Distance & Similarity

Boston University CS 506 - Lance Galletti

Data

$$\begin{array}{l}\textbf{n}\text{ data}\\ \text{points}\end{array}\left\{\underbrace{\begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & x_{ij} & \cdots & x_{im} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{nm} \end{pmatrix}}_{\textbf{m}\text{ features}}\right.$$

Feature Space

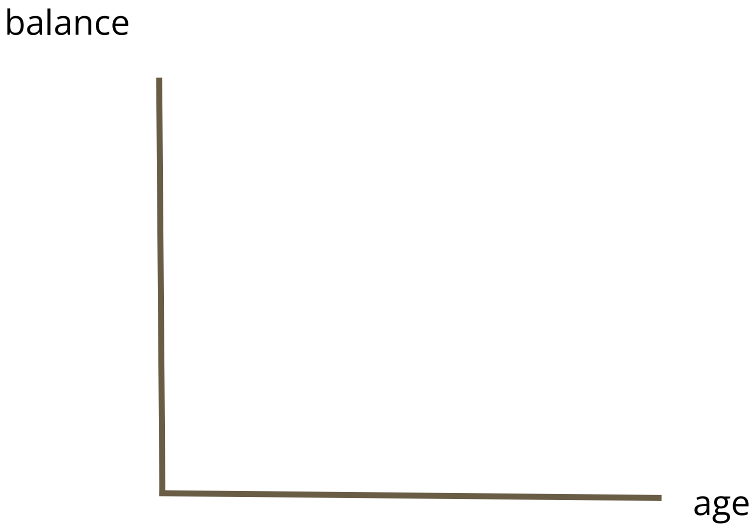
From our data we can generate a **feature space** of all possible values for the set of features in our data.

name	age	balance
Jane	25	150
John	30	100

Feature Space

From our data we can generate a **feature space** of all possible values for the set of features in our data.

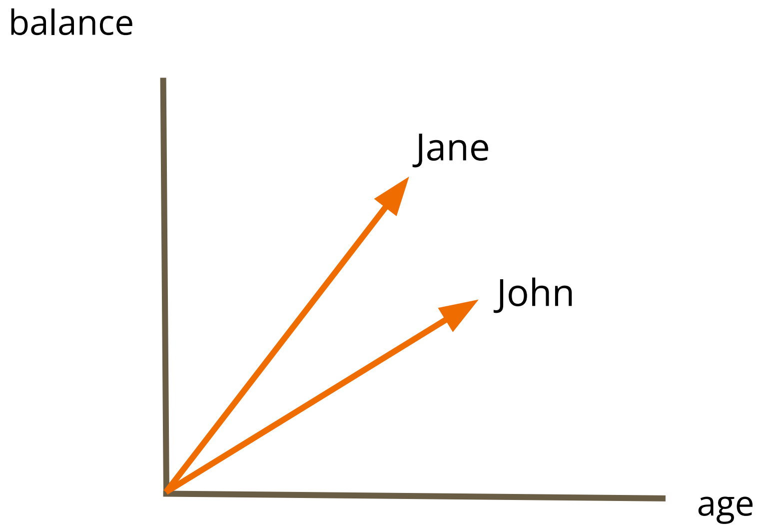
name	age	balance
Jane	25	150
John	30	100



Feature Space

From our data we can generate a **feature space** of all possible values for the set of features in our data.

name	age	balance
Jane	25	150
John	30	100



Our feature space is the Euclidean plane

Distance

In order to uncover interesting structure from our data, we need a way to **compare** data points.

A **dissimilarity function** is a function that takes two objects (data points) and returns a **large value** if these objects are **dissimilar**.

A special type of dissimilarity function is a **distance** function

Distance

d is a distance function if and only if:

- $d(i, j) = 0$ if and only if $i = j$
- $d(i, j) = d(j, i)$
- $d(i, j) \leq d(i, k) + d(k, j)$

We don't **need** a distance function to compare data points, but why would we prefer using a distance function?

