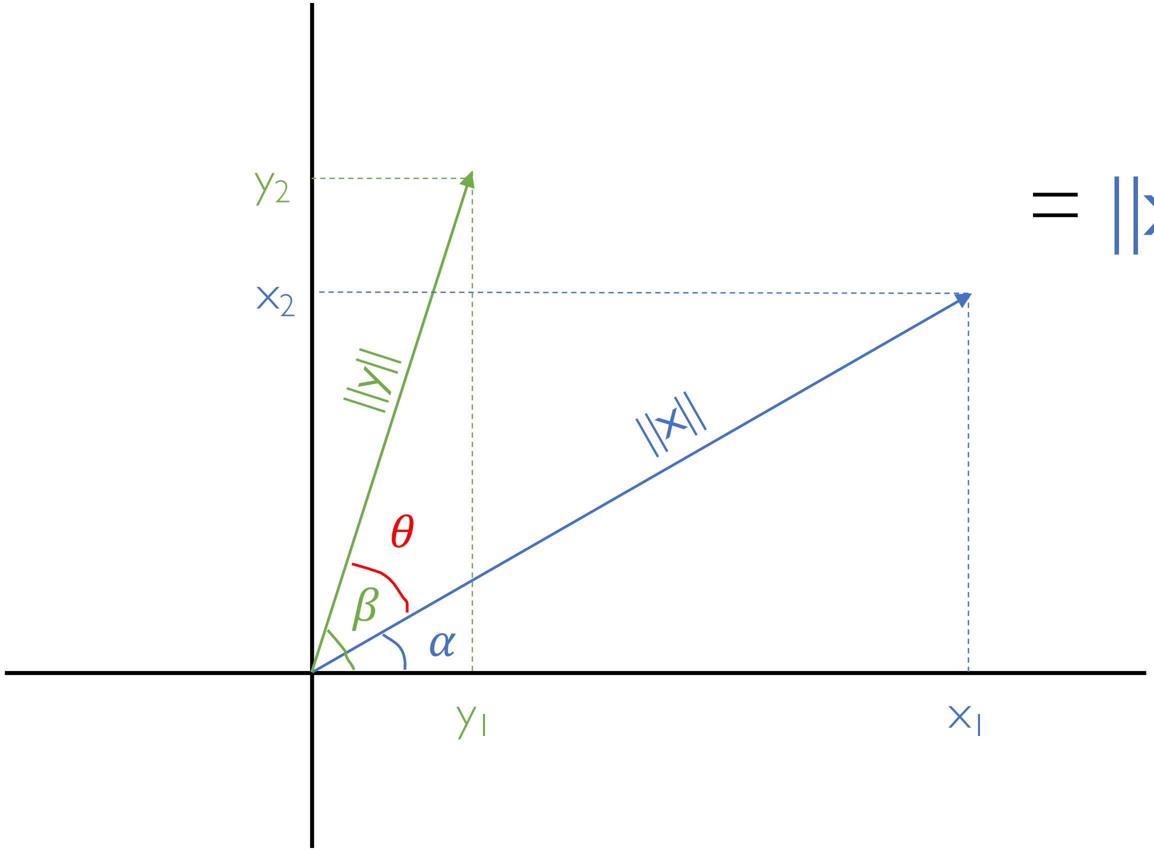
$$y = (\underbrace{||y||\cos\beta}_{y_1}, \underbrace{||y||\sin\beta}_{y_2})$$

