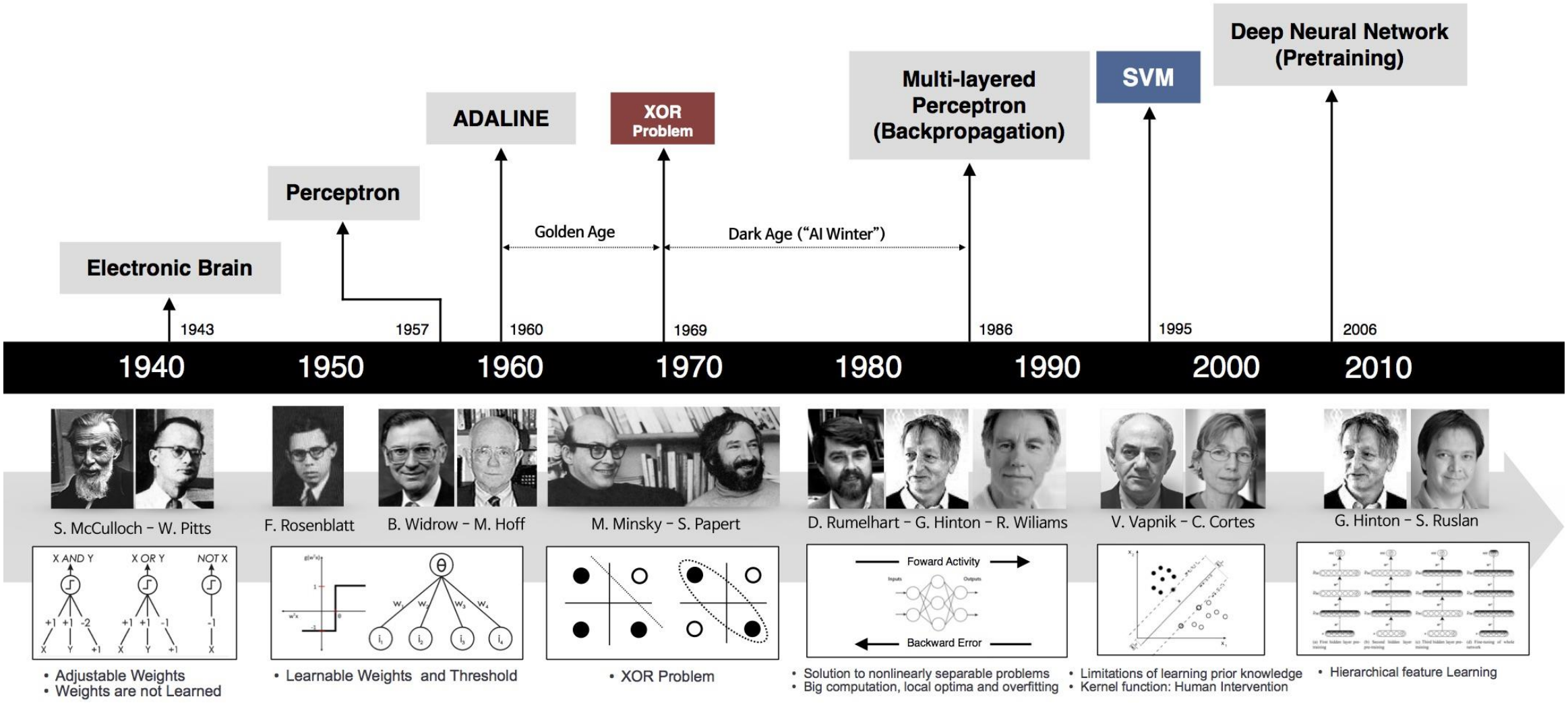
Artificial Neural Networks

Recap for midterm exam

Outline

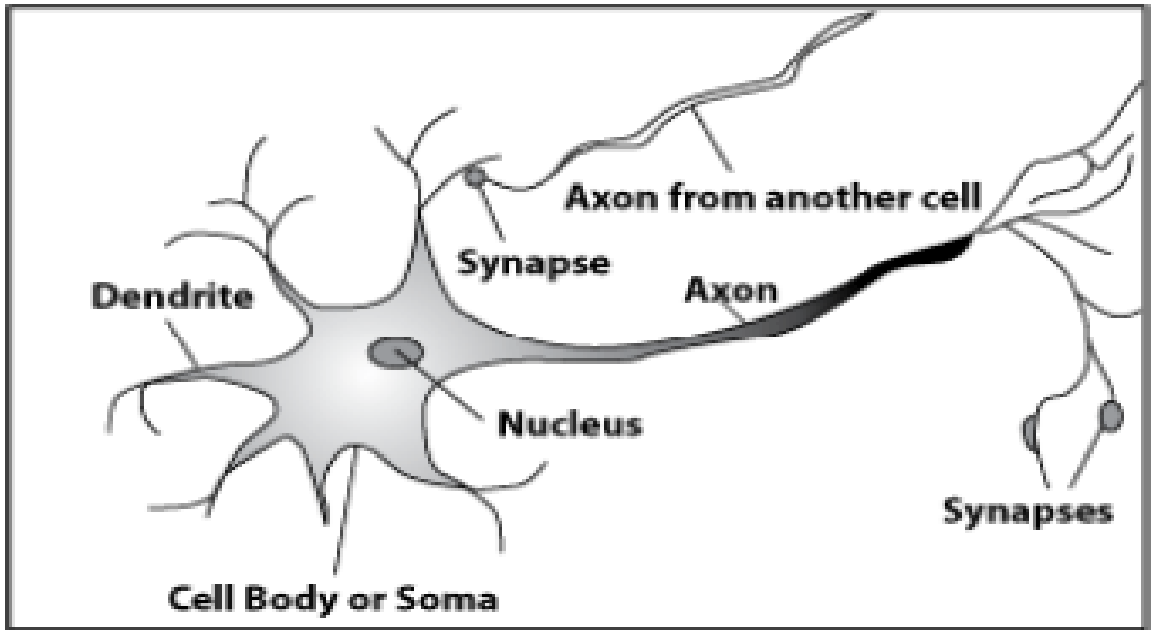
- **Artificial Neural Networks**
  - Neurons - Biological motivation
  - Linear threshold units
  - Weights, bias, activation functions
  - MCP neuron model
  - Perceptron model
    - Representational limitation and gradient descent training
  - Multilayer Networks
  - Back Propagation
    - Hidden layers
