21 feb 23
Monday

Phase 2 - Day 1

Phase 2 - 4 weeks

Phase flow ↓

Maths Concepts - 1 (Basics)
Maths Concepts - 2 (Intermediate)
} Foundation

→ Distance based approach / Nearest Neighbour Approach
  ↳ KNN (Reg. & Class.)
  - K-means clustering
  - Learning Vector Quantization (LVQ)
  - Self-Organizing Map (SOM)

→ Rule based approach
  ↳ Decision Trees (Reg & Class.)

→ Probability based Approach
  ↳ Naive bayes (Class & Reg (very Advanced))

→ Boundary Based Approach
  ↳ SVM (Reg. & Class)
  ↳ Linear Reg. (Reg.)
  ↳ Logistic Reg. (Class.)

} Phase 2

Combining / Grouping multiple approaches → Ensemble Approach
  ↳ Voting Ensembles
  ↳ Stacking
  ↳ Bagging → Random Forest (Reg. & Class)
  ↳ Cascading & Boosting
    ↳ ADA Boost
    ↳ GBDT
    ↳ XG Boost

} Phase 3

Maths Part - 1

Q How to represent a dataframe?

$D_R \rightarrow$ $n \times d$ (no. of rows, no. of columns)

$D_n = \{(x_i, y_i)\}_{i=1}^{n} \mid x_i \in R^{d-1}, \underbrace{y_i \in R}_{\text{Regression Task}}\}$

$D_n = \{(x_i, y_i)\}_{i=1}^{n} \mid x_i \in R^{d-1}, \underbrace{y_i \in (-1, +1)}_{\text{Classification Task}}\}$

$$D_n = \{ (x_i, y_i) \}_{i=1}^{n} \mid x_i \in R^{d-1},\ y_i \in \begin{cases} R & \text{(output is real value)} \Rightarrow \text{Regression} \\ \{0,1\} & \text{(output is a discrete set)} \Rightarrow \text{classification} \end{cases}$$

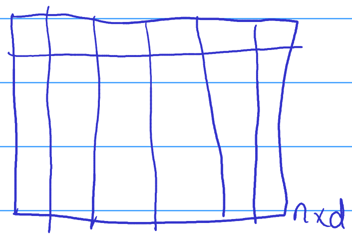
Such that

Real value of d-1 dimensions

Belongs to

(Collection of rows) (and each row is an ordered pair of i/p & o/p)

Q How to represent a dataframe in mathematics?



dataframe

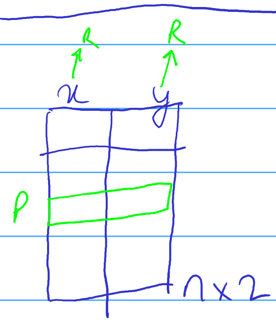
Matrix

$$D_n = \left\{ (x_i, y_i)_{i=1}^{n} \mid x_i \in R^{d-1},\ y_i \in R \text{ or } y_i \in \{0,1\} \right\}$$

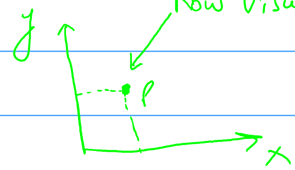
[Matrix of shape n x d]

{ } → discrete value set

[ ] → Range value set.

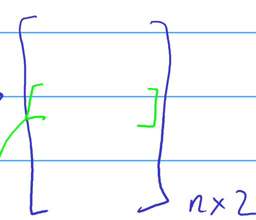


dataframe



Row Visual Representation

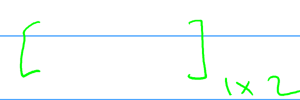
How to represent a dataframe/row in mathematics



matrix

It is a collection of vectors.