Cosine Similarity: 2-dimensional Case



$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 =$

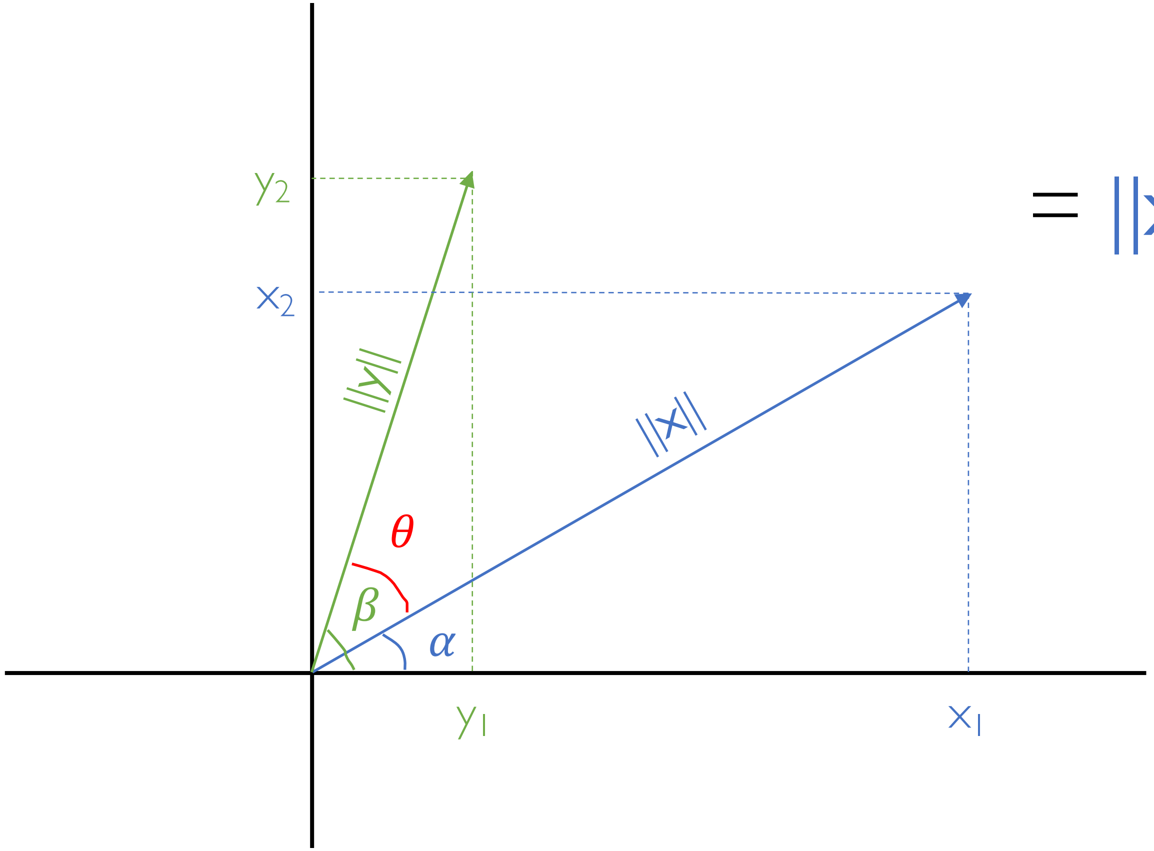
Cosine Similarity: 2-dimensional Case



$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 =$$

$$= ||\mathbf{x}|| \cos\alpha ||\mathbf{y}|| \cos\beta + ||\mathbf{x}|| \sin\alpha ||\mathbf{y}|| \sin\beta$$