    - Constructing intermediate & distributed representations

History of Artificial Neural Networks



Biological Neural Networks

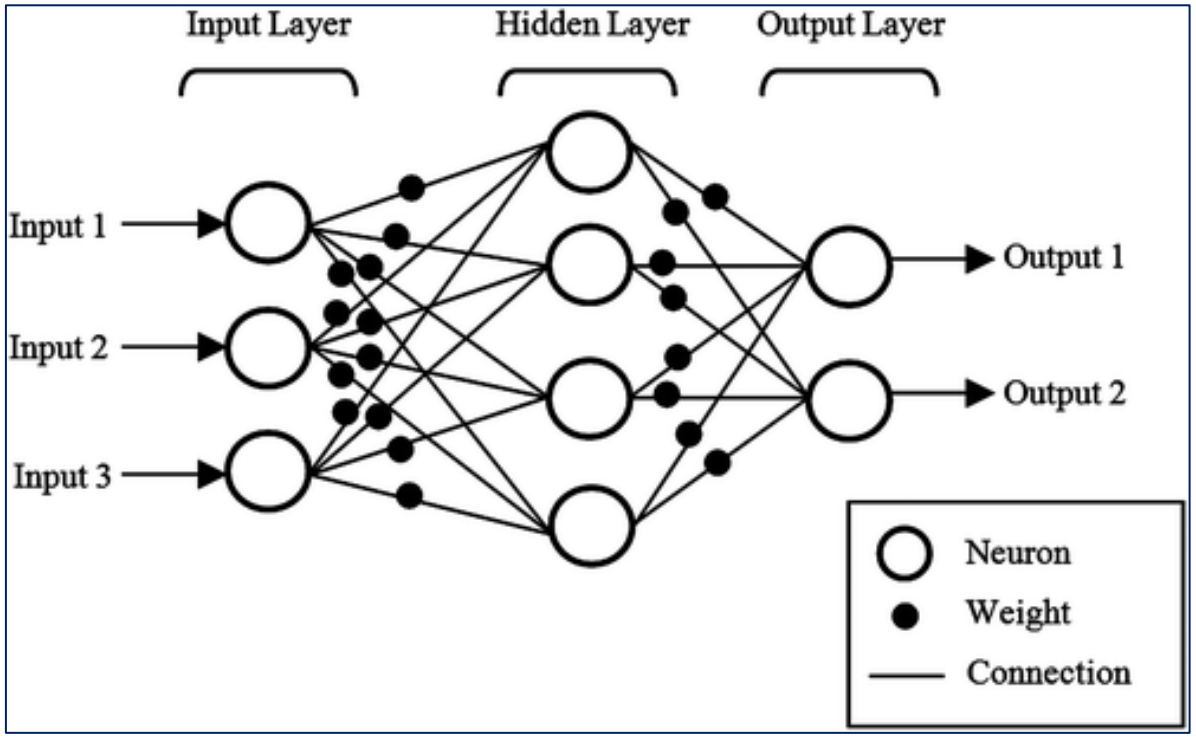
- In biology, a neuron is a cell that can transmit and process chemical or electrical signals.
- A neuron is connected with other neurons to create a network.
- Tens of billions of interconnected neuron structures – in human brain.
- Every neuron has an
  - Input called the dendrite
  - Cell body called Soma
  - Output called the axon



