K-Means III

Treatise on Distance Metric

M. R. Hasan

CSCE 411/811

Data Modeling for Systems Development

What We Will Cover

- Distance Metric

Distance Metric for Determining Similarity

K-Means

- The K-Means algorithm ***depends*** on how we define **distance** between samples.
- **Euclidean distance** is the most commonly used metric.

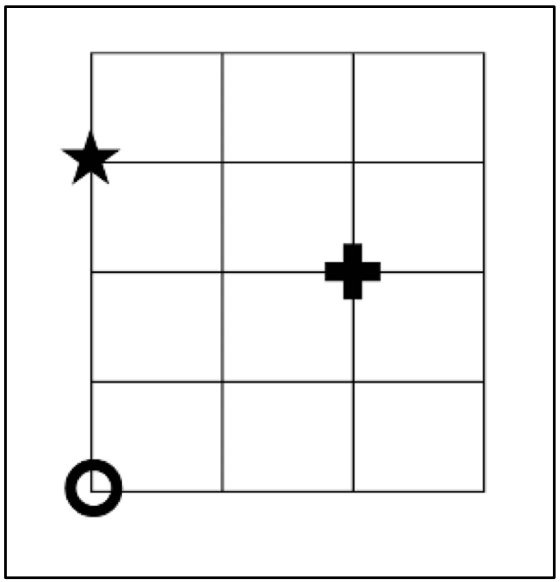
K-Means

- However, as we will see later that for **high-dimensional data** Euclidean distance may **not be a good measure** of "similarity".
- In general we should have **some flexibility** to define the distance of the data points to fit to the scenario at hand.

K-Means

- Let's discuss how **various distance measures arise**.
- We will illustrate various possible distance calculations by using a **2D feature vector**.
- We can represent the feature vectors using a **2D Cartesian coordinates**.
- In the example **3 data points** are shown: circle, star and cross.

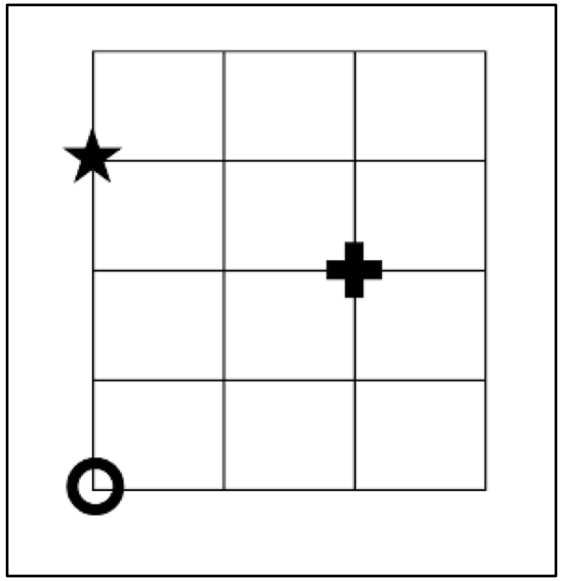
Is the circle ***closer*** to the cross or to the star?



K-Means

- **Euclidean distance** measure tells us that the distance between the circle and the cross is **2.83 units** ($\sqrt{2^2 + 2^2}$).
- However, the **star is at 3 units** distance from the circle.

Euclidean: The circle is closer to the **cross**.

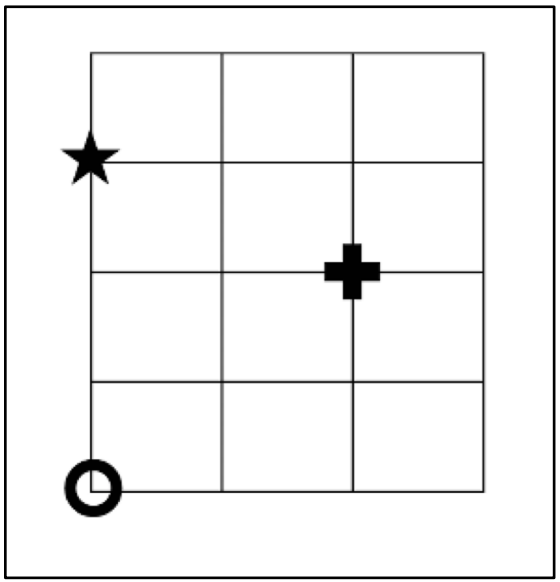


K-Means

- Now what if the lines in the picture **correspond to streets**?
- We have to ***stay on the streets*** to get from one place to another.
- Now the star remains 3 units from the circle.
- But the cross is now 4 units away.
- These distances are called **Manhattan distances**.

Manhattan: The circle is closer to the ***star***.

Euclidean: The circle is closer to the **cross**.



K-Means

- We ***generalize*** the distance metric for **d-dimensional features** by presenting the **Minkowski distance** metric or **L_P norm**.

$$L_p(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^p\right]^{1/p}$$

p = 2: Euclidean Distance

$$L_2(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^2\right]^{1/2}$$

p = 1: Manhattan Distance

$$L_1(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \sum_{i=1}^{d} |x_i - z_i|$$

Minkowski Distance (L_p Norm):