Minkowski Distance

For **x**, **y** points in **d**-dimensional real space

I.e. **x = [x_1 , … , x_d]** and **y = [y_1 , … , y_d]**

p ≥ 1

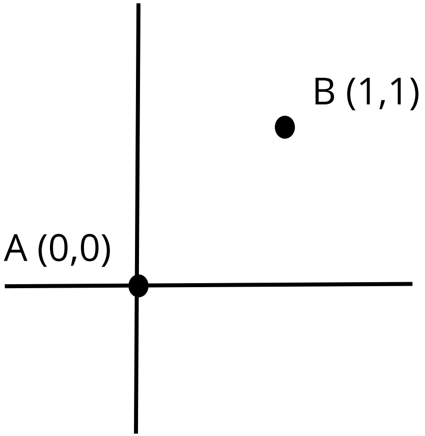
$$L_p(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

When **p** = 2 -> Euclidean Distance

When **p** = 1 -> Manhattan Distance

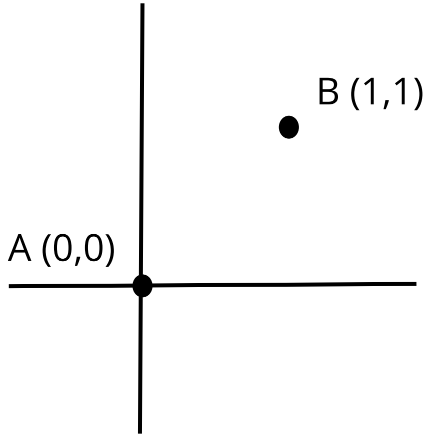
Example

d = 2



Example

d = 2

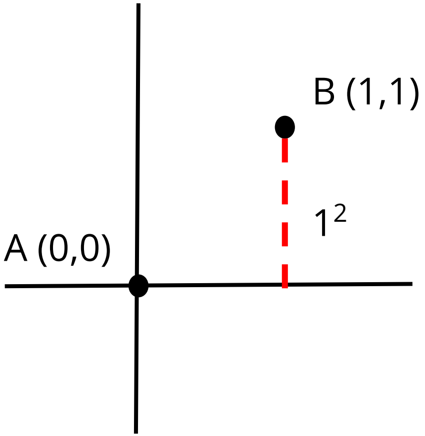


p = 2

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

Example

d = 2

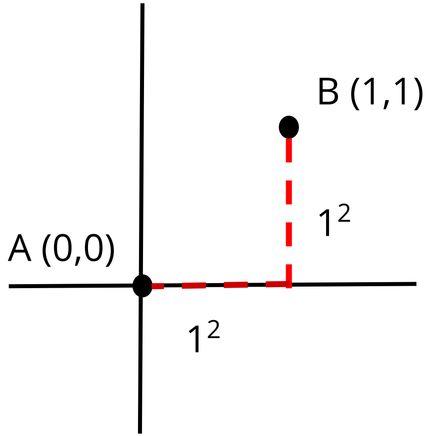


p = 2

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

Example

d = 2

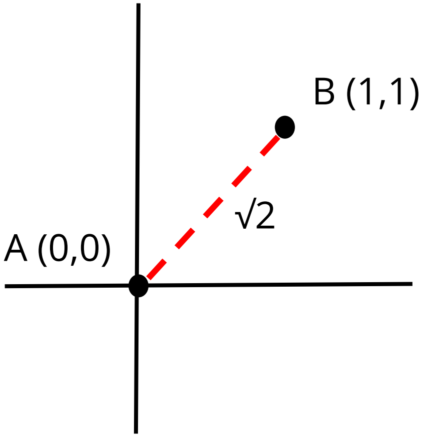