This is called a vector

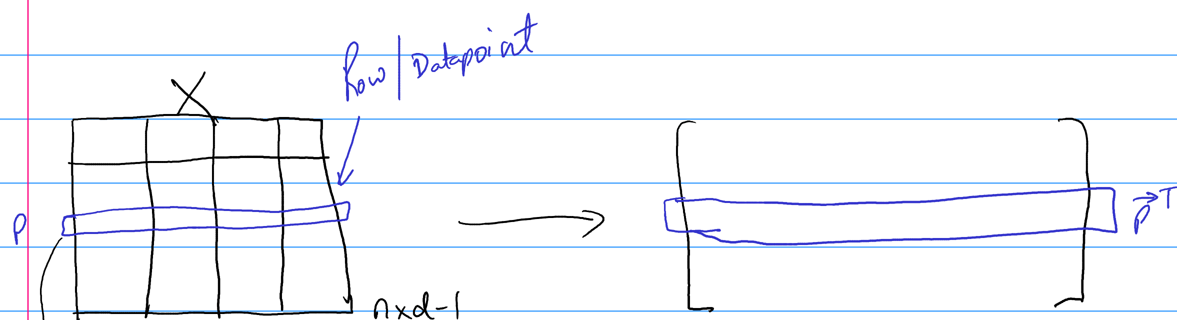


* Matrix is used to represent a dataframe
* Collection of vectors = matrix
* 1 vector = 1 datapoint / row

Types of Vector

↳ Column Vector → $\begin{bmatrix} \\ \\ \end{bmatrix}_{n \times 1}$ ← By default Representation of a datapoint/row.

↳ Row Vector → $\begin{bmatrix} & & & \end{bmatrix}_{1 \times n}$



↳ Mathematical Representation of Row ⇒ Column Vector

$$\Rightarrow \vec{p} = \begin{bmatrix} \\ \\ \end{bmatrix}_{d-1 \times 1}$$

↳ Column Vector representation of datapoint/row

F=1, M=0

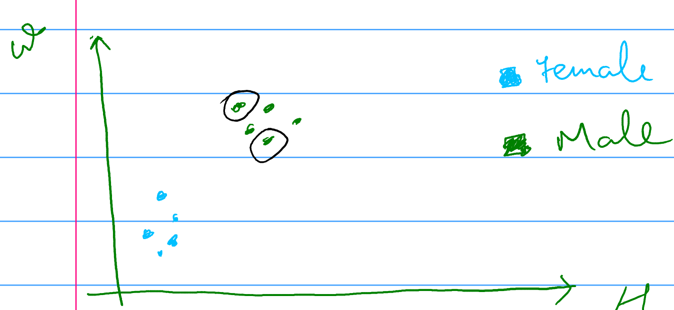
eg

	H	W	G
a	150	50	F
b	170	80	M
c	175	85	M
d	145	45	F

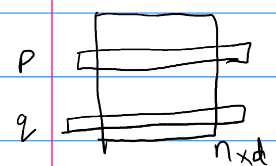
$$\longrightarrow \begin{bmatrix} 150 & 50 & 1 \\ 170 & 80 & 0 \\ 175 & 85 & 0 \\ 145 & 45 & 1 \end{bmatrix} = \begin{bmatrix} \vec{a}^T \\ \vec{b}^T \\ \vec{c}^T \\ \vec{d}^T \end{bmatrix}$$

Mathematical Representation of d row → Column Vector

$$\Rightarrow \vec{d} = \begin{bmatrix} 145 \\ 45 \\ 1 \end{bmatrix}$$



Performing Vector Algebra:



$$\vec{p} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_d \end{bmatrix}_{d\times 1} \qquad \vec{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_d \end{bmatrix}_{d\times 1}$$

Vector Addition

$$\vec{p} + \vec{q} = \begin{bmatrix} p_1 + q_1 \\ p_2 + q_2 \\ \vdots \\ p_d + q_d \end{bmatrix}_{d\times 1}$$

Vector Subtraction

$$\vec{p} - \vec{q} = \begin{bmatrix} p_1 - q_1 \\ p_2 - q_2 \\ \vdots \\ p_d - q_d \end{bmatrix}_{d\times 1}$$

