Distance & Similarity

Boston University CS 506 - Lance Galletti

Features/characteristics

Refund	Marital Status	Income	Age

Data point

Refund	Marital Status	Income	Age
1	Single	125k	25

Refund	Marital Status	Income	Age
1	Single	125k	25
0	Married	100k	27

Refund	Marital Status	Income	Age
1	Single	125k	25
0	Married	100k	27
0	Single	70k	22

Refund	Marital Status	Income	Age
1	Single	125k	25
0	Married	100k	27
0	Single	70k	22
1	Married	120k	30
0	Divorced	90k	28
0	Married	60k	37
1	Divorced	220k	24
0	Single	85k	23
0	Married	75k	23
0	Single	90k	26

Data

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

Data

Attribute / feature j

n data points

Data point i

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

Data

Attribute / feature j

Feature j of data point i

n data points

Data point i

$$\underbrace{\begin{pmatrix} x_{11} & \cdots & x_{1j} & \cdots & x_{1m} \\ \vdots & \ddots & \vdots & & \vdots \\ x_{i1} & \cdots & \boxed{x_{ij}} & \cdots & x_{im} \\ \vdots & & \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nj} & \cdots & x_{nm} \end{pmatrix}}_{\textbf{m}\text{ features}}$$

m features

Feature Space

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

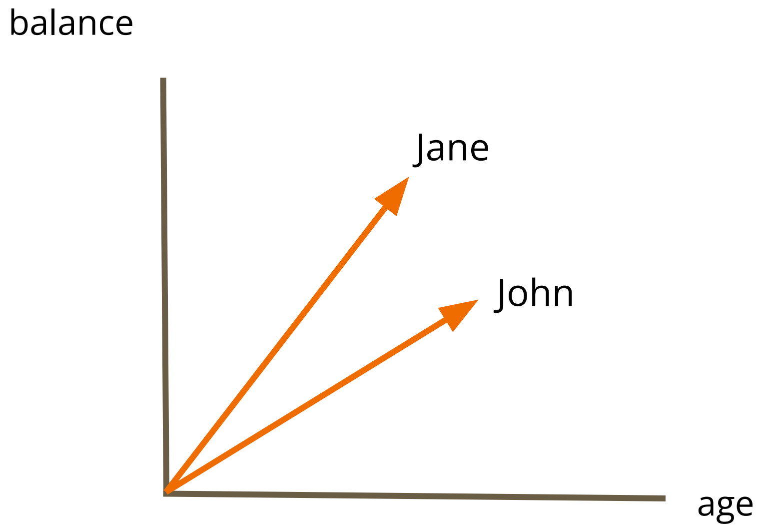
the n-dimensions where your variables live (not including a target variable, if it is present)

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



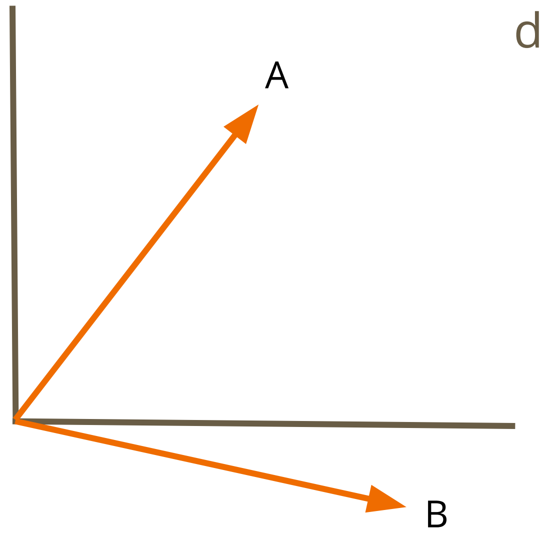
Our feature space is the Euclidean plane