p = 2

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

Example

d = 2

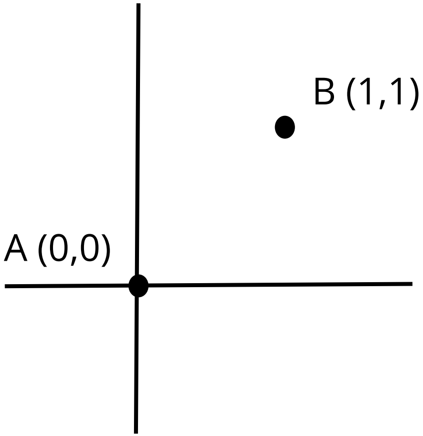


p = 2

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

Example

d = 2

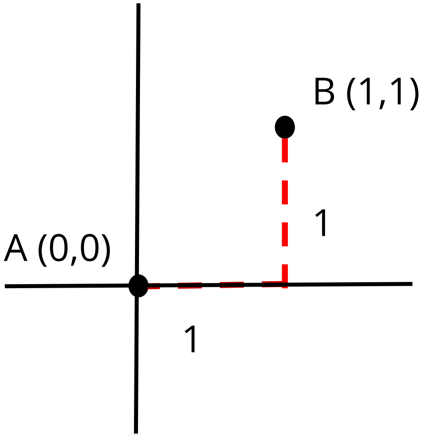


p = 1

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

Example

d = 2

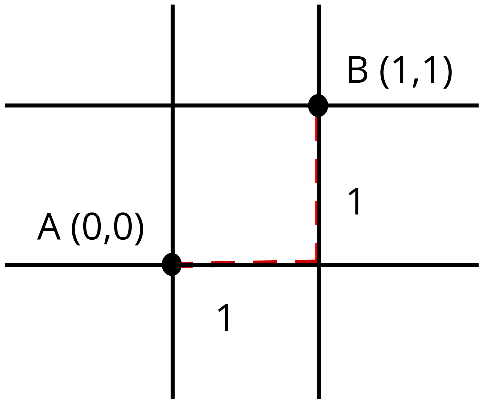


p = 1

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

Example

d = 2

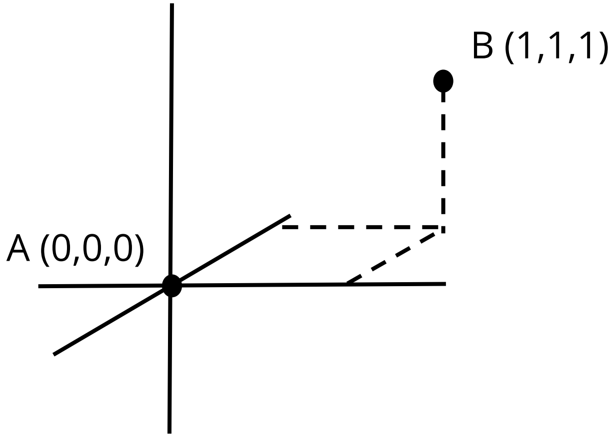


p = 1

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

Example

d = 3

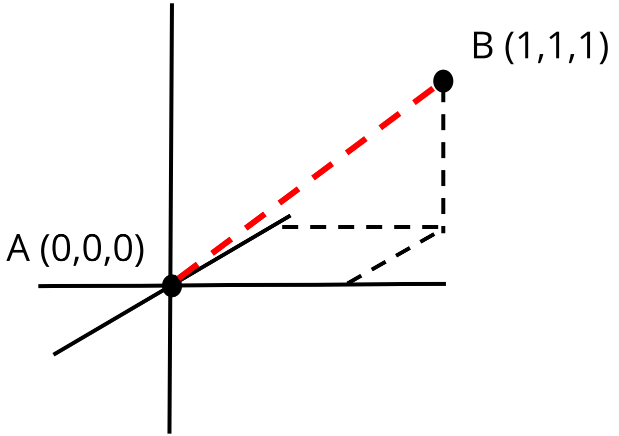


p = 2

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

Example