$$L_p(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^p\right]^{1/p}$$

p = 2:
Euclidean Distance

$$L_2(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^2\right]^{1/2}$$

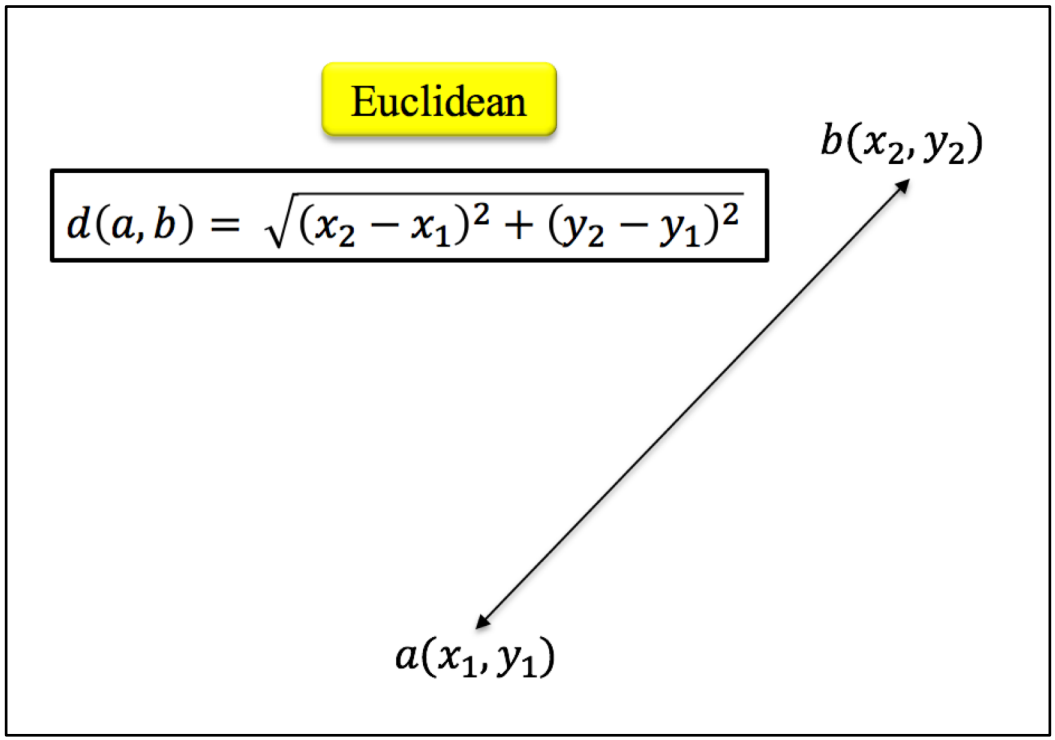
p = 2:
Squared Euclidean Distance

$$d^2(\vec{x}, \vec{z}) = \sum_{i=1}^{d} |x_i - z_i|^2$$

Squared L_2 Norm

Euclidean

$$d(a, b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Minkowski Distance (L_p Norm):

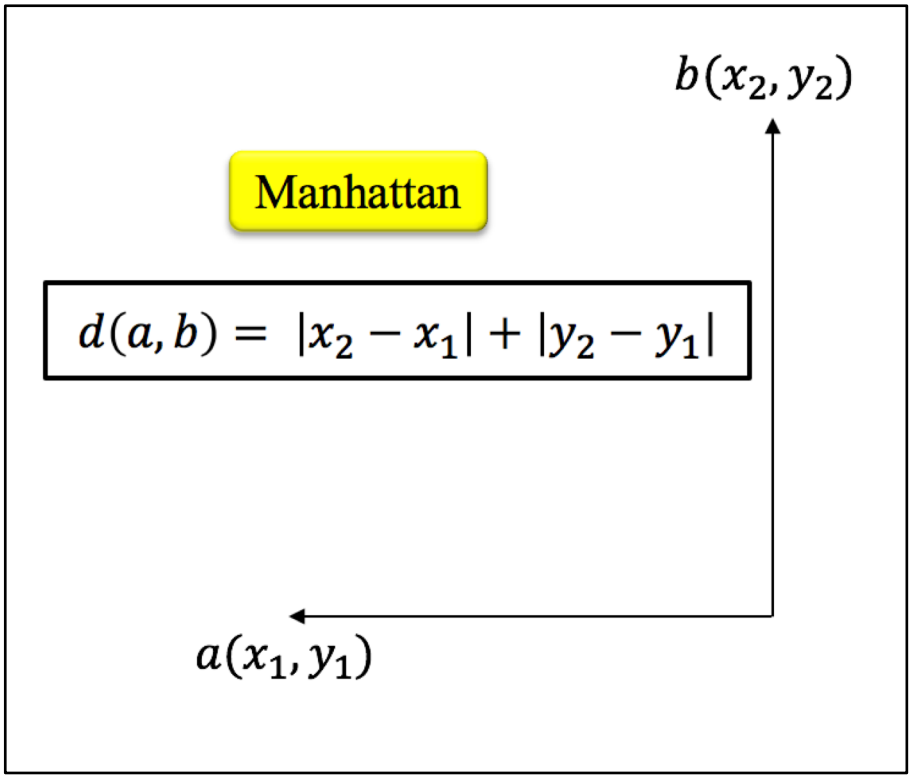
$$L_p(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^p\right]^{1/p}$$

p = 1:
Manhattan or City Block Distance

$$L_1(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \sum_{i=1}^{d} |x_i - z_i|$$

L_1 Norm

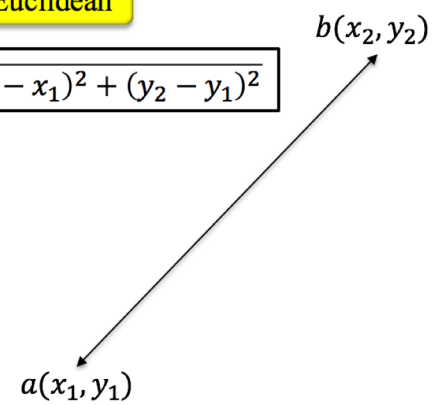
It is defined as the **sum of the lengths of the projections** of the line segment between the points (a and b) **onto the coordinate axes**.