Cosine Similarity: 2-dimensional Case

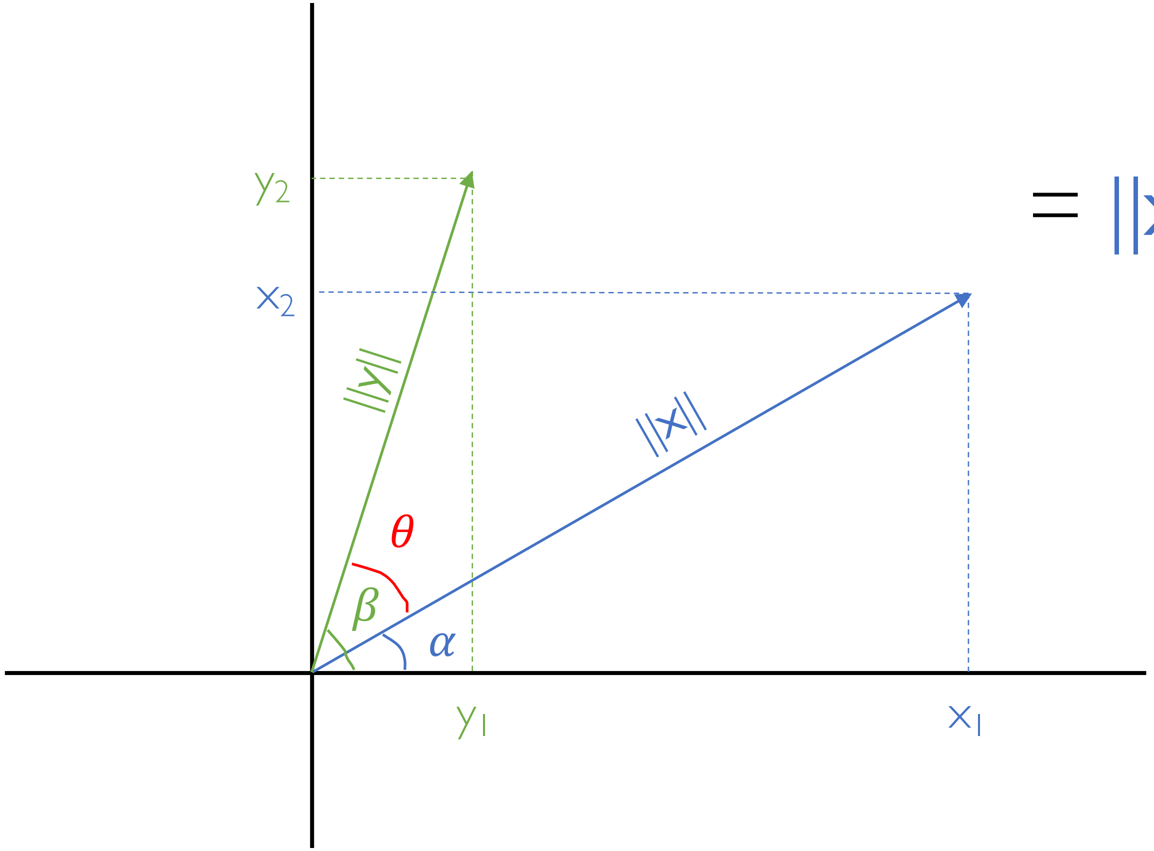


$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 =$$

$$= ||\mathbf{x}|| \cos\alpha ||\mathbf{y}|| \cos\beta + ||\mathbf{x}|| \sin\alpha ||\mathbf{y}|| \sin\beta$$

$$= ||\mathbf{x}|| ||\mathbf{y}|| (\cos\alpha \cos\beta + \sin\alpha \sin\beta)$$

