CIS 635 Knowledge Discovery & Data Mining

Introduction to Linear Algebra

Basic Math - Data Science

We are aware of Scalars: A person’s

Height (1.72m)

Basic Math - Data Science

We are aware of Scalars: A person's

Height (1.72m)
Weight (72kg)

Basic Math - Data Science

We are aware of Scalars: A person's

Height (1.72m)
Weight (72kg)
Salary (100K)

Basic Math - Data Science

We are aware of Scalars: A person's

Height (1.72m)
Weight (72kg)
Salary (100K)
....

Basic Math - Data Science

A closed form definition of person through some features

[Height (1.72m), Weight (72kg), Salary   (100K)]

Basic Math - Data Science

A closed form definition of person through some features

- no explicit unit mentions

[1.72, 72, 100]

Basic Math - Data Science

A closed form definition of person through some features

- no explicit unit mentions

[1.72, 72, 100]

Is a vectoried representation of the person through some attributes: height, weight,salary

Basic Math - Data Science

We are aware of Scalars: A person's height, weight, salary

1. Vectors

We begin by defining a mathematical abstraction known as a **vector space**. In linear algebra the fundamental concepts relate to the n-**tuples** and their algebraic properties.

Definition: An ordered n-**tuple** is considered as a sequence of n **terms** $(a_1, a_2, \cdots, a_n)$, where n is a positive integer.

We see that an ordered n-**tuple** has **terms** whereas a set has members.

Example: A sequence (5) is called an ordered 1-**tuple**. A 2-**tuple**, for example (3, 6) (where 6 appears after 3) is called an ordered pair, and 3-**tuple** is called an ordered triple. A sequence (9, 3, 4, 4, 1) is called an ordered 5-**tuple**.

Let us denote the set of all ordered 1-**tuples** of real numbers by $\mathbb{R}$. We will write for example $(3.5) \in \mathbb{R}$.

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Basic Math - Data Science

We are aware of Scalars: A person's height, weight, salary

Physics vector: velocity (scalar value + direction)

Algebraic vector (in general): Common representation of an entity (1 to n dimension):

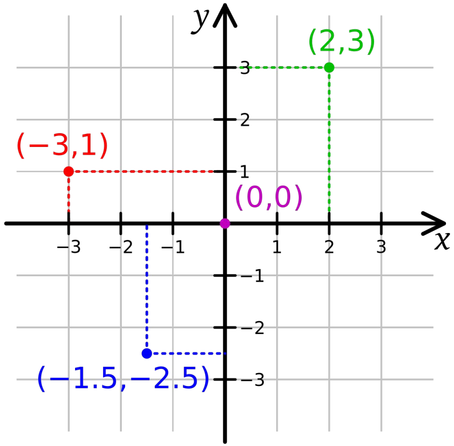
- A person's (height, weight, salary), say [1.78, 72, 100] : once defined, we have to follow it.

$$\mathbf{x} = \begin{bmatrix} 1 \\ 3 \\ 4 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