Euclidean vs Manhattan Distance

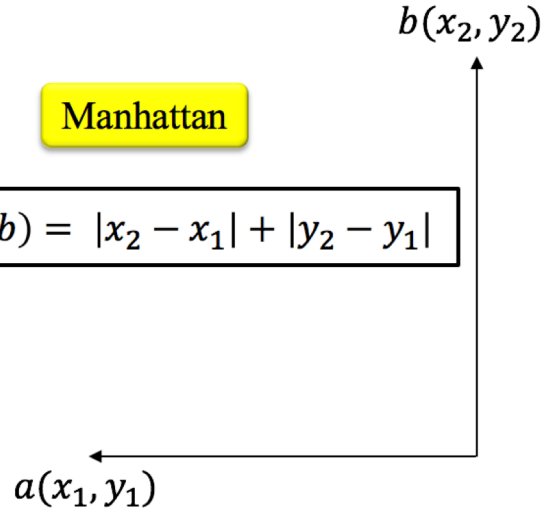
Euclidean

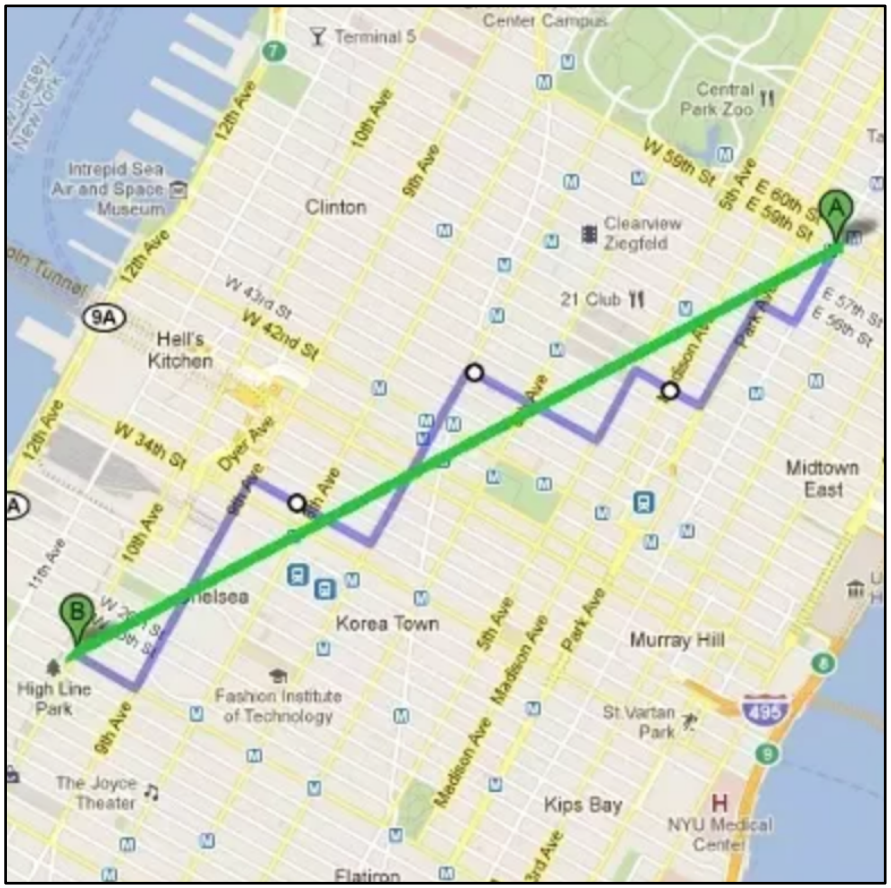
$$d(a,b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$



Manhattan

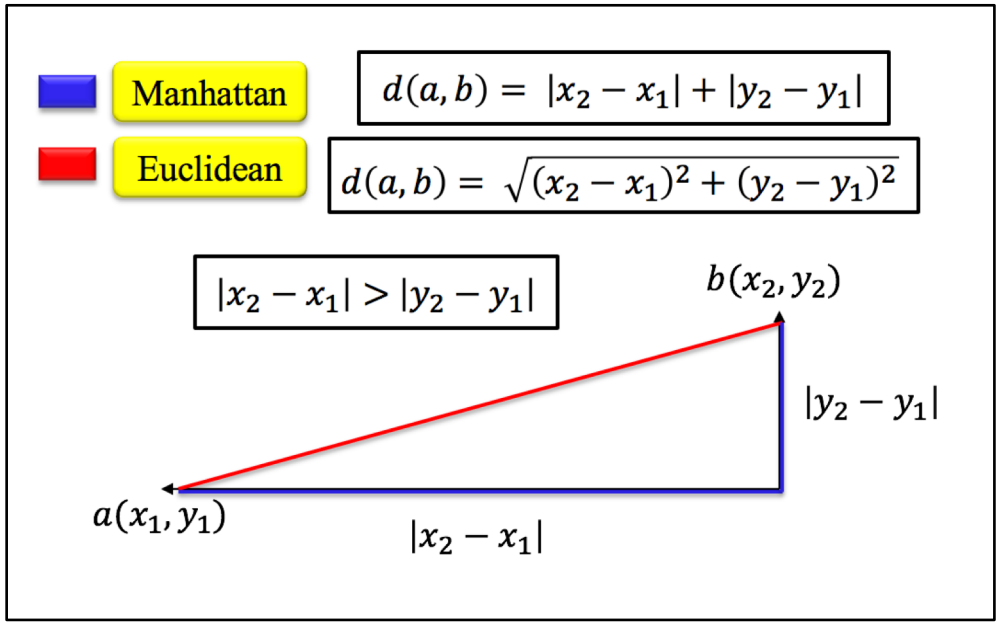
$$d(a,b) = |x_2 - x_1| + |y_2 - y_1|$$





Which distance metric should we use?

Euclidean vs Manhattan Distance



In **high-dimensional** data, the Euclidean distance could ***ignore most of the dimensions.***

Thus, in high-dimensional data, the **Manhattan distance** metric will perform better.

Which distance metric should we use?

It **depends on the dimension** (no. of the features) of the data

In the example, the vectors a and b has the **highest difference** along the x dimension (axis)

Euclidean: the distance between a and b is **dominated by the difference along x dimension**

Manhattan: **both x and y dimensions contribute equally** to measure the distance between a and b

p = 2:
Euclidean Distance

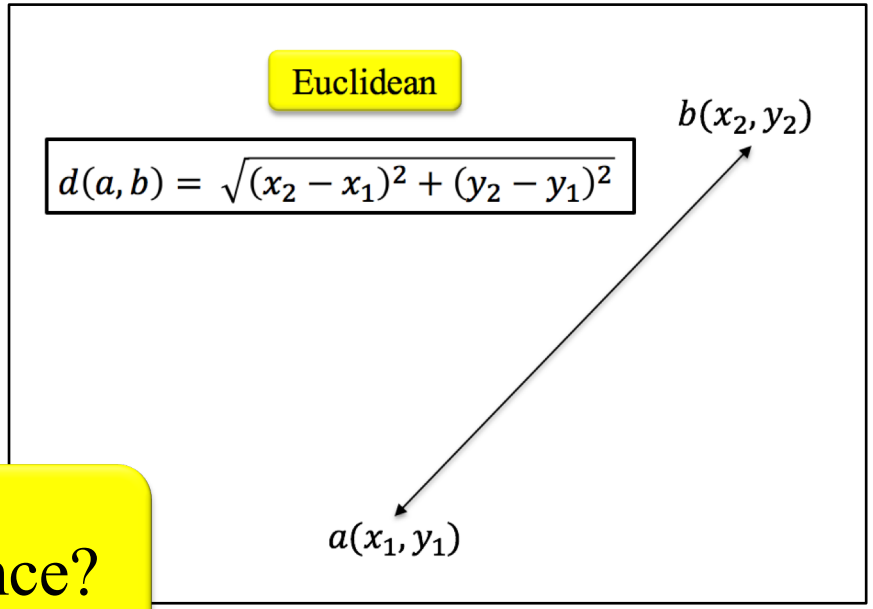
$$L_2(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \left[\sum_{i=1}^{d}|x_i - z_i|^2\right]^{1/2}$$