Dissimilarity

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

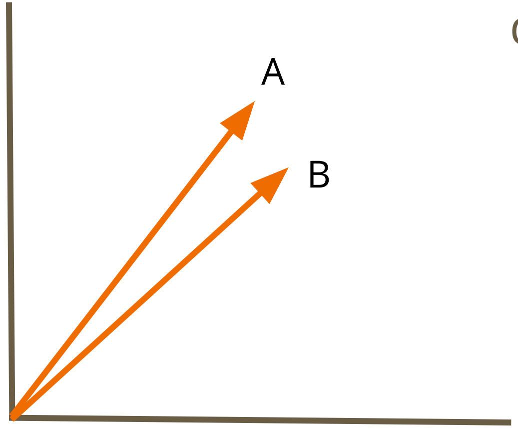
A **dissimilarity function** is a function that takes two objects (data points) and returns a **large value** if these objects are **dissimilar**.

Dissimilarity



dissim(A, B) is large

Dissimilarity



dissim(A, B) is small

Distance

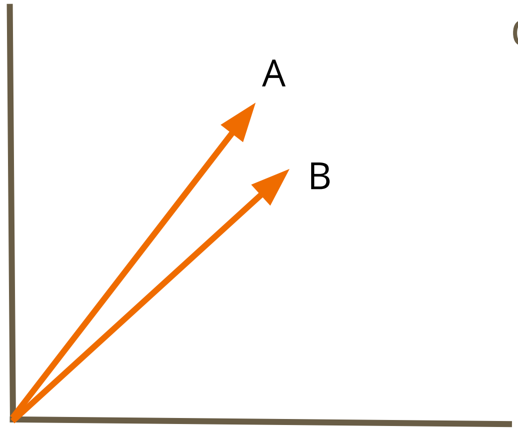
A special type of dissimilarity function is a **distance** function

d is a distance function if and only if:

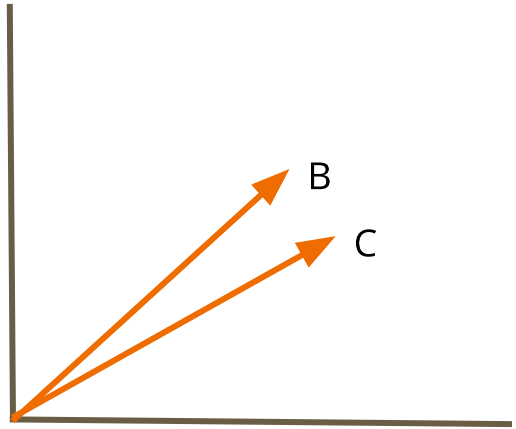
- $d(i, j) = 0$ if and only if $i = j$
- $d(i, j) = d(j, i)$
- $d(i, j) \leq d(i, k) + d(k, j)$ **<- meaning that d is the shortest distance**

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

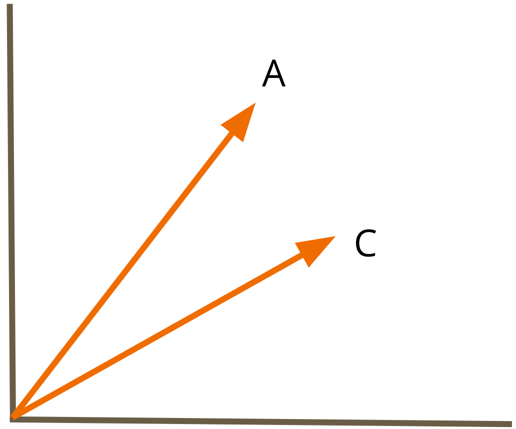
Why prefer distance function to the dissimilarity function?
-> generalizable/dijestable
-> Distance functions satisfy the metric properties (non-negativity, identity, symmetry, and triangle inequality), ensuring reliable comparisons.
Distance functions provide a clear spatial interpretation (e.g., Euclidean distance can be visualized in space).
Dissimilarity measures are often problem-specific, making them harder to generalize.