Artificial Neural Network

Artificial Neural Network

- Pool of simple processing units
- Communication to each other over a large number of weighted connections.



Biological Vs Artificial Neural Networks

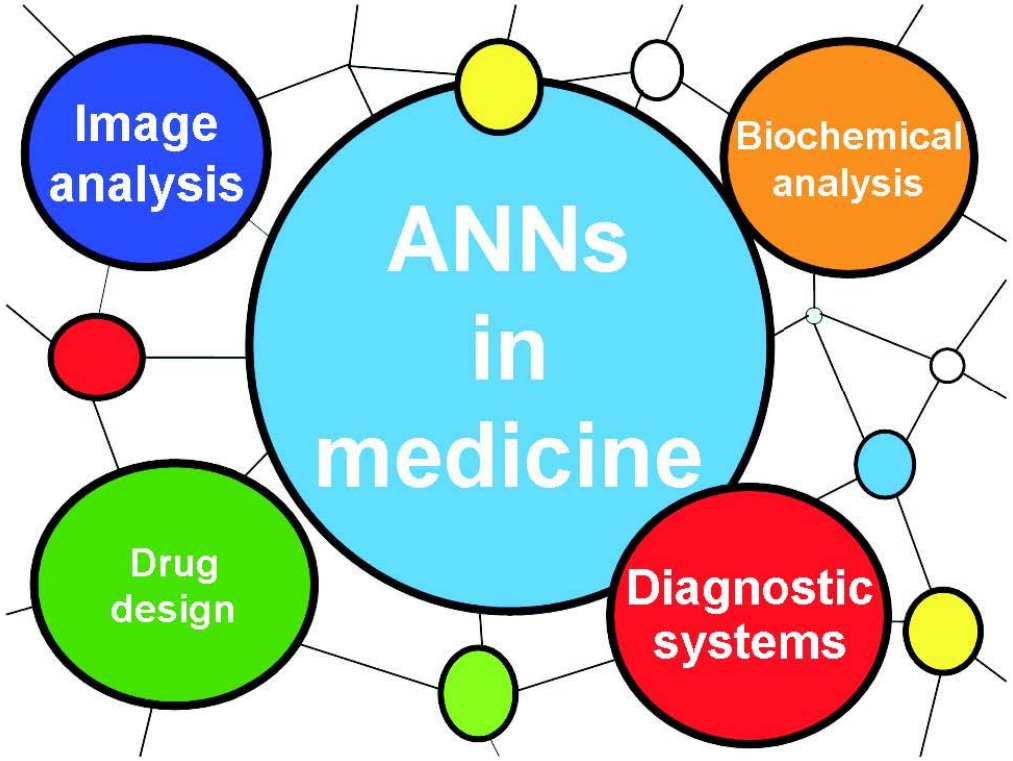
	
Biological neurons or nerve cells	Silicon transistors
200 billion neurons, 32 trillion interconnections.	1 billion bytes RAM, trillion of bytes on disk.
Neuron size: 10-6 m.	Single transistor size: 10-9m.
Energy consumption: 6-10 joules per operation per sec.	Energy consumption: 10-16 joules per operation per second.
Learning capability	Programming capability

Applications of ANN

- Inspired by biological neural networks
  - Numerous advances have been made in developing intelligent systems.