p = 2:
Squared Euclidean Distance

$$d^2(\vec{x},\vec{z}) = \sum_{i=1}^{d}|x_i - z_i|^2$$

Euclidean

$$d(a,b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

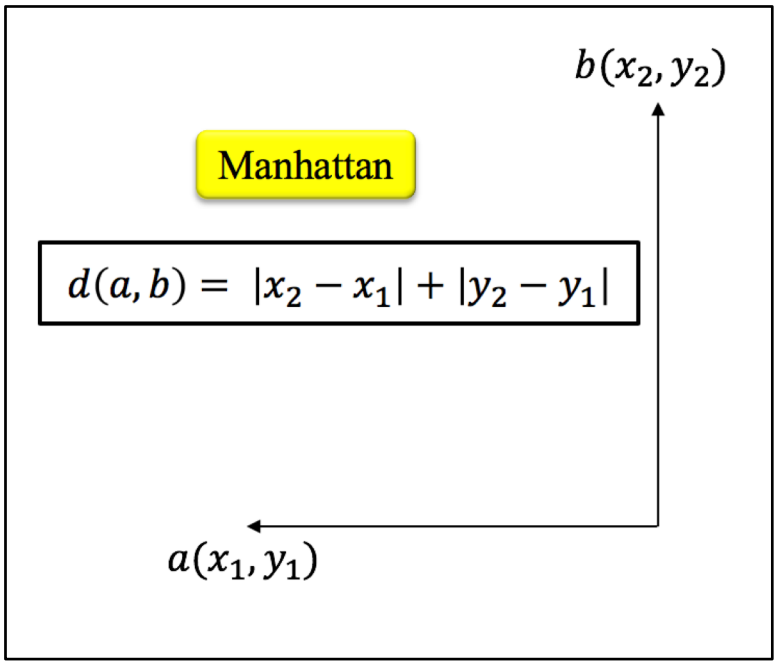


When do we use Euclidean Distance?

When the **dimensions are measuring similar properties**, such as the width, height and depth of parts on a conveyor belt.

p = 1:
Manhattan or City Block Distance

$$L_1(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \sum_{i=1}^{d} |x_i - z_i|$$



When do we use Manhattan Distance?

Manhattan distance is used **dimensions are measuring dissimilar properties**, such as age, weight, and gender of a patient.

It's a common metric used for samples with **binary predictors**.

K-Means

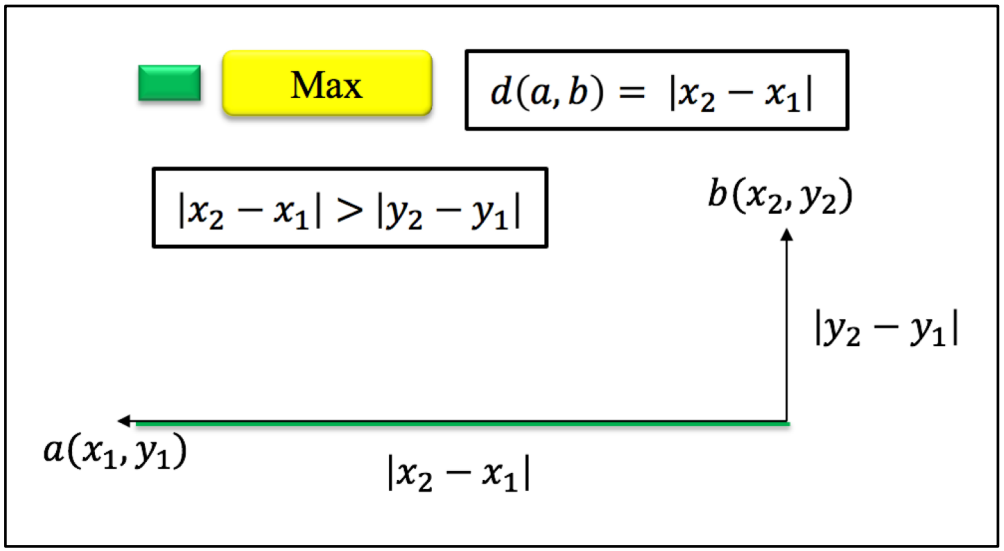
Minkowski Distance (L_p Norm):

$$L_p(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \left[\sum_{i=1}^{d}|x_i - z_i|^p\right]^{1/p}$$

p = ∞:
Max Norm

$$L_\infty(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \max_i|x_i - z_i| \qquad 1 \le i \le d$$

It measures the **absolute value of the difference between elements** with the **largest magnitudes in the two vector**.



p = ∞:
Max Norm

$$L_\infty(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \max_i |x_i - z_i| \qquad 1 \le i \le d$$

Proof:

$$\lim_{p\to\infty}\left[\sum_{i=1}^{d}|x_i - z_i|^p\right]^{1/p}$$

Let j = index of **max difference**

$$= \lim_{p\to\infty}\left[|x_j - z_j|^p + \sum_{i\neq j}|x_i - z_i|^p\right]^{1/p}$$

$|x_j - z_j|$ = absolute value of the difference of the elements with the **largest magnitudes** in the two vectors.

$$= \lim_{p\to\infty}|x_j - z_j|^{p/p} + \left[\sum_{i\neq j}\left(\frac{|x_i - z_i|}{|x_j - z_j|}\right)^p\right]^{1/p}$$