One hot encoding: Important DS/ML concept

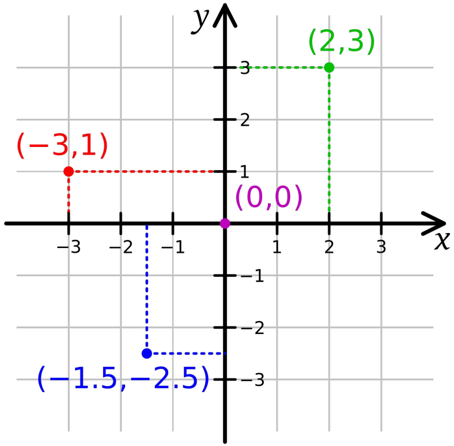
Cartesian coordinate system

- Lets learn some basic ML modelling
    - k-NN



Cartesian coordinate system

- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2



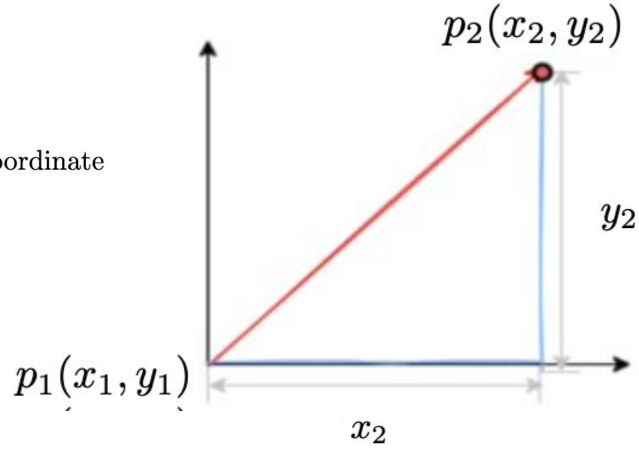
Cartesian coordinate system

L1 distance: The L1 distance between point $p_2(x_2, y_2)$ and $p_1(x_1, y_1)$ is :

$$|x_2 - x_1| + |y_2 - y_1|$$

$$= x_2 + y_2 \quad \text{given that } p_1(x_1, y_1) = (0, 0), \text{ the origin of the coordinate}$$

I.e. L1 distance is the summation of the **horizontal** and the **vertical** sides of a triangle at the right.



Cartesian coordinate system

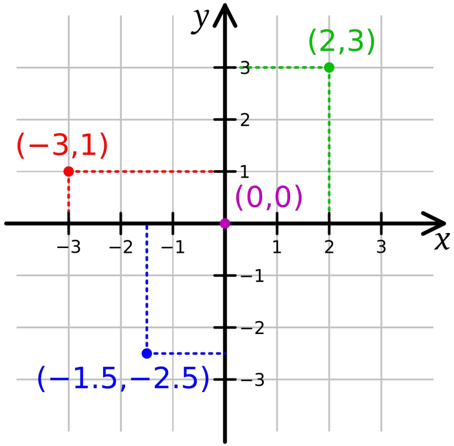
- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2

- L1 distance between vectors [2, 3] and [0, 0] is:

  |2-0| + |3-0| = 5

- L1 distance between vectors [2, 3] and [-3, 1] is :

  |2 - (-3)| + |3 - 1| = 5 + 2 = 7

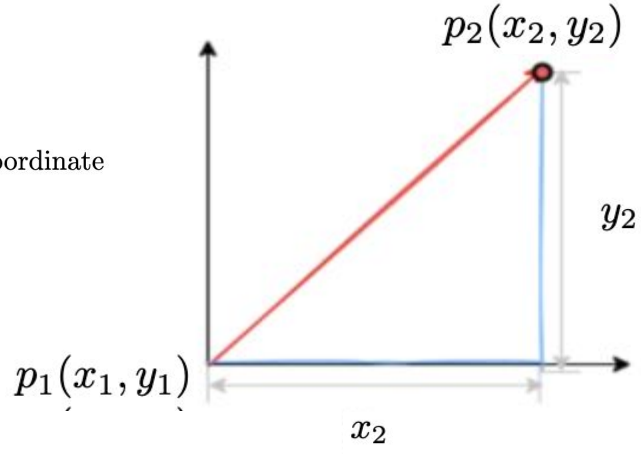


Cartesian coordinate system

L2 distance: The L1 distance between point $p_2(x_2, y_2)$ and $p_1(x_1, y_1)$ is :

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

$$= \sqrt{x_2^2 + y_2^2} \quad \text{given that } p_1(x_1, y_1) = (0, 0)\text{, the origin of the coordinate}$$



I.e. **L2 distance** is the **diagonal** side of a triangle at the right, also known as **Euclidean distance**

Cartesian coordinate system

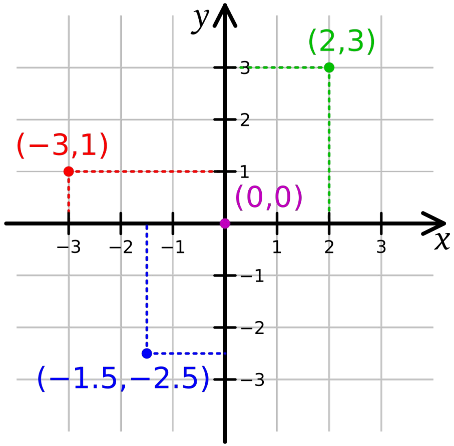
- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2