Component wise operation

Output → Vector

Vector Multiplication

dot product ⇓ $\vec{p}\cdot\vec{q}$

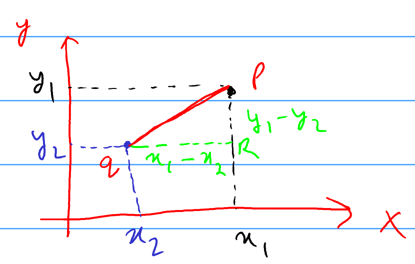
Cross product

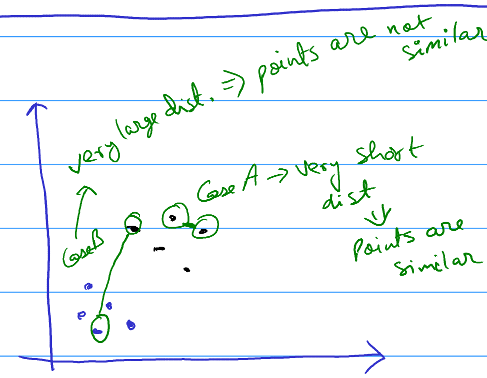
$$\boxed{\vec{p}\cdot\vec{q} = p_1 q_1 + p_2 q_2 + \cdots + p_d q_d}$$

Output → Scalar Quantity.

Distance between 2 vectors / datapoints

	x	y
p	x_1	y_1
q	x_2	y_2





$$dist(pq) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

ΔPQR is Right angled

$\Rightarrow pq^2 = qR^2 + pR^2$

$\Rightarrow \boxed{pq = \sqrt{qR^2 + pR^2}}$

$\vec{p} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ $\vec{q} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ $\boxed{dist(\vec{p},\vec{q}) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}}$ ☆

This is known as <u>Euclidean distance</u> b/w 2 vectors

✱ In case of 3 dimensions

$\vec{p} = \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix}$ $\vec{q} = \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix}$ $dist(\vec{p},\vec{q}) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$

✱ In case of d dimensions

datapoint with 'd' features

$\vec{p} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \end{bmatrix}$ $\vec{q} = \begin{bmatrix} x_2 \\ y_2 \\ \vdots \end{bmatrix}$ $dist(\vec{p},\vec{q}) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2+ \cdots\cdots}$

---

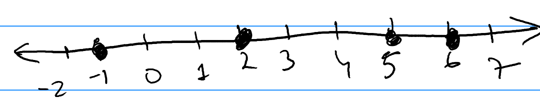
22 Feb 2023
Wednesday

Dataframe

	f1
dp1	5
dp2	-1
dp3	6
dp4	2

$n \times 1$

Vizualize $\longrightarrow$



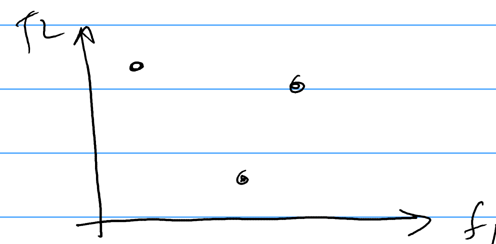
(Scatter Plot)

Q Which datapoint is most similar to 5?

Ans DP3 i.e. 6

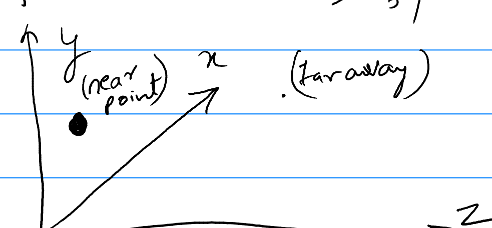
f1	f2

$\longrightarrow$



f1	f2	f3

$\longrightarrow$



any higher dimension is not possible to visualize

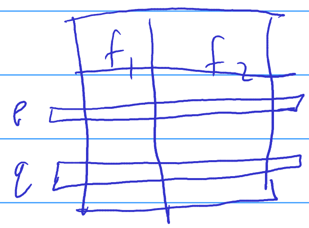
* If you have more than 3 dimensions → Vizualization fails (but projections may be used)

Q. How do you find patterns in higher dimension dataset?

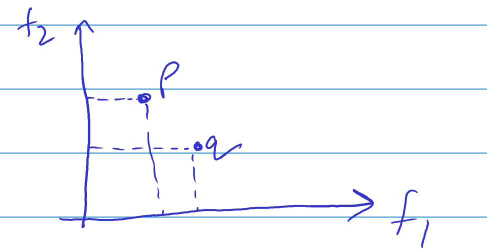
Ans. Maths.