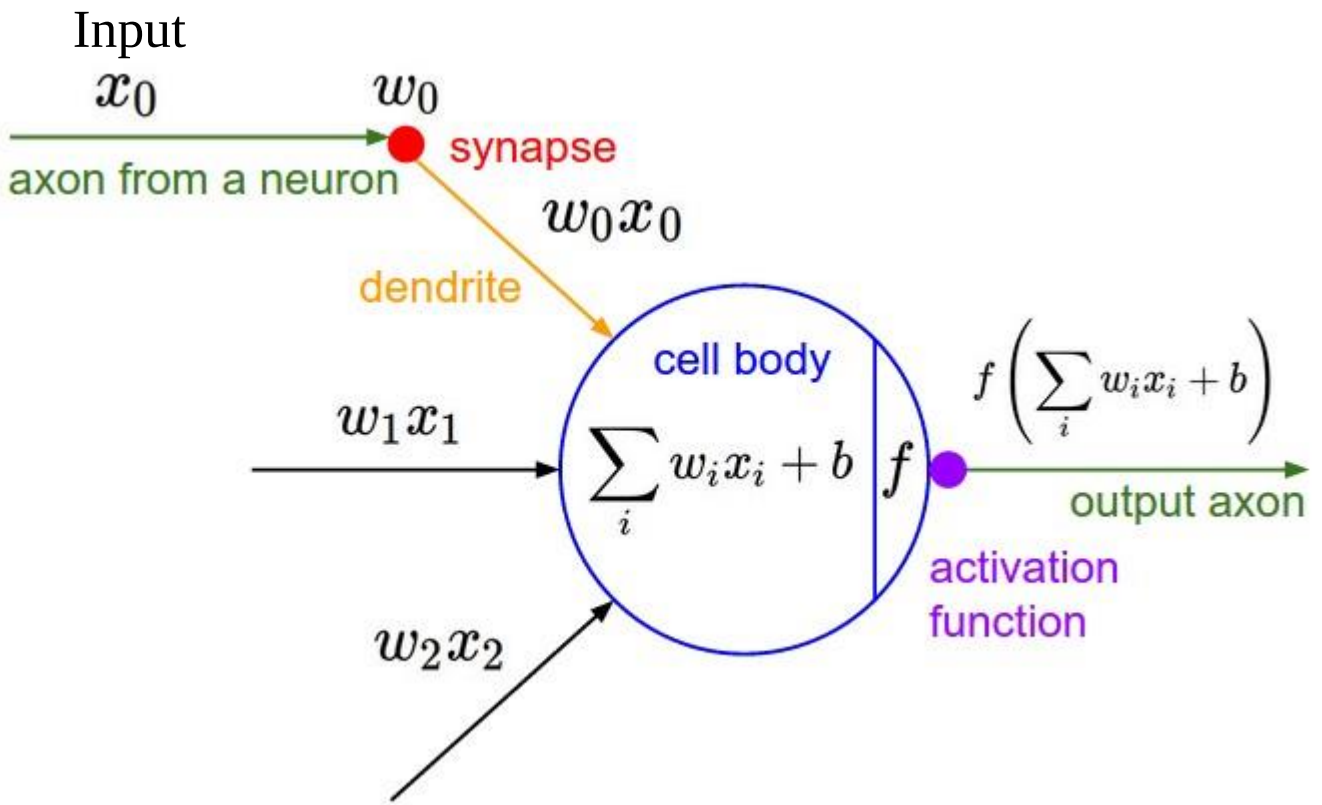
- ANNs developed to solve a variety of problems in
  - pattern recognition
  - prediction
  - optimization
  - associative memory