$$\frac{|x_i - z_i|}{|x_j - z_j|} < 1$$

$$= |x_j - z_j|$$

$$\left(\frac{|x_i - z_i|}{|x_j - z_j|}\right)^p \to 0 \quad as\ p \to \infty$$

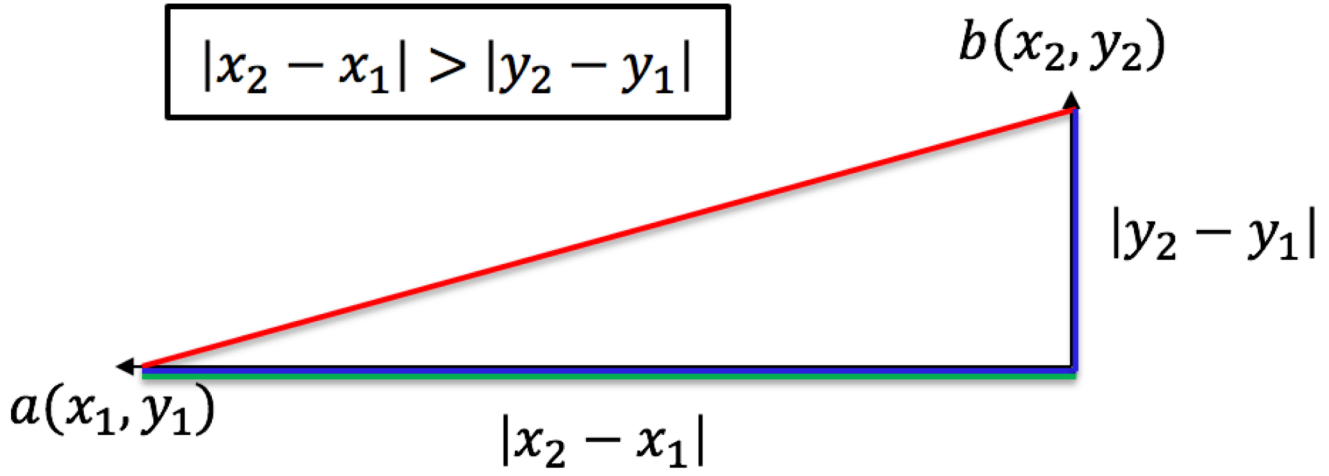
Euclidean vs Manhattan vs Max Distance

Euclidean $d(a,b) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Manhattan $d(a,b) = |x_2 - x_1| + |y_2 - y_1|$

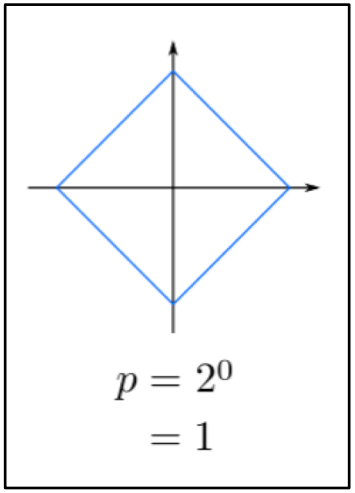
Max $d(a,b) = |x_2 - x_1|$

$|x_2 - x_1| > |y_2 - y_1|$



Let's draw the **contour plot** for L_1 norm & L_2 norm.

On a 2D plane (x-y coordinate), we represent the distance of an arbitrary ***point a*** from the origin (0, 0) by the contour line.

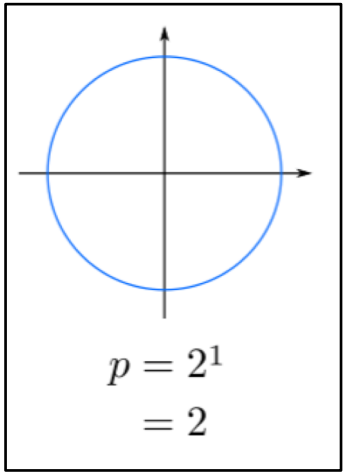


L_1 norm (Manhattan distance)

$$d(a) = |x| + |y|$$

L_2 norm (Euclidean distance)

$$d(a) = \sqrt{x^2 + y^2}$$

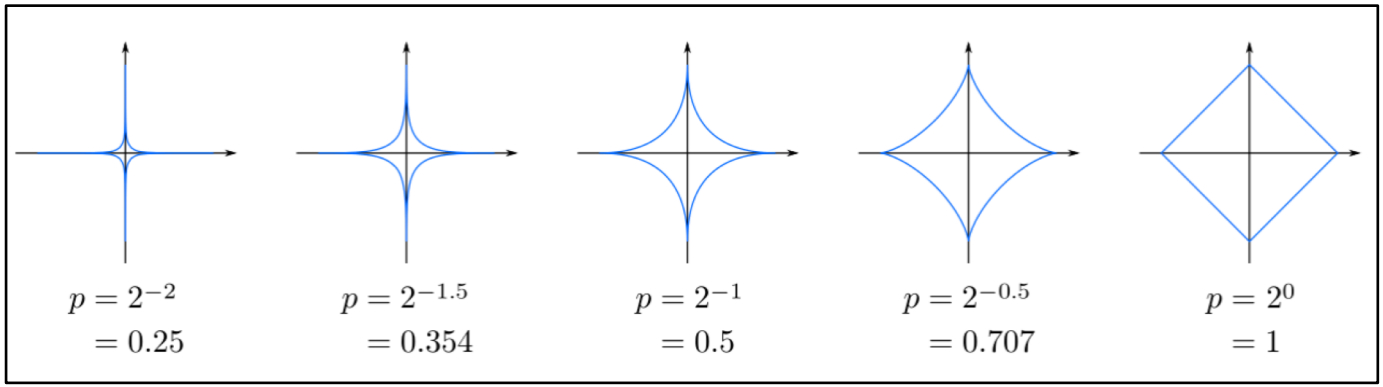


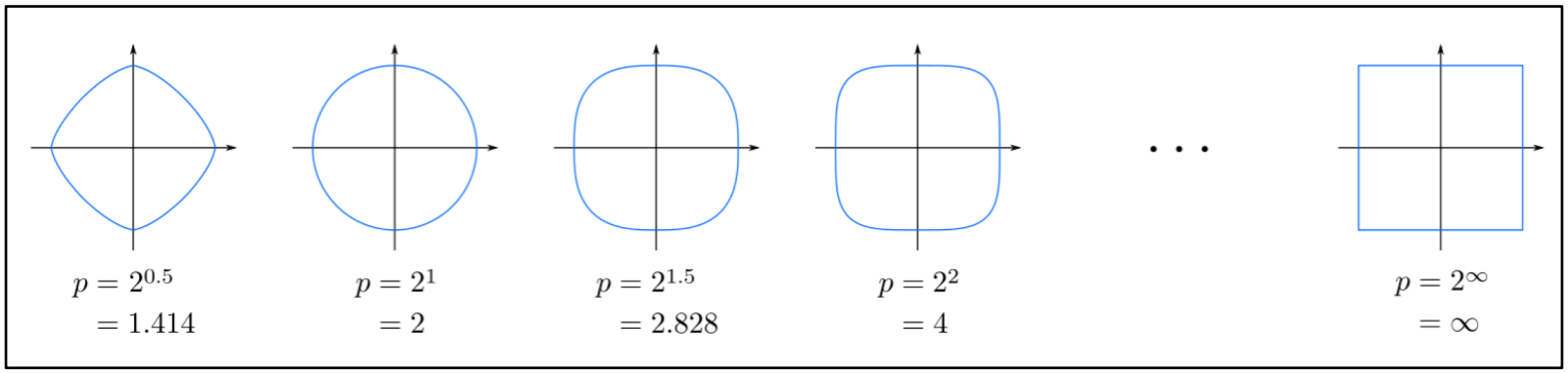
K-Means

Contour Plot for Minkowski Distance (L_p Norm):

$$L_p(\vec{x}, \vec{z}) := d(\vec{x}, \vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^p\right]^{1/p}$$