Vector representation of a data point / Row

2-dim:

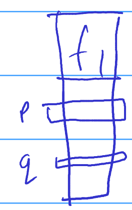


dataframe

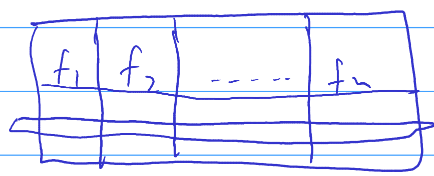


Vizualizat^n (Scatter Plot)

1-D:



n-D:



→ Viz. fails

Maths Representation

2-D → $\vec{p} = \begin{bmatrix} - \\ - \end{bmatrix}_{2\times1}$ $\quad \vec{q} = \begin{bmatrix} - \\ - \end{bmatrix}_{2\times1}$

1-D → $\vec{p} = \begin{bmatrix} - \end{bmatrix}_{1\times1}$ $\quad \vec{q} = \begin{bmatrix} - \end{bmatrix}_{1\times1}$

n-D → $\vec{p} = \begin{bmatrix} - \\ - \\ - \\ - \\ \vdots \\ - \end{bmatrix}_{n\times1}$ $\quad \vec{q} = \begin{bmatrix} - \\ - \\ - \\ - \\ \vdots \\ - \end{bmatrix}_{n\times1}$

$\left.\begin{array}{l} \vec{p} + \vec{q} \\ \vec{p} - \vec{q} \end{array}\right\}$ Vector o/p

$\vec{p} \cdot \vec{q}$ } Scalar

$dist(\vec{p}, \vec{q})$

for d-dimension

$$\boxed{dist(\vec{p},\vec{q}) = \left(\sum_{i=1}^{d}(p_i - q_i)^2\right)^{1/2}}$$

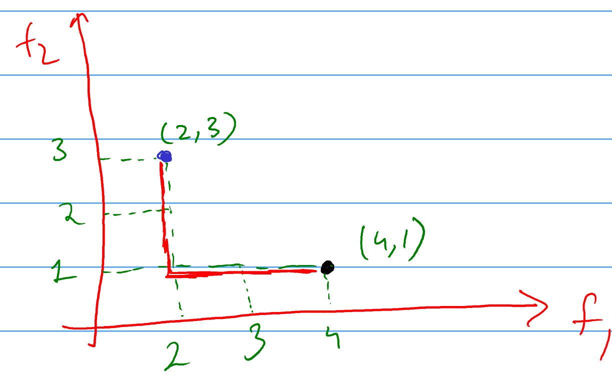
General form of Euclidean distance from which distance formula for all other dimensions is derived

Distance is important as it helps in identifying similarity b/w datapoints

$$\boxed{\text{Similarity} \propto \frac{1}{\text{Distance}}}$$

---

	f_1	f_2
dp1	2	3
dp2	4	1



Manhattan distance : when you travel in grid fashion [Hollywood movie : 3 blocks cast]

Idea behind Manhattan distance is to find the shortest distance between 2 points following grid traversal rules [↔ ↕ no diagonal traversal]

$$dist(dp1, dp2) = abs(2-4) + abs(3-1)$$

$$\begin{bmatrix}2\\3\end{bmatrix} \begin{bmatrix}4\\1\end{bmatrix} = 2 + 2 = 4$$