Example: ANN in medical Applications



Computational Model of ANNs

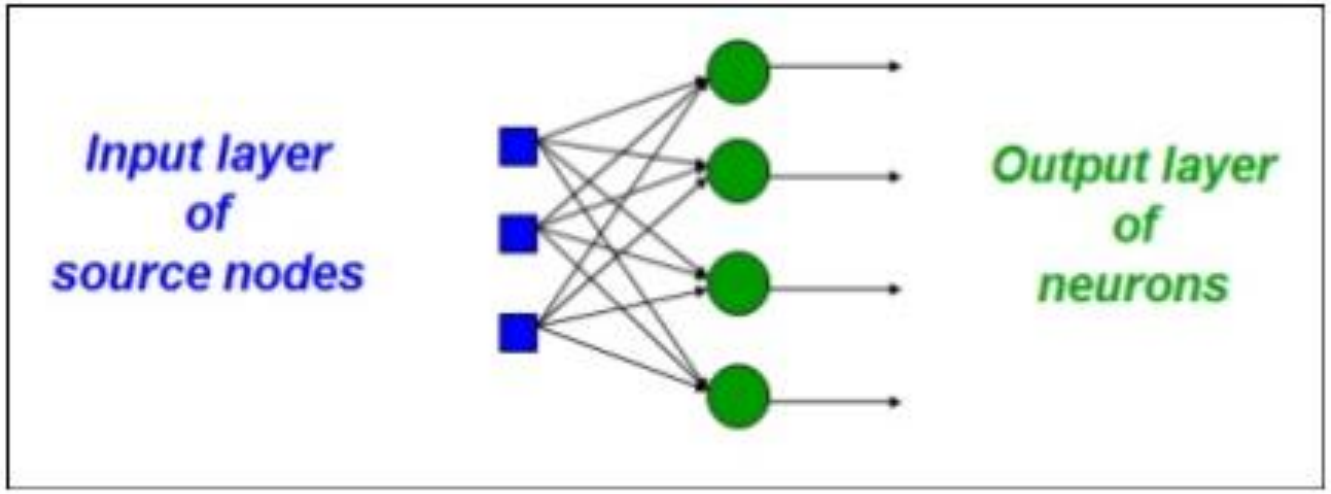
- ANN is an information processing system
  - Consists of many nodes called **neurons** (processing units)
  - Signals are transmitted by connection **links**
  - Links possess an associated **weight**, which is multiplied with input signal (net input)
  - Output is obtained by applying **activations** to net input.