d = 3

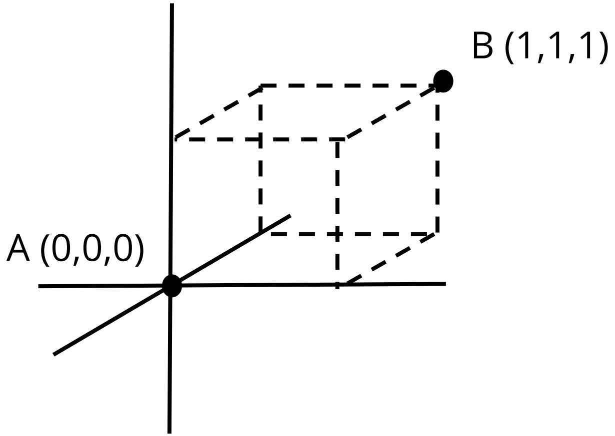


p = 2

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

Example

d = 3

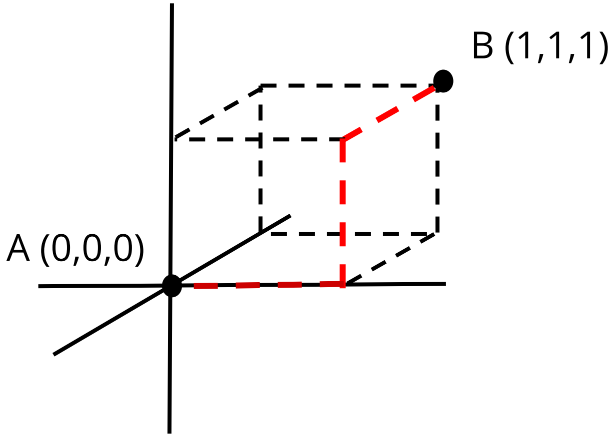


p = 1

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

Example

d = 3



p = 1

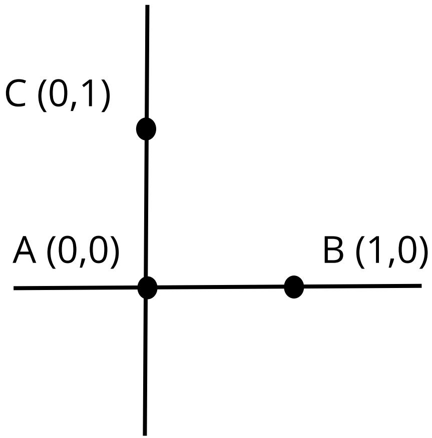
$$L_p(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{\frac{1}{p}}$$

Minkowski Distance

Is $\mathbf{L_p}$ a distance function when $\mathbf{0 < p < 1}$?

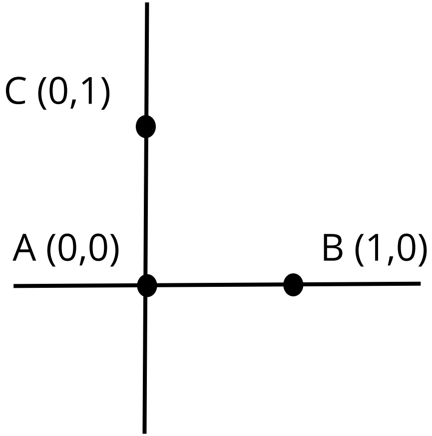
Minkowski Distance

Is $\mathbf{L_p}$ a distance function when $\mathbf{0 < p < 1}$?



Minkowski Distance

Is $\mathbf{L_p}$ a distance function when $\mathbf{0 < p < 1}$?