Cosine Similarity: 2-dimensional Case

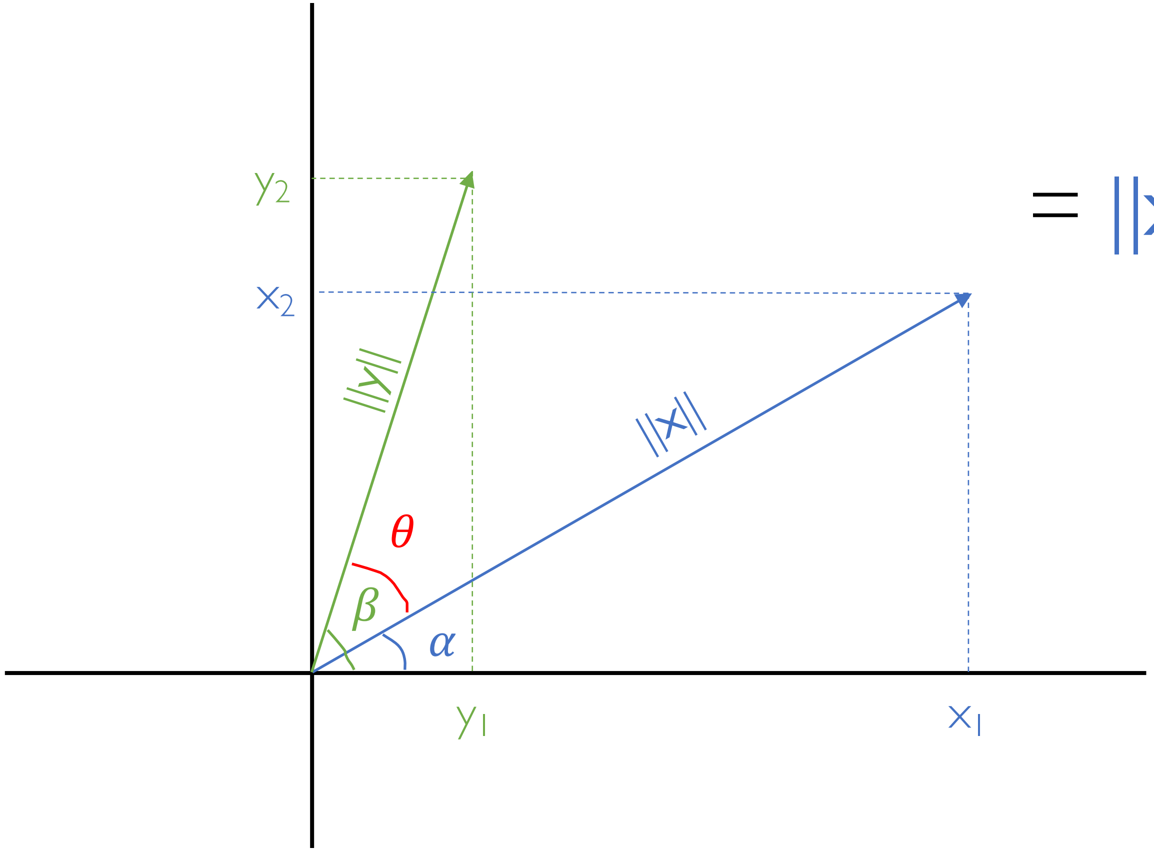


$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 =$$

$$= ||x||\cos\alpha||y||\cos\beta + ||x||\sin\alpha||y||\sin\beta$$

$$= ||x||||y||\underbrace{(\cos\alpha\cos\beta + \sin\alpha\sin\beta)}_{\cos(\underbrace{\beta - \alpha}_{\theta})}$$

Cosine Similarity: 2-dimensional Case



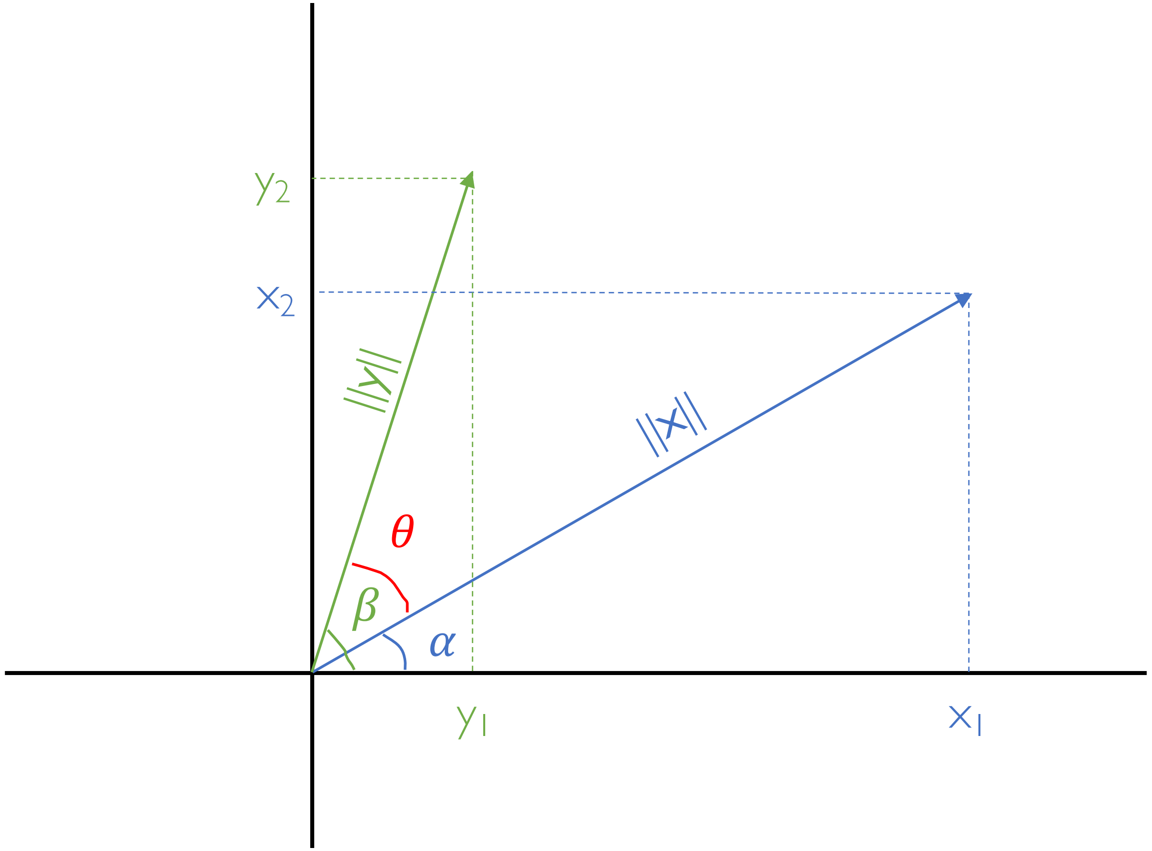
$$\mathbf{x}\cdot\mathbf{y} = x_1 y_1 + x_2 y_2 =$$