ANN Models

- ANN Models - Classified
  - Single Layer ANN
  - Multi-layer ANN

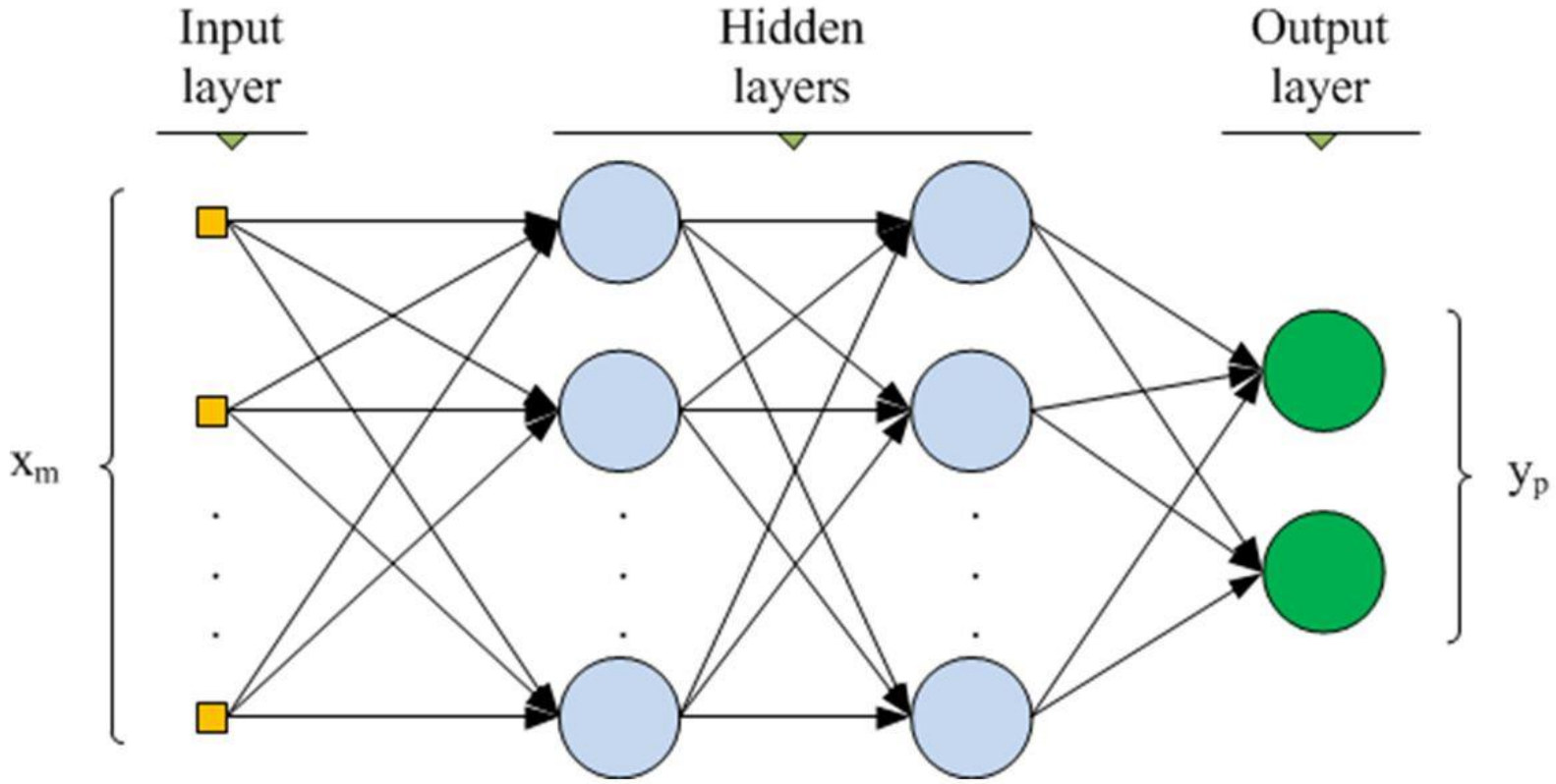
Single Layer ANN



- Only one layer of weighted interconnections
- Weighted input(s) are processed by only one layer and provide output(s)

Multilayer ANN

- Multilayer ANN are called layered networks
  - Input layer
  - Hidden layer(s)
  - Output layer