- L2 distance between vectors [2, 3] and [0, 0] is:

$$\sqrt{(2-0)^2+(3-0)^2}=\sqrt{13}=3.61$$

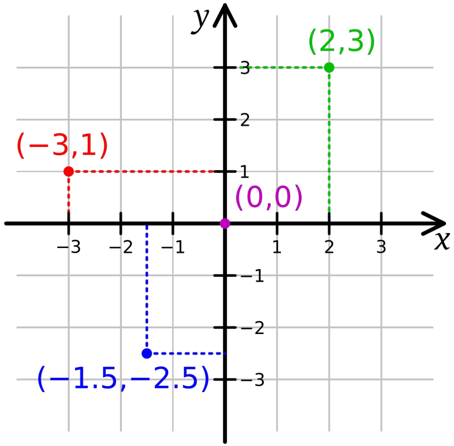
- L2 distance between vectors [2, 3] and [-3, 1] is:

$$\sqrt{(2-(-3)^2+(3-1)^2}=\sqrt{29}=5.39$$



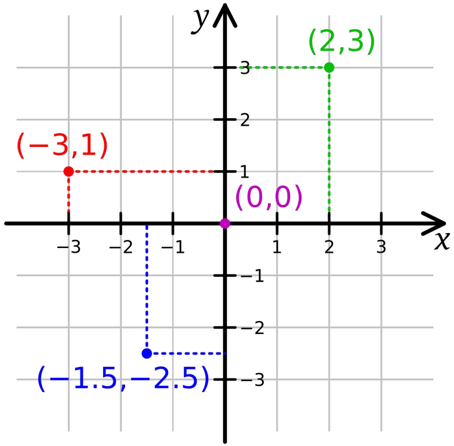
Cartesian coordinate system

- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2
- K-nearest neighbors (k-NN)
  - Supervised learning



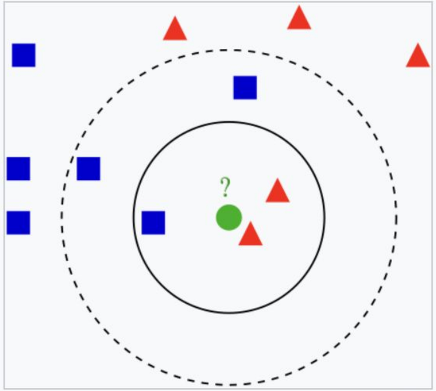
Cartesian coordinate system

- Lets learn some basic ML modelling
    - k-NN
- Distances: L1, L2
- K-nearest neighbors (k-NN)
    - Supervised learning
    - Non parametric



Cartesian coordinate system

- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2
- K-nearest neighbors (k-NN)
  - Supervised learning
  - Non parametric (distance based method)
  - Both for Classification and Regression solutions



Example of *k*-NN classification. The test sample (green dot) should be classified either to blue squares or to red triangles. If $k = 3$ (solid line circle) it is assigned to the red triangles because there are 2 triangles and only 1 square inside the inner circle. If $k = 5$ (dashed line circle) it is assigned to the blue squares (3 squares vs. 2 triangles inside the outer circle).

src: https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

Basic Math - Data Science

Vector operation rules

1. $\mathbf{x}+\mathbf{y} \in \mathbb{R}^n$
2. $\alpha \cdot \mathbf{x} \in \mathbb{R}^n$
3. $\mathbf{x}+\mathbf{y} = \mathbf{y}+\mathbf{x} \in \mathbb{R}^n$ *(commutativity)*
4. $\alpha \cdot (\mathbf{x}+\mathbf{y}) = \alpha \cdot \mathbf{x} + \alpha \cdot \mathbf{y}$ *(distributivity)*
5. $(\alpha+\beta) \cdot \mathbf{x} = \alpha \cdot \mathbf{x} + \beta \cdot \mathbf{x}$ *(distributivity)*
6. $(\mathbf{x}+\mathbf{y})+\mathbf{z} = \mathbf{x}+(\mathbf{y}+\mathbf{z})$ *(associativity)*
7. $(\alpha\beta) \cdot \mathbf{x} = \alpha \cdot (\beta \cdot \mathbf{x})$ *(associativity)*