dissim(A, B) is small



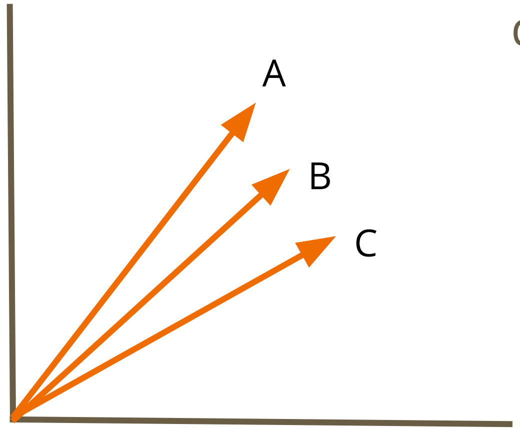
dissim(B, C) is small



dissim(A, C) not necessarily small

dissimilarity function
if dissim(A,B) is small and dissim(B,C) is small but dissim(A,C) may not be small
- how different

distance function
if d(A,B) is small and d(B,V) is small, then d(A,C) is small
how far



d(A, B) is small

d(B, C) is small

Triangle inequality guarantees d(A, C) small

d(A,B)+d(B+C)>d(A,C)
small+small>much small

Minkowski Distance

d features = 2 or 3 or d dimensional space

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

summing every feature i
for each feature i, calculating the difference between data points

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

p ≥ 1

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

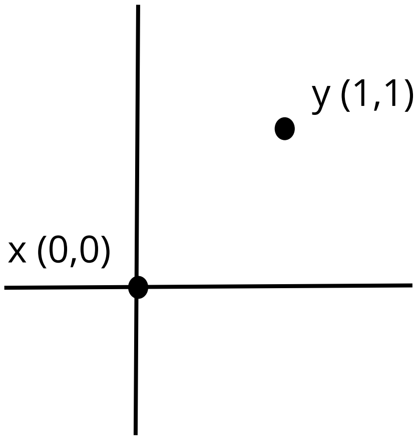
When **p** = 2 -> Euclidean Distance

if d = 2: (lx_1-y_1l^p)+lx_2-y_2l^p)^(1/p)
if d = 3: (lx_1-y_1l^p)+lx_2-y_2l^p+lx_3-y_3l^p)^(1/p)