$$= ||x||\cos\alpha||y||\cos\beta + ||x||\sin\alpha||y||\sin\beta$$

$$= ||x||||y||\underbrace{(\cos\alpha\cos\beta + \sin\alpha\sin\beta)}_{\cos(\underbrace{\beta - \alpha}_{\theta})}$$

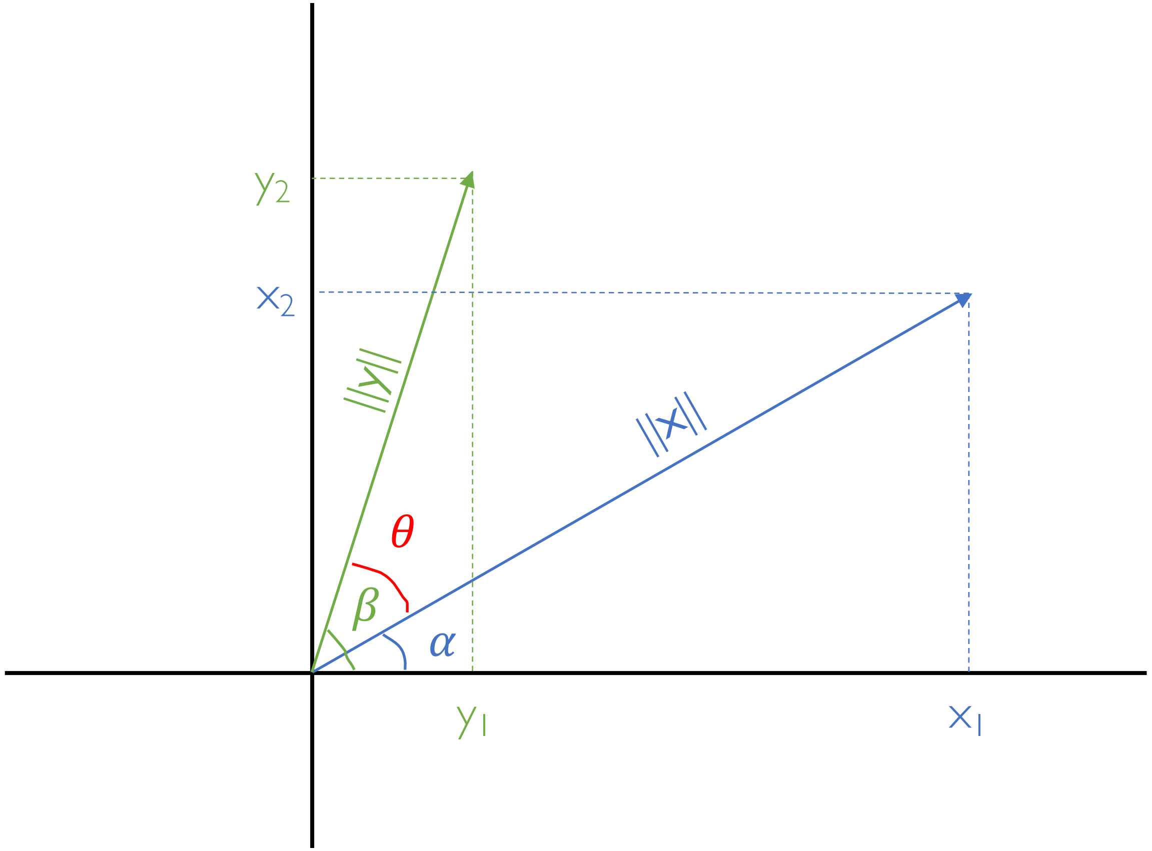
$$\boxed{\mathbf{x}\cdot\mathbf{y} = ||x||||y||\cos\theta}$$