This distance is Manhattan distance b/w dp1 & dp2

$$\text{Euclidean distance} = \sqrt{(2-4)^2+(3-1)^2}$$
$$= \sqrt{4+4}$$
$$= 2\sqrt{2}$$

for d-dim: $dist(\vec{p},\vec{q}) = abs(p_1-q_1) + abs(p_2-q_2) + \dots + abs(p_d-q_d)$

$$\begin{bmatrix}p_1\\p_2\\\vdots\\p_d\end{bmatrix} \begin{bmatrix}q_1\\q_2\\\vdots\\q_d\end{bmatrix}$$

$$\Rightarrow \boxed{\left(\sum_{i=1}^{d}(|p_d - q_d|)^{1}\right)^{1/1} = dist(\vec{p},\vec{q})}$$

General form of Manhattan distance

General form of Euclidean distance

$$dist(\vec{p}, \vec{q}) = \left(\sum_{i=1}^{d} \left(abs(p_i - q_i)\right)^{2} \right)^{\frac{1}{2}}$$

Minkowski distance for power $= [3, \infty)$

$$dist(\vec{p}, \vec{q}) = \left(\sum_{i=1}^{d} \left(abs(p_i - q_i)\right)^{3} \right)^{\frac{1}{3}}$$

$$dist(\vec{p}, \vec{q}) = \left(\sum_{i=1}^{d} \left(abs(p_i - q_i)\right)^{4} \right)^{\frac{1}{4}}$$

$$\vdots$$

$$dist(\vec{p}, \vec{q}) = \left(\sum_{i=1}^{d} \left(abs(p_i - q_i)\right)^{n} \right)^{\frac{1}{n}}$$

Minkowski Distance

$$dist(\vec{a}, \vec{b}) = \left(\sum_{i=1}^{d} \left(abs(a_i - b_i)\right)^{p} \right)^{\frac{1}{p}}$$

General formula to compute distance b/w 2 point

$p=1$ [manhattan distance]

$$dist(\vec{a}, \vec{b}) = \left(\sum_{i=1}^{d} \left(abs(a_i - b_i)\right)^{1} \right)^{\frac{1}{1}}$$

$p=2$ [Euclidean distance]

$$dist(\vec{a}, \vec{b}) = \left(\sum_{i=1}^{d} \left(abs(a_i - b_i)\right)^{2} \right)^{\frac{1}{2}}$$

for $p >= 3$, it is minkowski distance of that power.

* whenever performing vector Algebra $(+, -, \times$ (dot product)), ensure that the vectors are in same vector space i.e. their shape should be same.

23 feb 2023
Thursday

$$x_i = \begin{bmatrix} h_i \\ w_i \\ g_i \end{bmatrix} \qquad x_j = \begin{bmatrix} h_j \\ w_j \\ g_j \end{bmatrix}$$

$$\vec{x_i} \cdot \vec{x_j} = h_i \times h_j + w_i \times w_j + g_i g_j$$

let $\vec{x_i} = \vec{a}$ & $\vec{x_j} = \vec{b}$

$$\Rightarrow \boxed{\vec{a} \cdot \vec{b} = \sum_{i=1}^{d} a_i b_i}$$

General form of dot product.

* In case of matrix multiplication

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}_{3\times 2} \qquad B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}_{2\times 3}$$

Only for multiplication can they be equal & multiplied.

$C = A \times B$

$$C = \begin{bmatrix} 1 & 2 & 0 \\ 3 & 4 & 0 \\ 5 & 6 & 0 \end{bmatrix}_{3\times 3}$$

$\Rightarrow$ From above, we can infer that

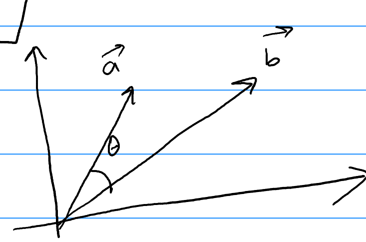
$$\vec{a} \cdot \vec{b} = a^T \times b$$

in General form

$$\boxed{\begin{array}{c} \vec{a} \cdot \vec{b} = \sum_{i=1}^{d} a_i \times b_i \\ \hline \vec{a} \cdot \vec{b} = a^T \times b = a^T b \end{array}}$$