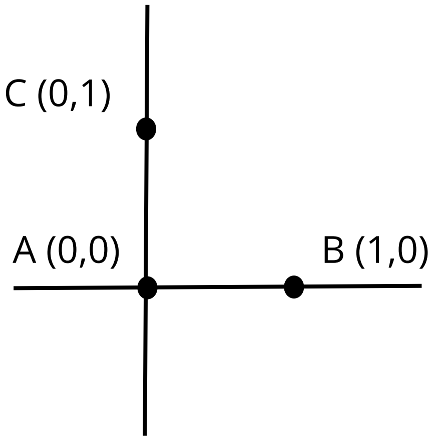


D(B,A) = D(A, C) = 1

D(B, C) = $\mathbf{2^{1/p}}$

Minkowski Distance

Is $\mathbf{L_p}$ a distance function when $\mathbf{0 < p < 1}$?



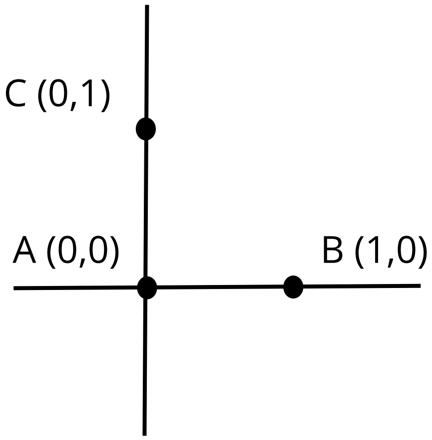
$\mathbf{D(B,A) + D(A, C) = 2}$

$\mathbf{D(B, C) = 2^{1/p}}$

But... if $\mathbf{p < 1}$ then $\mathbf{1/p > 1}$

Minkowski Distance

Is $\mathbf{L_p}$ a distance function when $\mathbf{0 < p < 1}$?



D(B,A) + D(A, C) = 2

D(B, C) = $\mathbf{2^{1/p}}$

So **D(B, C) > D(B, A) + D(A, C)** which violates the triangle inequality

Cosine Similarity

A **similarity** function is a function that takes two objects (data points) and returns a **large value** if these objects are **similar**.

$$\mathbf{s(x, y) = cos(}\boldsymbol{\theta}\mathbf{)}$$

where $\boldsymbol{\theta}$ is the angle between **x** and **y**

Cosine Similarity

To get a corresponding **dissimilarity** function, we can usually try

$$\mathbf{d(x, y) = 1 / s(x, y)}$$

or

$$\mathbf{d(x, y) = k - s(x, y)} \text{ for some } \mathbf{k}$$

Here, we can use

$$\mathbf{d(x, y) = 1 - s(x, y)}$$

Cosine Similarity

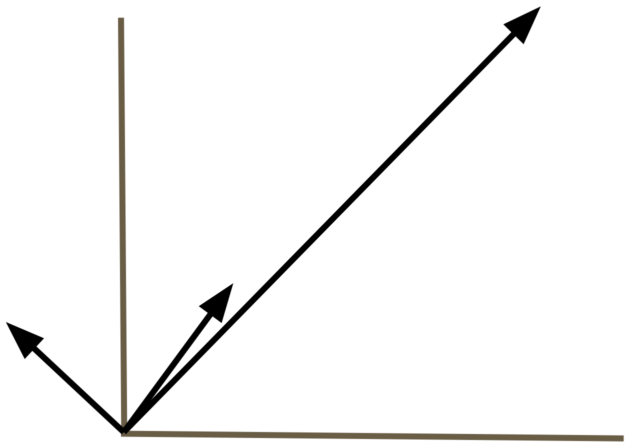
When should you use **cosine (dis)similarity** over **euclidean distance**?

When **direction** matters more than **magnitude**

Cosine Similarity

When should you use **cosine (dis)similarity** over **euclidean distance**?