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

Example

d = 2

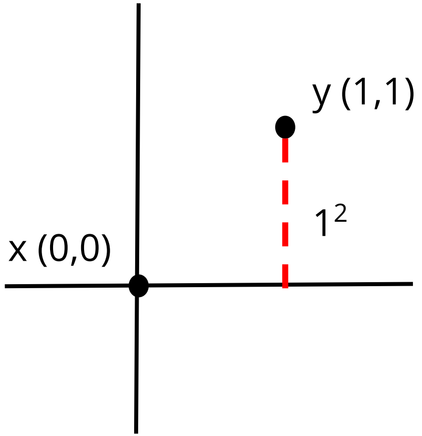


p = 2

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

Example

d = 2

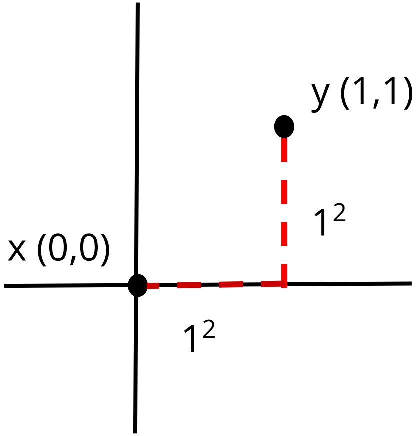


p = 2

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

Example

d = 2

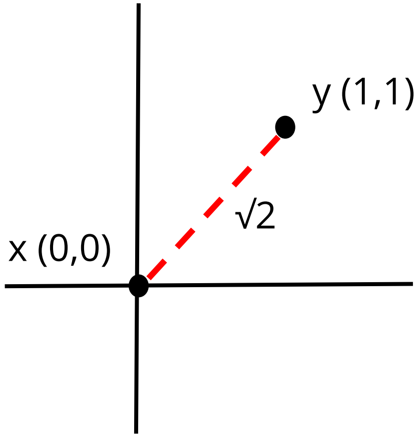


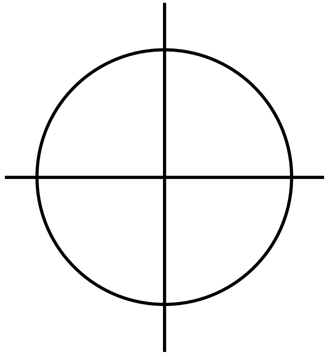
p = 2

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

Example

d = 2





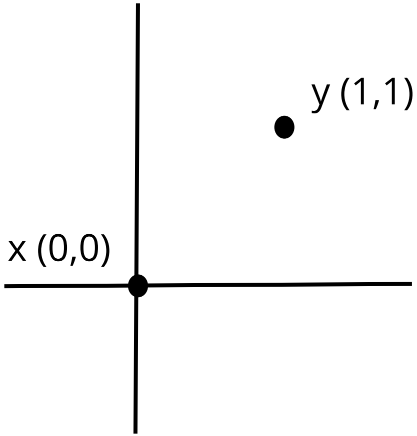
on the Euclidian distance, the distance is calculated with the shape of circle

p = 2

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

Example

d = 2

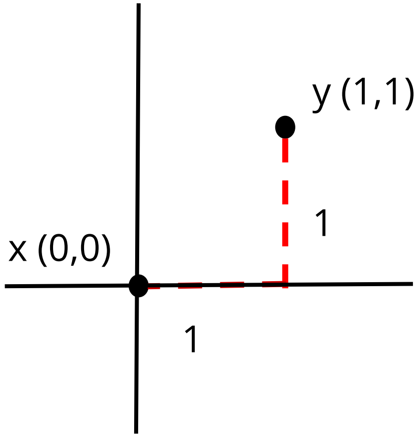


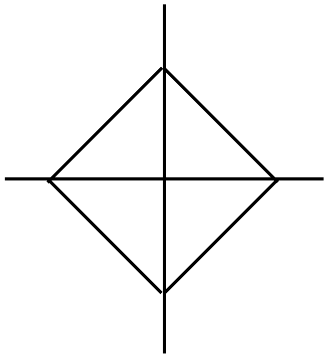
p = 1

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

Example

d = 2





if p = 1, cant go on the diagonal line

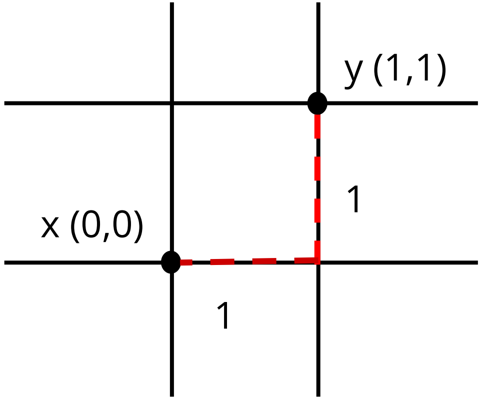
on the Manhattan distance, the distance is calculated within the shape of diamond