Contour Plot

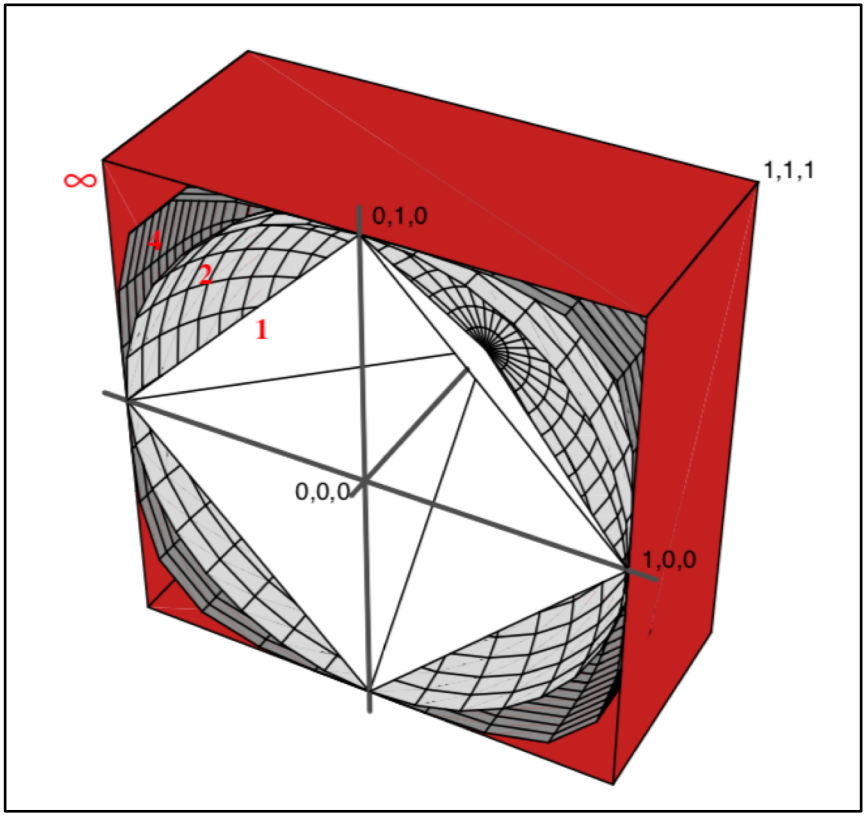


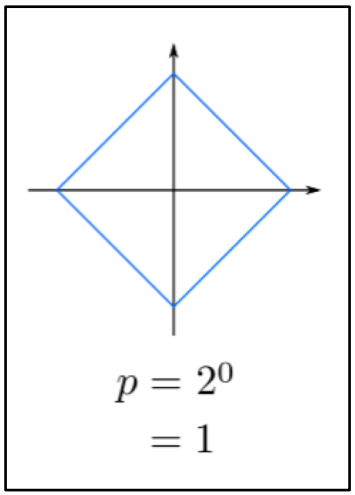


Minkowski Distance (L_p Norm):

$$L_p(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^p\right]^{1/p}$$

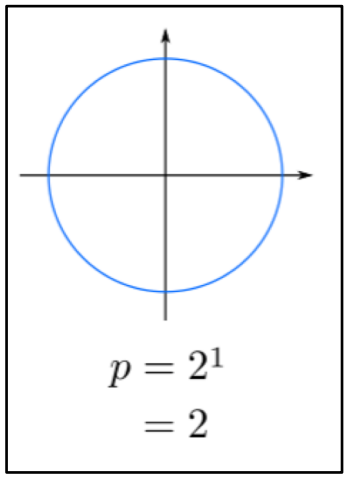
Each **colored surface** consists of points a **distance 1.0** from the origin, measured using **different values for p** in the Minkowski metric (p is printed in red).





L_1 norm (Manhattan distance): white surfaces

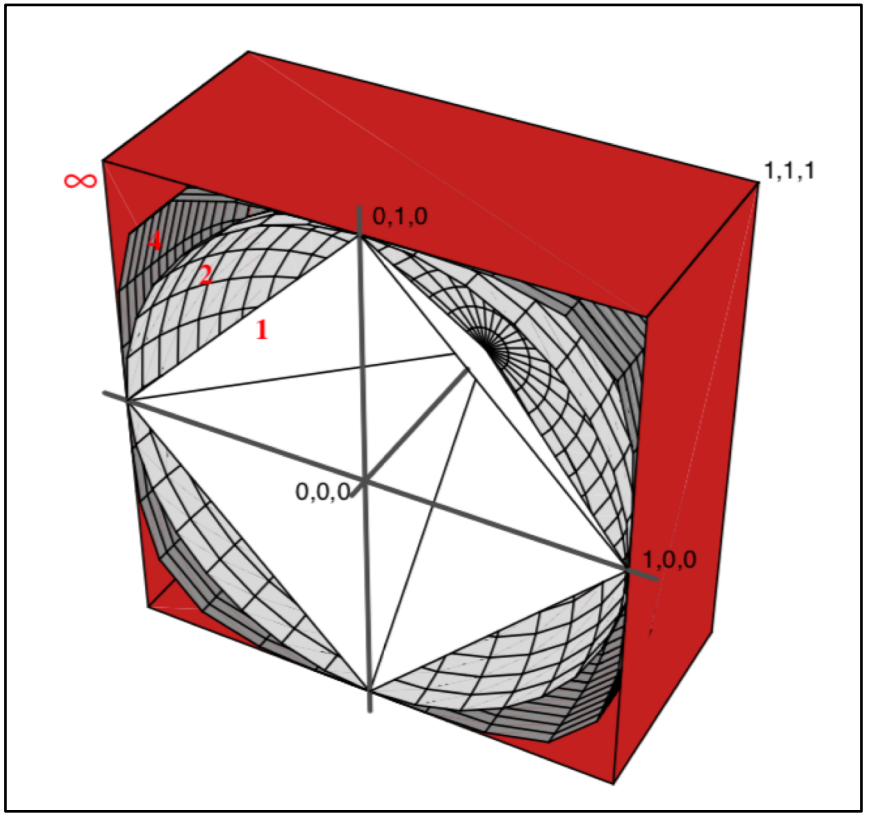
L_2 norm (Euclidean distance): light gray surfaces