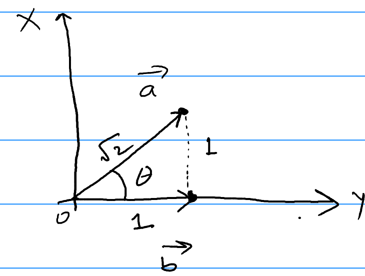
* magnitude of a vector gives the length of the vector

* $\boxed{\vec{a} \cdot \vec{b} = |\vec{a}| |\vec{b}| \cos\theta}$



Q $\vec{a} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $\vec{b} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$\vec{a} \cdot \vec{b} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 0 + 1 = 1$$



$$\vec{a} \cdot \vec{b} = |\vec{a}|\,|\vec{b}| \cos\theta$$

$$|\vec{a}| = \sqrt{1^2 + 1^2} = \sqrt{2}$$

$$|\vec{b}| = \sqrt{1^2 + 0^2} = \sqrt{1} = 1$$

$$\cos\theta = \frac{1}{\sqrt{2}} \Rightarrow \theta = \cos^{-1}\left(\frac{1}{\sqrt{2}}\right) = 45^0$$

$$\vec{a} \cdot \vec{b} = \sqrt{2} \times 1 \times \cos 45 = \cancel{\sqrt{2}} \times 1 \times \frac{1}{\cancel{\sqrt{2}}} = 1$$

$$\boxed{\vec{a} \cdot \vec{b} = 1}$$ Ans

Note: ① Length of a vector = Euclidean distance b/w the vector and origin

$$\sqrt{(x_i - 0)^2 + (y_i - 0)^2} = \sqrt{x_i^2 + y_i^2}$$

* Hamming Distance

↳ Calculates the dist. b/w 2 binary vectors.

↳ aka binary string or bitstrings

* Generally encountered for One hot Encoded data.

eg red = [1, 0, 0]

Green = [0, 1, 0]

blue = [0, 0, 1]

$HD_{RG} = |(1-0)| + |0-1| + |0-0|$

$= 1+1+0$

$= 2$

for OHE string

$HD = \sum_{i=1}^{n} |V1[i] - V2[i]|$

for bitstrings with many 1 it is more common to calculate the avg. no. of bit differences to give a hamming distance score b/w 0 (identical) & 1 (All different)

$HD = \frac{\sum_{i=1}^{n} |V1[i] - V2[i]|}{n}$

eg $R_1 = [0, 0, 0, 0, 0, 1]$

$R_2 = [0, 0, 0, 0, 1, 0]$

$HD = \frac{|0-0| + |0-0| + |0-0| + |0-0| + |0-1| + |1-0|}{6}$

$HD = \frac{2}{6} = \frac{1}{3} = 0.33$

Note: Vector L1 norm → Manhattan Distance

Vector L2 norm → Euclidean Distance

Vector max norm → $\text{maxnorm}(v) = \|v\|^{inf} = \|v\|^{\infty}$

$\|v\|^{inf} = \max(|a_1|, |a_2|, |a_3|)$

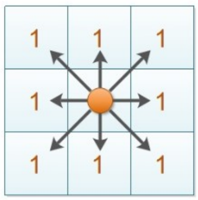
eg [1, 2, 3]

$\text{maxnorm}([1, 2, 3]) = 3.0$ Ans

also used as regularization on Neural Network weights, called max norm regularization

Chebyshev Distance: It is a special case of minkowski distance when $p = \infty$.

Chebyshev Distance



$$\text{Chebyshev dist} = \max\left(|x_1 - x_2|, |y_1 - y_2|\right)$$

General form:

Given two Points $P_1(x_1, x_2, \dots, x_n)$

$P_2(y_1, y_2, \dots, y_n)$

Chebyshev dist $= \max(|x_i - y_i|)$ where i is 1 to n.

In daily life programming of chess based problems:

↳ dist. for Rook = Manhattan distance.

↳ dist. for Queen = Chebyshev distance

↳ dist. for bishop = Manhattan dist.