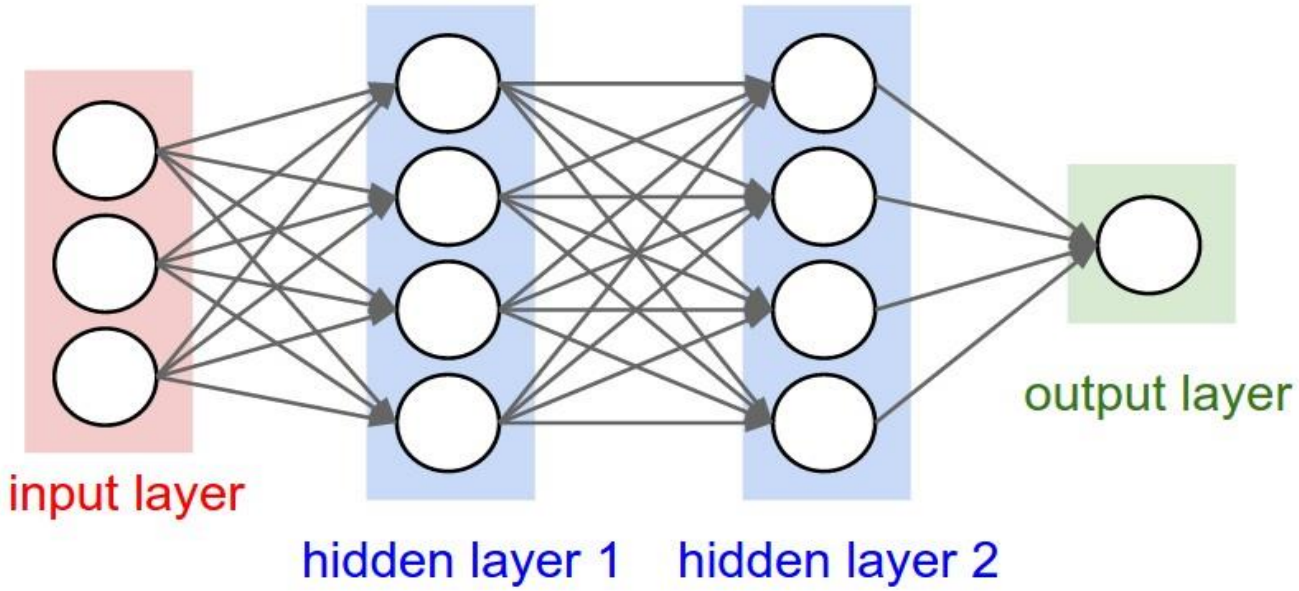


Basic building blocks of ANN

- Network Architecture
- Setting weights
- Activation function

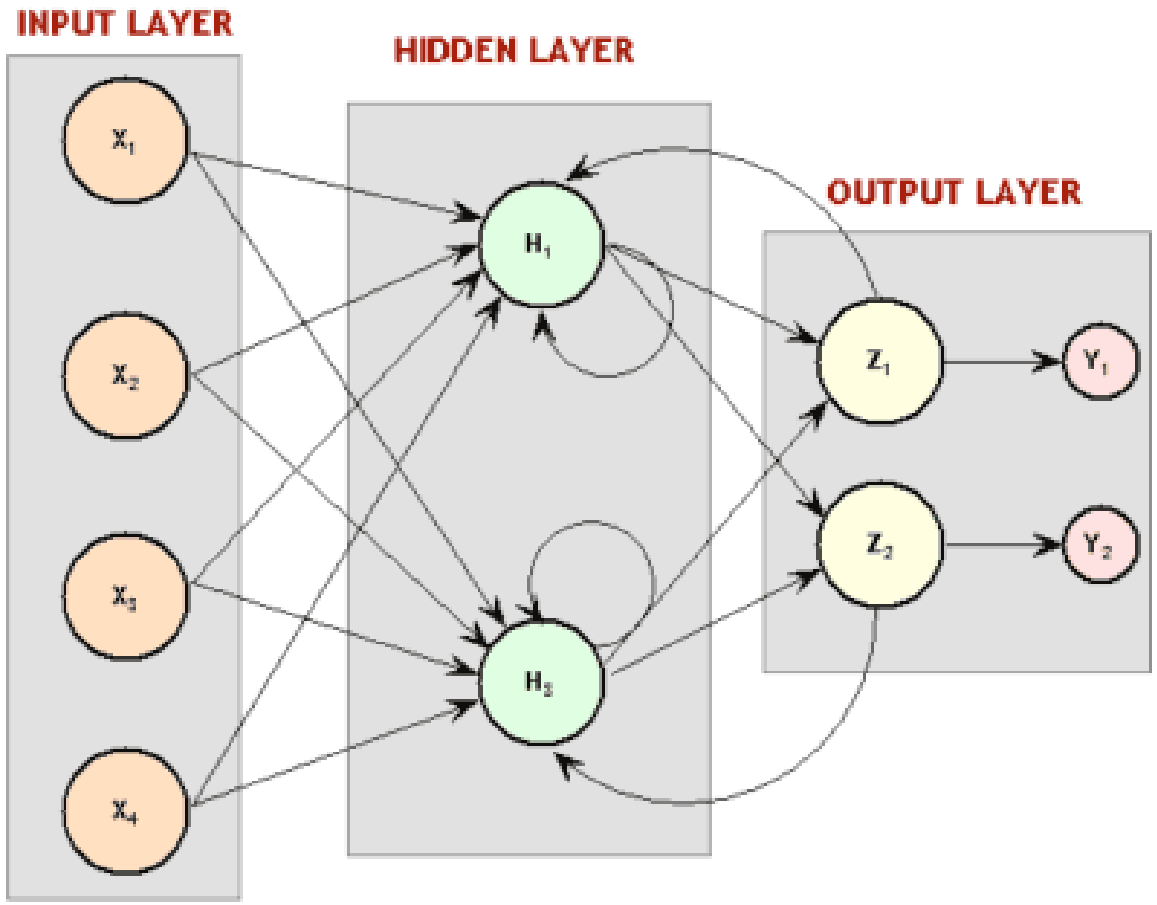
- Network Architecture
  - Arrangement of neurons into layers
  - Connection pattern between neurons