p = 1

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

Example

d = 2

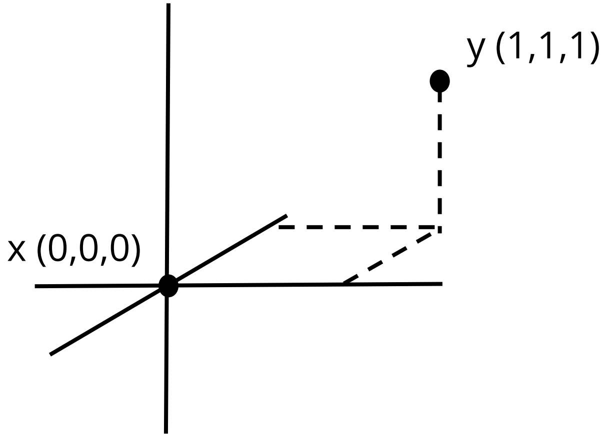


p = 1

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

Example

d = 3

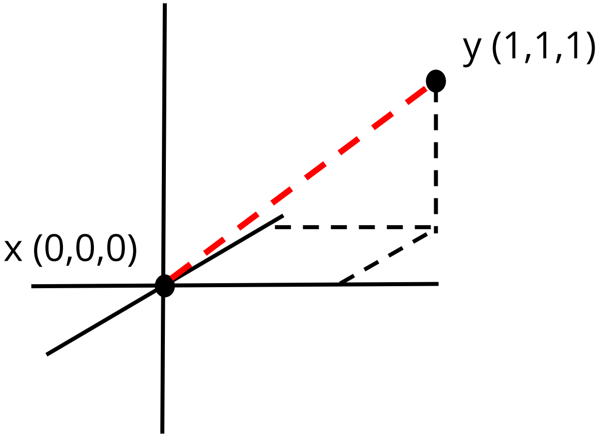


p = 2

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

Example

d = 3

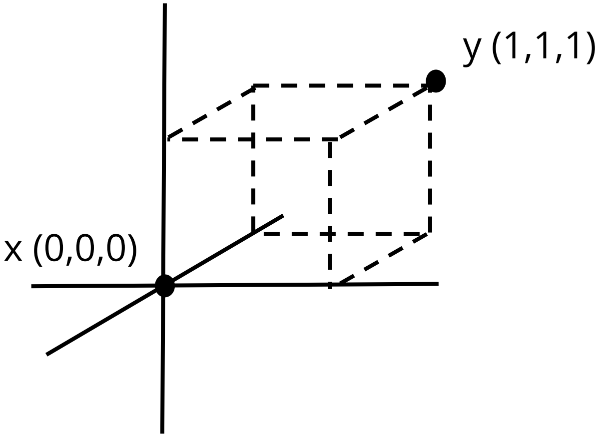


p = 2

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

Example

d = 3

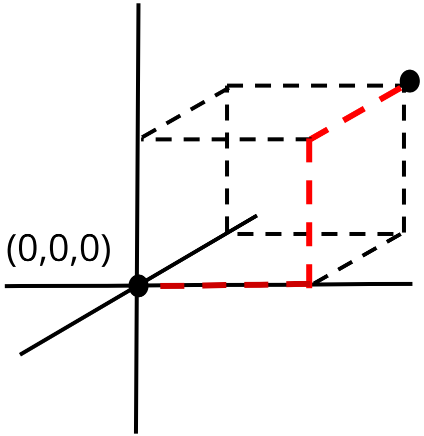


p = 1

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

Example

d = 3



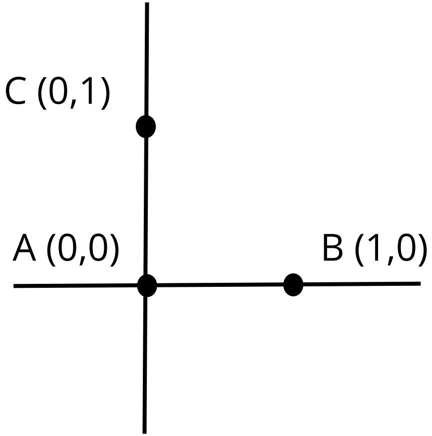
p = 1

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

Minkowski Distance

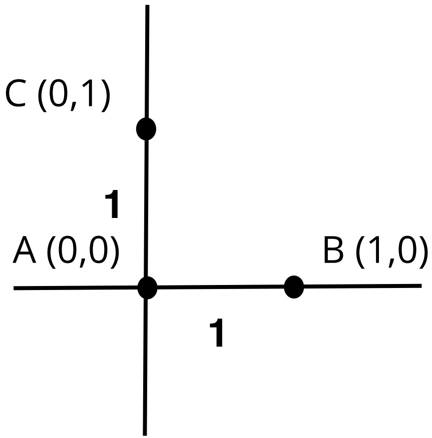
Is L_p a distance function when $0 < p < 1$?