Basic Math - Data Science

Vector Operation

1.1.2. Vector Addition

Addition of vectors is defined:

$$\mathbf{x}+\mathbf{y}=\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}+\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}=\begin{bmatrix} x_1+y_1 \\ x_2+y_2 \\ \vdots \\ x_n+y_n \end{bmatrix}$$

Example:

$$\mathbf{x}+\mathbf{y}=\begin{bmatrix} 2 \\ 6 \\ -5 \end{bmatrix}+\begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}=\begin{bmatrix} 2 \\ 9 \\ -1 \end{bmatrix}$$

Basic Math - Data Science

Vector Operation

1.1.4. Zero Vector

The **zero** vector **sometimes denoted 0** is a vector having all elements equal to zero, e.g., the 2-dimensional **0** vector:

$$\mathbf{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{A.7}$$

Basic Math - Data Science

Vector Operation

1.1.9. Inner Product

The **inner** or **dot** product of two vectors **x** and **y of the same dimension** is a **scalar** defined by:

$$\mathbf{x}^T \cdot \mathbf{y} = (\mathbf{x}, \mathbf{y}) = x_1 y_1 + x_2 y_2 + \cdots + x_n y_n = \sum_{i=1}^{n} x_i y_i \tag{A.11}$$

Note that the inner product of vector **x** and **y** requires that a transposed vector **x** be multiplied by the **y** vector. Sometimes the inner product is denoted simply by juxtaposition of the vectors x and y, for example, as $< \mathbf{x}, \mathbf{y} >$ or $(\mathbf{x}, \mathbf{y})$.

Example: The inner product of two vectors $\mathbf{x} = \begin{bmatrix} 4 \\ 1 \\ 7 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 0 \\ 2 \\ -3 \end{bmatrix}$

$$\mathbf{x}^T \mathbf{y} = [4\,1\,7]^T \begin{bmatrix} 0 \\ 2 \\ -3 \end{bmatrix} = 4 \cdot 0 + 1 \cdot 2 + 7 \cdot (-3) = 19$$

Basic Math - Data Science

Vector Operation

1.1.10. Orthogonal Vectors

Two vectors **x** and **y** are said to be **orthogonal** if their inner product is equal to zero

$$\mathbf{x}^T\mathbf{y} = 0 \tag{A.12}$$

here 0 is a scalar.

Example: Two vectors $\mathbf{x} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ and are orthogonal, since their inner product is equal to zero

$$\mathbf{x}^T \cdot \mathbf{y} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}^T = \begin{bmatrix} 0 \; 2 \end{bmatrix} = 4 \cdot 0 + 0 \cdot 2 = 0$$

Basic Math - Data Science

Vector Operation

1.1.10. Orthogonal Vectors

Two vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be **orthogonal** if their inner product is equal to zero

$$\mathbf{x}^T\mathbf{y} = 0 \tag{A.12}$$

here 0 is a scalar.

Example: Two vectors $\mathbf{x} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}$ and $\mathbf{y} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}$ and are orthogonal, since their inner product is equal to zero

$$\mathbf{x}^T \cdot \mathbf{y} = \begin{bmatrix} 4 \\ 0 \end{bmatrix}^T = \begin{bmatrix} 0 \; 2 \end{bmatrix} = 4 \cdot 0 + 0 \cdot 2 = 0$$

Basic Math - Data Science

Vector Operation

1.1.11. Vector Norm

The magnitude of a vector may be measure in different ways. One method, called the vector **norm**, is a function from $\mathbb{R}^n$ into $\mathbb{R}$ for $\mathbf{x}$ an element of $\mathbb{R}^n$. It is denoted $||\mathbf{x}||$ and satisfies the following conditions:

1. $||\mathbf{x}|| \geq 0$, and the equality holds if and only if $\mathbf{x} = \mathbf{0}$
2. $||\alpha\mathbf{x}|| = |\alpha| \cdot ||\mathbf{x}||$, where $|\alpha|$ is the absolute value of scalar α