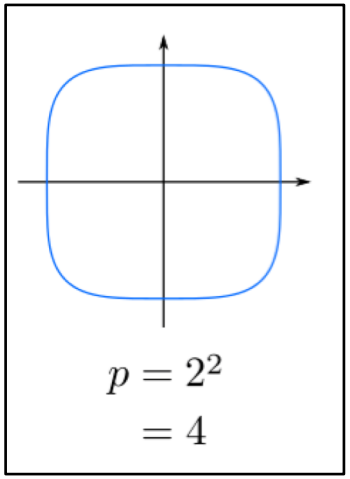
Minkowski Distance (L_p Norm):

$$L_p(\vec{x},\vec{z}) := d(\vec{x},\vec{z}) = \left[\sum_{i=1}^{d} |x_i - z_i|^p\right]^{1/p}$$

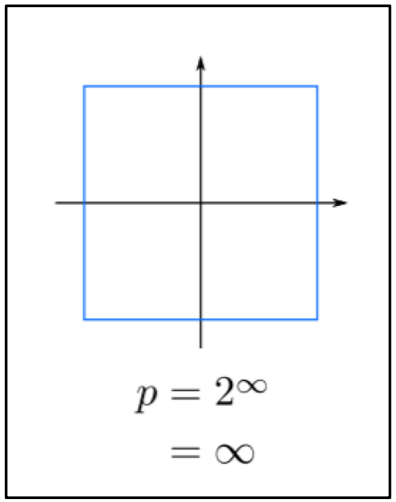
Each **colored surface** consists of points a **distance 1.0** from the origin, measured using **different values for p** in the Minkowski metric (p is printed in red).



L_4 norm: dark gray surfaces



L_∞ norm (max norm): Red surfaces



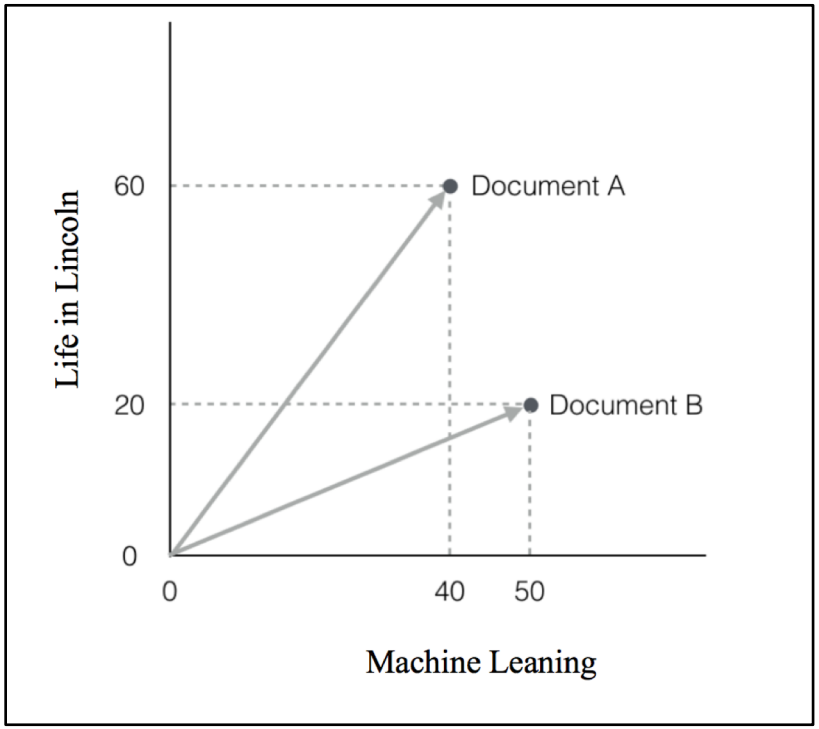
K-Means

- In some scenarios, we **don't want to measure the similarity (distance) of the magnitudes** of the vectors.
- For example, say that we have a **set of documents** on two topics: ***Machine Learning*** and ***Life in Lincoln***.

We want to measure how similar are two documents.

The **length** of the documents could vary!

So, measuring the similarity of their magnitudes will **not be informative**.

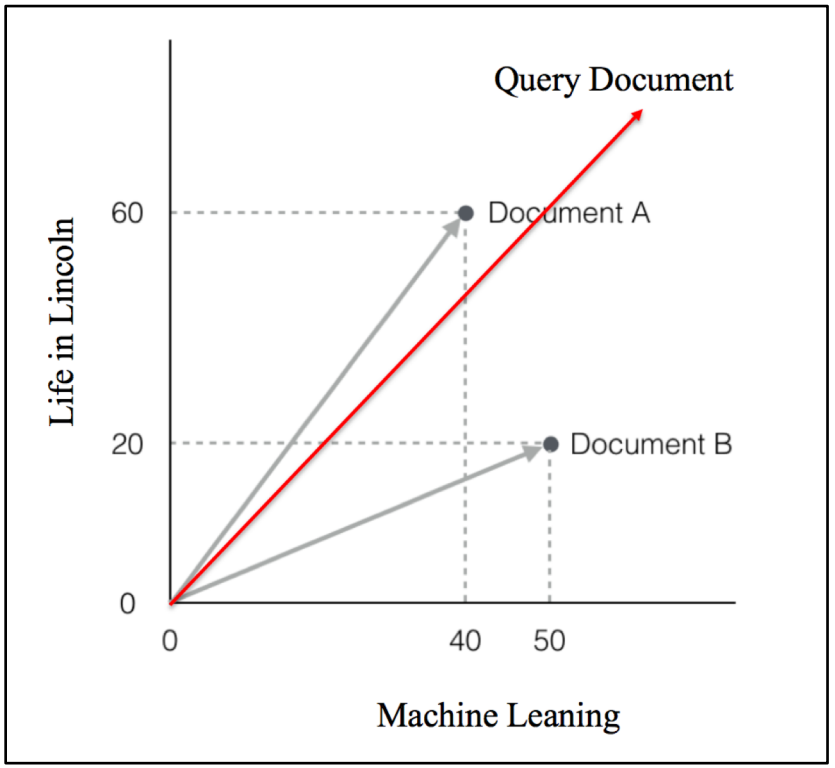


K-Means

- We could probably **measure the orientation** of the two documents to learn their similarity.
- The **cosine similarity measure** enables us to do this.

The cosine similarity between two vectors is a measure that calculates the **cosine of the angle** between them.