L_p is not a distance function if 0<p<1



Minkowski Distance

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

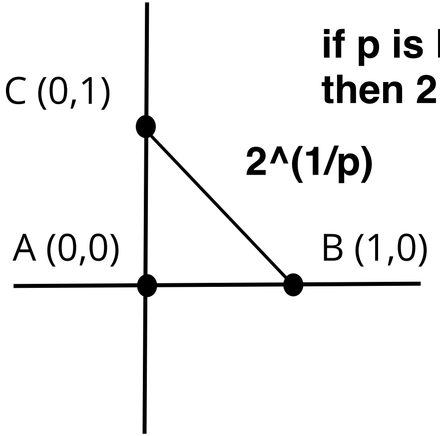


D(B,A) = D(A, C) = 1

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

Minkowski Distance

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



if p is less than 1, then 1/p가 커짐
then 2^(1/p) will be greater than 2

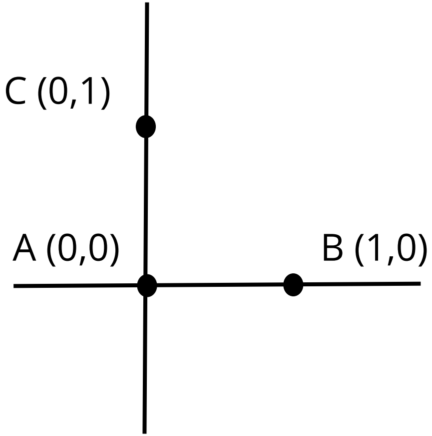
D(B,A) + D(A, C) = 2

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

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) = $\mathbf{2^{1/p}}$

Trianble inequality:
for each side of a triangle a,b,c:
a+b>c
a+c>b
b+c>a

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

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

counting mismatch -> whether the word appears in two documents or not

	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

giving the context whereas Manhattan distance does not account for the context

We need to account for the size of the intersection!

Given two documents x and y:

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

Jaccard Similarity

We need to account for the size of the intersection!

Given two documents x and y:

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

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

Here, x is the set of words (not the binary vector representation)

Jaccard Similarity

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

	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

What is the jaccard distance in each?

Jaccard Similarity

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

Here, x is the set of words (not the binary vector representation)

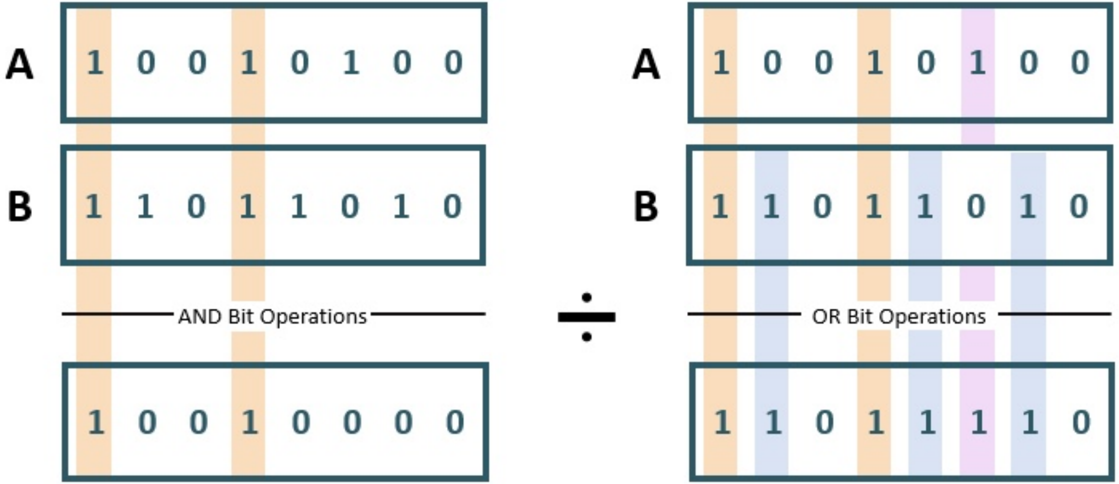
Counting similarity:

$$J(doc_1, doc_2) = \frac{\{'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'\} \bigcap \{'data', 'is', 'a', 'new', 'oil'\}}{\{'data', 'is', 'the', 'new', 'oil', 'of', 'digital', 'economy'\} \bigcup \{'data', 'is', 'a', 'new', 'oil'\}}$$

$$= \frac{\{'data', 'is', 'new', 'oil'\}}{\{'data', 'a', 'of', 'is', 'economy', 'the', 'new', 'digital', 'oil'\}}$$

$$= \frac{4}{9} = 0.444$$

Jaccard similarity for 1s and 0s:



$$\text{Jaccard Similarity} = \frac{\|A \cap B\|}{\|A \cup B\|} = \frac{2}{6} = 0.33$$

$$\text{Jaccard Distance} = 1 - \text{Jaccard Similarity} = 1 - 0.33 = 0.67$$

Cosine Similarity

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

large = being similar
small = not being similar

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

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

Cosine Similarity

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

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

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

two proportional vectors have a cosine similarity of: 1

직각

two orthogonal vectors have a similarity of: 0

two opposite vectors have a similarity of: - 1

Vectors that have the same magnitude but point in opposite directions

Cosine Similarity

작으면 dissimilar

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

$$d(x, y) = 1 / s(x, y)$$

or

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

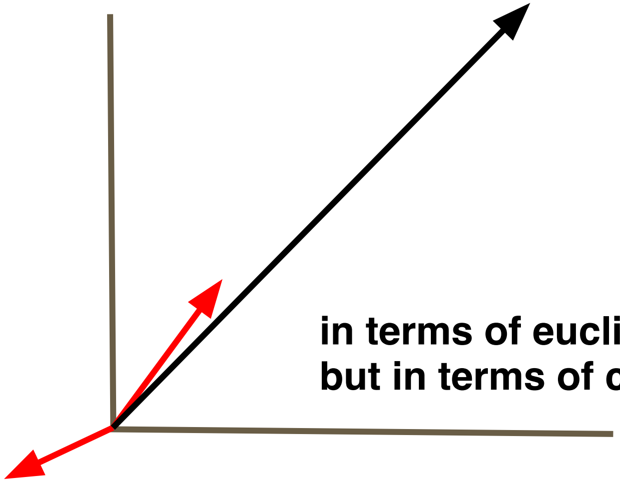
Here, we can use

$$d(x, y) = 1 - s(x, y)$$

Cosine Similarity

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

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



in terms of euclidean distance, it might be large
but in terms of cosine similarity, it might be similar

Close under
Euclidean distance

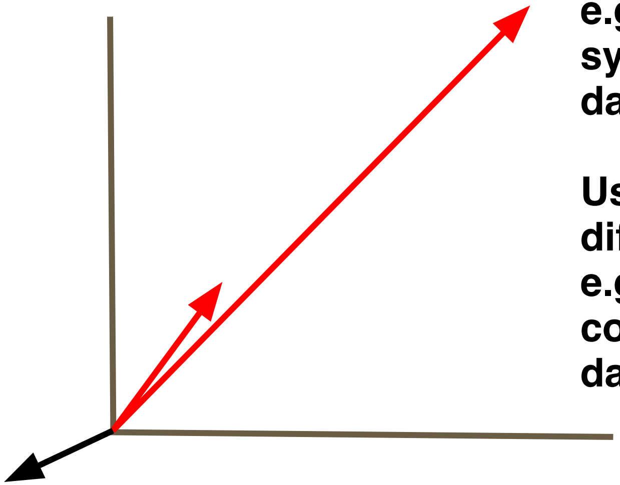
Cosine Similarity

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

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

Use Cosine Similarity when direction matters more than magnitude, e.g., text similarity, recommendation systems, high-dimensional sparse data.

Use Euclidean Distance when absolute differences matter, e.g., spatial distances, image comparison, low-dimensional dense data.



Close under Cosine Similarity

A quick Note on Norms

$$d(A, B) = \|A - B\|$$

Size = Distance from the origin $$d(0, X) = \|X\|$$

- Minkowski Distance <=> Lp Norm
- Not all distances can create a Norm