[b/w the squares of same color with axes rotated by 45° i.e. diagonals are axis]

↳ dist. for kings = Euclidean distance

Cosine distance:

$$\left.\begin{aligned} \vec{a}\cdot\vec{b} &= \sum_{i=1}^{d} a_i \times b_i \\ \vec{a}\cdot\vec{b} &= |\vec{a}|\,|\vec{b}|\cos\theta \end{aligned}\right\} \Rightarrow \sum_{i=1}^{d} a_i \times b_i = |\vec{a}|\,|\vec{b}|\cos\theta$$

$$\Rightarrow \cos\theta = \frac{\sum_{i=1}^{d} a_i \times b_i}{|\vec{a}|\,|\vec{b}|}$$

$|\vec{a}|$ & $|\vec{b}|$ can be found using ①

$|\vec{a}| = \sqrt{a_1^2 + a_2^2}$ for n dimension $|\vec{a}| = \sqrt{a_1^2 + a_2^2 + \dots + a_n^2}$

$= \sqrt{\sum_{i=1}^{d} a_i^2}$

$$\Rightarrow \boxed{\theta = \cos^{-1}\left(\frac{\sum_{i=1}^{d} a_i \times b_i}{|\vec{a}|\,|\vec{b}|}\right)}$$

$$\Rightarrow \boxed{\theta = \cos^{-1}\left(\frac{\sum_{i=1}^{d} a_i \times b_i}{\sqrt{\sum_{i=1}^{d} a_i^2}\,\sqrt{\sum_{i=1}^{d} b_i^2}}\right)}$$

Cosine Similarity

(1) This metric is used when you want to determine how similar two documents or corpus of text are.

(2) The cosine similarity returns the normalized dot product of two vectors.

dot product → operatⁿ where Two equal length vectors are multiplied resulting in a single scalar.

(3) Cosine similarity has the best use case when we are interested in orientation but not in the magnitude of the vector.

(4) Given two vectors:

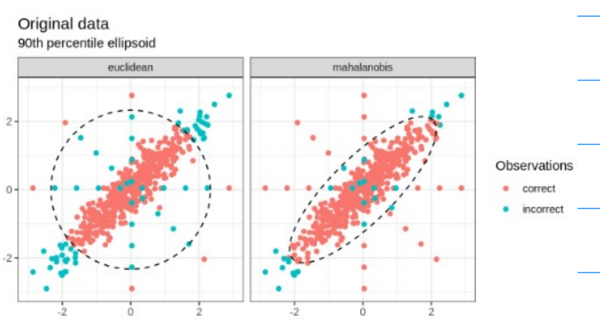
They are being taken with independence of magnitude

- → with same orientation ⇒ cosine similarity = 1
- → with 90° orientation ⇒ cosine similarity = 0
- → with opposite orientation ⇒ cosine similarity = −1

Mahalanobis Distance

↳ Used to calculate distance between two data points in multivariate space.

It can be understood as a measure of the distance between a point P and a distribution D. The basic idea of measuring is, how many standard deviations away P is from the mean of D



Advantage: It takes covariance in account which helps in measuring the strength / similarity between two different data objects.

$$\text{Mahalanobis distance} = \sqrt{(\vec{x}-\vec{\mu})^T S^{-1} (\vec{x}-\vec{\mu})}$$

$S \rightarrow$ Covariance Metric

* We are using S^{-1} to get a variance-normalized distance equatⁿ.

* Jaccard Distance : It is used to calculate distance between sets.

set → $\{1, 2, 3, 4\}$

cardinality of set = |set| = number of elements contained in the set.

Jaccard dist. (Jaccard Similarity) $= \dfrac{|A \cap B|}{|A \cup B|}$

eg

$A = \{$"flower", "dog", "cat", 1, 3$\}$

$B = \{$"flower", "cat", "boat"$\}$

$\Rightarrow |A \cap B| = 2$

$\Rightarrow |A \cup B| = 6$

$\Rightarrow$ Jaccard dist $= \dfrac{|A \cap B|}{|A \cup B|} = \dfrac{2}{6} = \dfrac{1}{3}$

* K-means algo. uses Jaccard dist.