Cosine Similarity: 2-dimensional Case



$$x \cdot y = ||x|| \, ||y|| \cos\theta$$

Cosine Similarity: 2-dimensional Case



$$x \cdot y = ||x||\,||y|| \cos\theta$$

$$\boxed{\cos\theta = x \cdot y / ||x||\,||y||}$$

Cosine Similarity: *d*-dimensional Case

- Computed as in the case of 2-dimensional vectors
- If two *d*-dimensional vectors are not collinear then they span a 2-dimensional plane $E \subset \mathbb{R}^d$
- This plane E inherits the dot product in $\mathbb{R}^d$ and so becomes an ordinary Euclidean plane
- The angles in this plane are related to the dot product as they are in 2-dimensional vector geometry

Jaccard Index (Coefficient)

Measures similarity between finite sample sets

Jaccard Index (Coefficient)

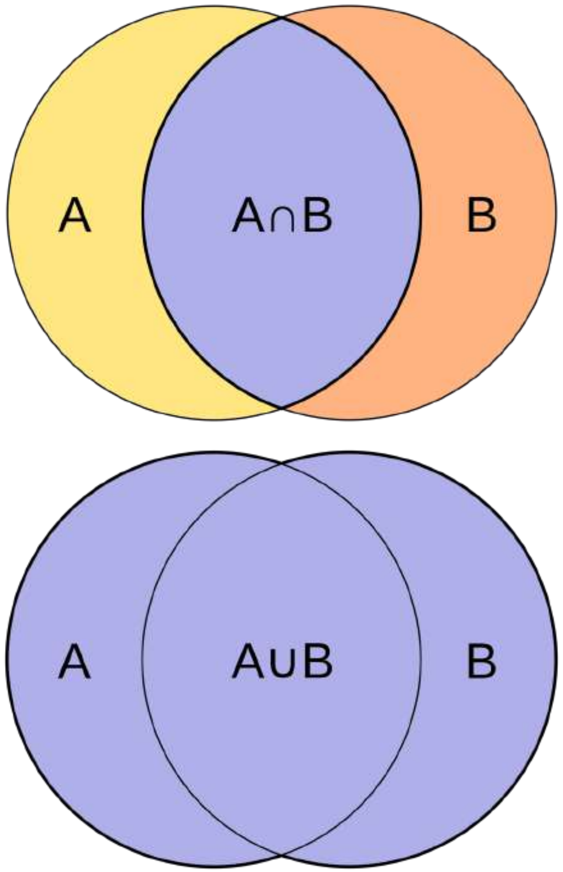
Measures similarity between finite sample sets

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

$$J(A,B) = 1 \text{ if } A = B = \emptyset$$

$$0 \leq J(A,B) \leq 1$$

Jaccard Index (Coefficient): Interpretation



source: [Wikipedia]

Jaccard Distance

Complementary to the Jaccard coefficient

$$\delta_J(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$

This distance is a **metric** on the collection of all finite sets

Take-Home Message of Today

- Clustering is an unsupervised learning technique to group "similar" data objects together

Take-Home Message of Today

- Clustering is an unsupervised learning technique to group "similar" data objects together
- Depends on:
  - object representation
  - similarity measure

Take-Home Message of Today

- Clustering is an unsupervised learning technique to group "similar" data objects together
- Depends on:
  - object representation
  - similarity measure
- Harder when data dimensionality gets large (curse of dimensionality)

Take-Home Message of Today

- Clustering is an unsupervised learning technique to group "similar" data objects together
- Depends on:
  - object representation
  - similarity measure
- Harder when data dimensionality gets large (curse of dimensionality)
- Number of output clusters is part of the problem itself!