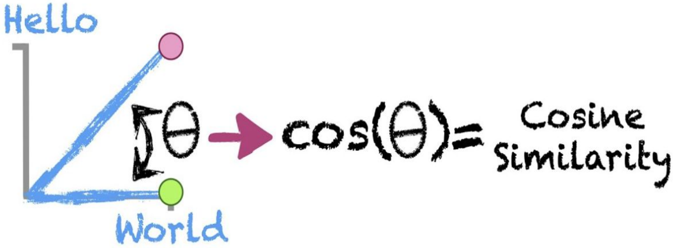
and is defined as:

$$||\mathbf{x}|| = \sqrt{\mathbf{x}^T\mathbf{x}} = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2} \tag{A.13}$$

Example: For the vector $\mathbf{x} = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$ the norm is

$$||\mathbf{x}|| = \sqrt{\mathbf{x}^T\mathbf{x}} = \sqrt{4^2 + 3^2} = 5$$

Cosine distance



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

$$\|\vec{a}\| = \sqrt{a_1^2 + a_2^2 + a_3^2 + \cdots + a_n^2}$$

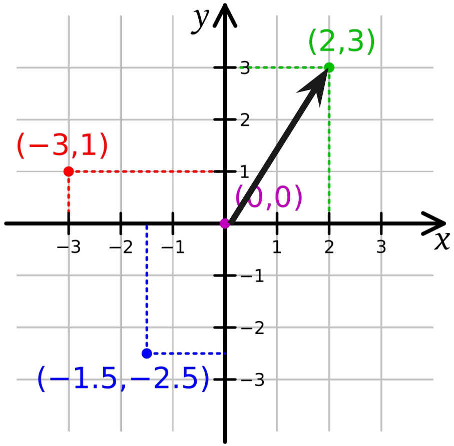
$$\|\vec{b}\| = \sqrt{b_1^2 + b_2^2 + b_3^2 + \cdots + b_n^2}$$

Cartesian coordinate system

- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2, Cosine

- Cosine distance between vectors [2, 3] and [0, 0] is:

0.00

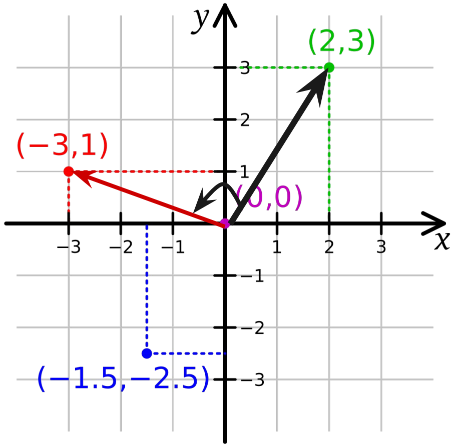


Cartesian coordinate system

- Lets learn some basic ML modelling
  - k-NN
- Distances: L1, L2, Cosine

- Cosine distance between vectors [2, 3] and [-3, 1] is :

$$\frac{-3}{\sqrt{13}\sqrt{10}} = -0.26$$



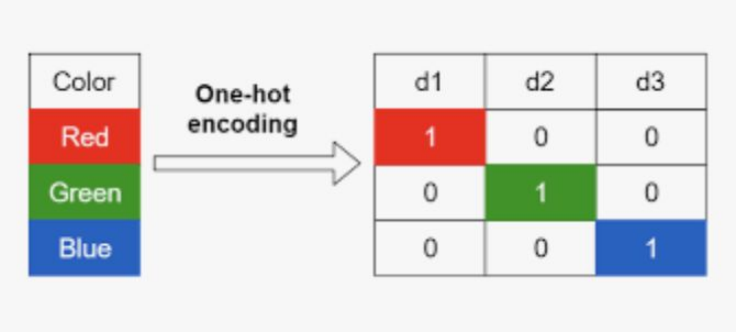
Digital data

- In computing everything is digital and binary
- All data types we talked about
- Bit(0 / 1): Digital letter
- Byte (000 0011): Digital word
- Kilo (Byte), Mega(Byte), Giga (Byte): We are talking about Digital data and their sizes mainly

Categorical Data

One hot encoding

- Only one bit is 1
- A vector representation of categorical values



One hot encoding (cont.)

Classification task:

- Binary example {Cat vs Dog}
- Set size is 2
  - Cat (0, 1)
  - Dog (1, 0)
  - Or vice versa
- Same rule applies every categorical data

Binary, gray-scale, and color images