When **direction** matters more than **magnitude**

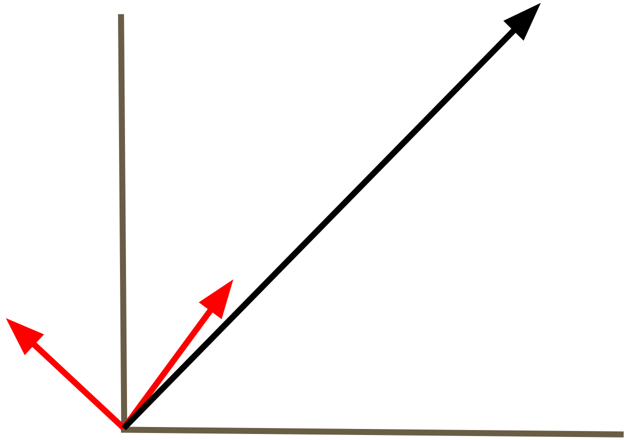


Cosine Similarity

When should you use **cosine (dis)similarity** over **euclidean distance**?

When **direction** matters more than **magnitude**

Close under
Euclidean distance

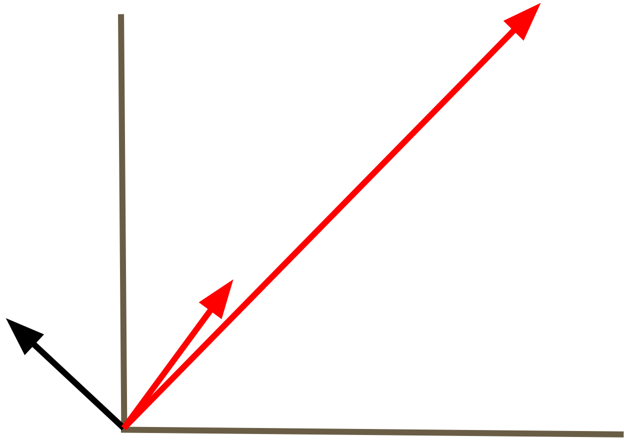


Cosine Similarity

When should you use **cosine (dis)similarity** over **euclidean distance**?

When **direction** matters more than **magnitude**

Close under Cosine Similarity



Jaccard Similarity

How similar are the following documents?

	w_1	w_2	…	w_d
x	1	0	…	1
y	1	1	…	0

Jaccard Similarity

One way is to use the Manhattan distance which will return the size of the set difference

	w_1	w_2	...	w_d
x	1	0	...	1
y	1	1	...	0

$$L_1(x, y) = \sum_{i=1}^{d} |x_i - y_i|$$

Jaccard Similarity

One way is to use the Manhattan distance which will return the size of the set difference

	w_1	w_2	...	w_d
x	1	0	...	1
y	1	1	...	0

$$L_1(x, y) = \sum_{i=1}^{d} |x_i - y_i|$$

Will only be 1 when $x_i \neq y_i$

Jaccard Similarity

But how can we distinguish between these two cases?

	w_1	w_2	...	w_{d-1}	w_d
x	1	1	1	0	1
y	1	1	1	1	0

Only differ on the last two words

	w_1	w_2
x	0	1
y	1	0

Completely different

Jaccard Similarity

But how can we distinguish between these two cases?

	w_1	w_2	...	w_{d-1}	w_d
x	1	1	1	0	1
y	1	1	1	1	0

Only differ on the last two words

	w_1	w_2
x	0	1
y	1	0

Completely different

Both have Manhattan distance of 2

Jaccard Similarity

We need to account for the size of the intersection!

$$JSim(x, y) = \frac{|x \cap y|}{|x \cup y|}$$

$$JDist(x, y) = 1 - \frac{|x \cap y|}{|x \cup y|}$$

A quick note on Norms

- Distance from the origin
  - Minkowski Distance <=> Lp Norm
  - Not all distances can create a norm

- Notion of size
- Has the following properties:
  - $p(x + y) \leq p(x) + p(y)$
  - $p(ax) = |a|p(x)$
  - $p(x) = 0$ iff $x = 0$
  - $p(x) \geq 0$ for all x

Implement these distance functions in the CS506 python package