This metric is a **measurement of orientation** and ***not magnitude***.

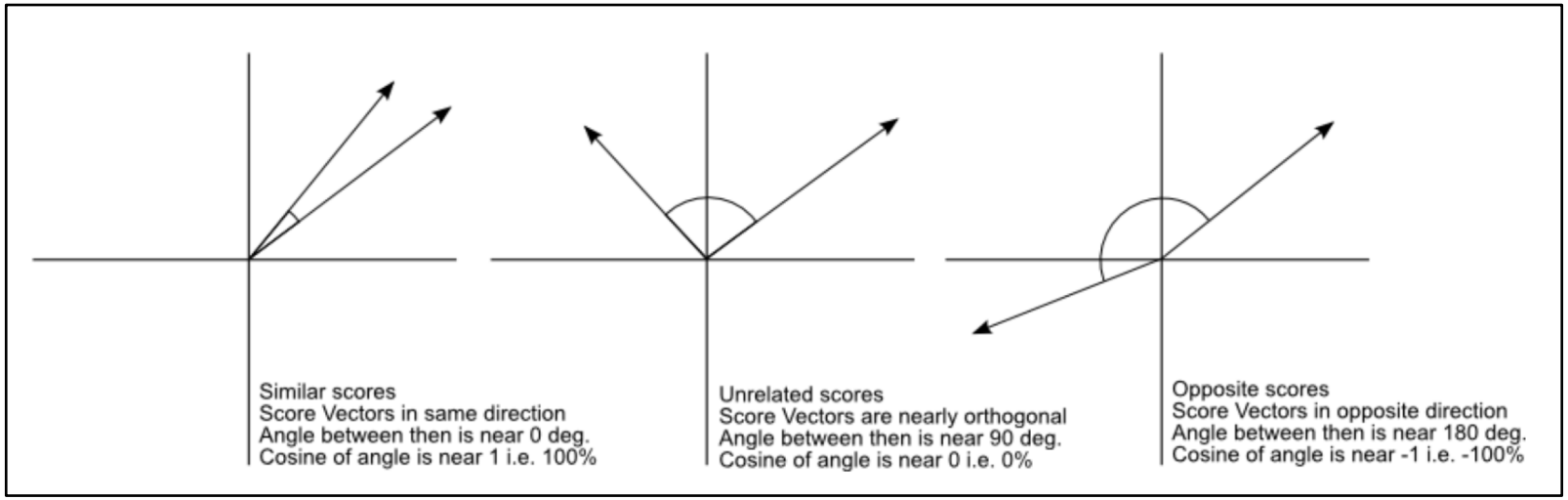


K-Means

- The **cosine** of two non-zero vectors can be derived by using the **Euclidean dot product** formula:

$$\vec{a}.\vec{b} = \|\vec{a}\|.\|\vec{b}\|cos\theta$$

$$\text{Similarity} = cos\theta = \frac{\vec{a}.\vec{b}}{\|\vec{a}\|.\|\vec{b}\|}$$

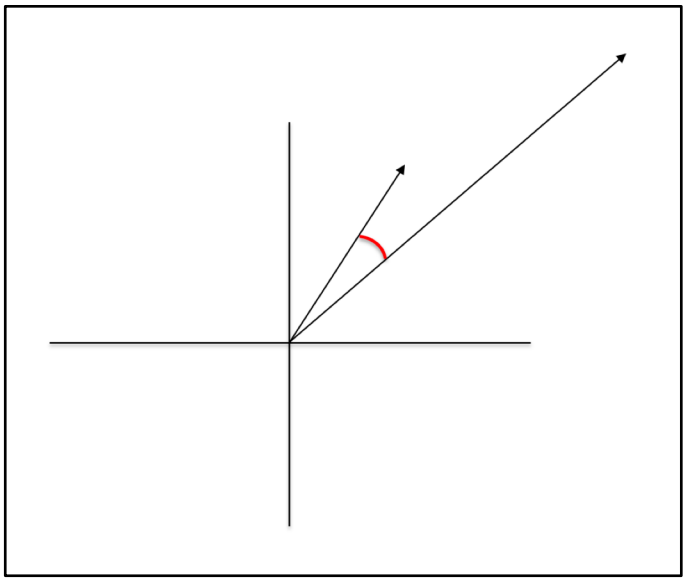


K-Means

- Cosine similarity measure is **useful for comparing documents**.
- Say that we want to compute the **similarity of two documents**.
- One document has the word "Machine Learning" **300 times** and the other document has the same word only **50 times**.

The **magnitude** of the two vectors would ***vary a lot***.

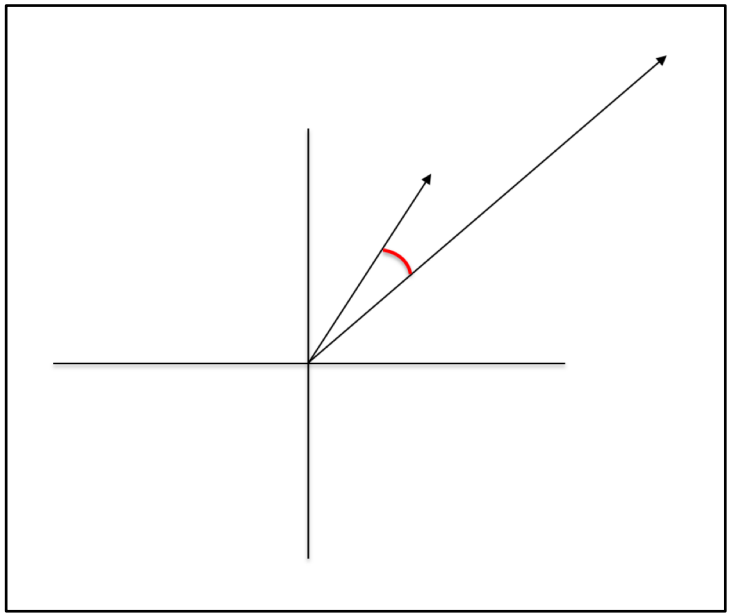
Thus, the **Euclidean distance** between the two documents will be **higher**.



K-Means

- However, if the two documents **point to the same direction** (small angle), their cosine similarity would be larger.

In this scenario **cosine similarity is suitable** because it tends to **ignore the higher term count** on documents.



K-Means

- In summary, there is **no optimal distance metric** for all types of datasets.
- See the results of **comparison** between various distance metrics:
- https://arxiv.org/pdf/1708.04321.pdf