Feed Forward Neural Networks



- The information is propagated from the inputs to the outputs
- Time has no role (NO cycle between outputs and inputs)

Recurrent Neural Networks



- All nodes are connected to all other nodes
- Every node is both input/ output node
- Delays are associated
- Training is more difficult
- Performance may be problematic
  - Stable Outputs may be more difficult

Setting weights

- Setting the values for weights – enable learning/ training

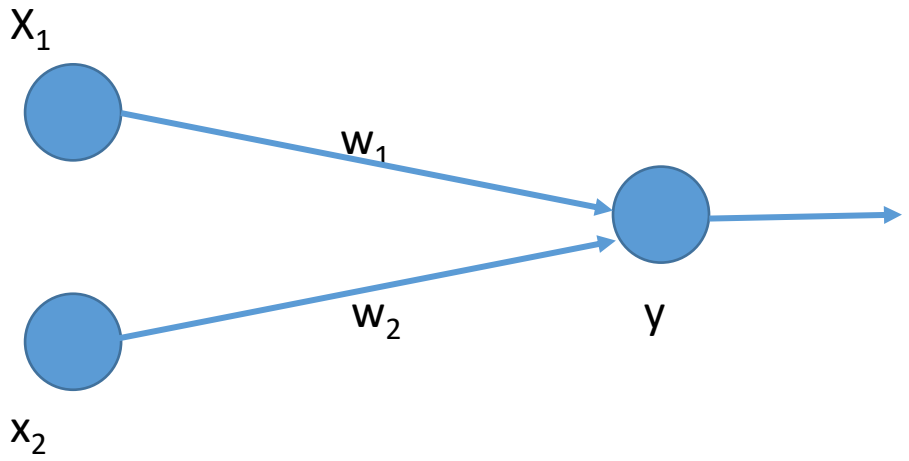
- Training
  - Process of modifying the weights in the network to achieve the expected output

- Learning
  - Internal process when the network is trained

Artificial Neural Network (ANN) Terminologies

1. Weights

- Weight is an information used by the neural net to solve the problem
- Weights can set to zero or can be calculated by some methods



- x_1 – activation of neuron-1 (input signal)
- x_2 – activation of neuron-2 (input signal)
- y – output neuron
- w_1 – weight connecting neuron-1 to output
- w_2 –weight connecting neuron-2 to output

Net input = Net = $x_1 w_1 + x_2 w_2$

Net input = Net = $\sum_{i=1}^{n} x_i w_i$

Artificial Neural Network (ANN) Terminologies

2. Activation functions

- Used to calculate the output response of a neuron.
- Sum of the weighted input signal is applied with an activation to obtain the response.
- For neurons in the same layer same activation functions are used.

- Activation function
  - Linear
  - Non-linear (used in multilayer net)

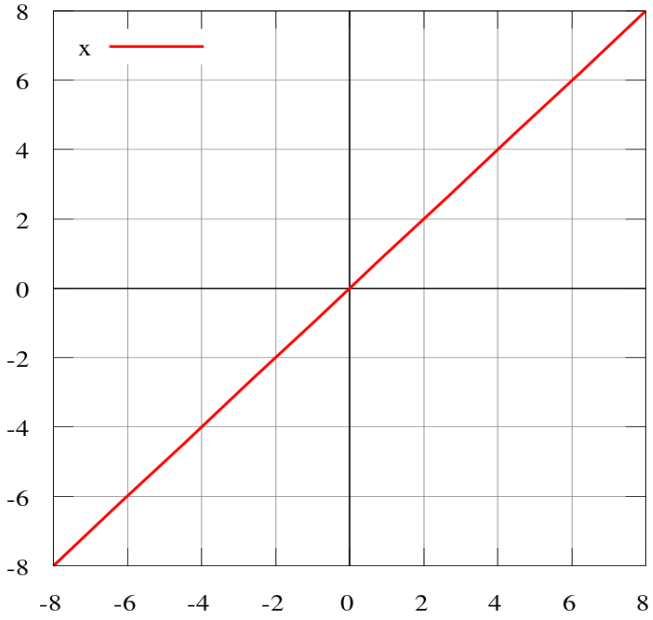
θ

Artificial Neural Network (ANN) Terminologies

2. Activation functions

Identity (Linear) Function:

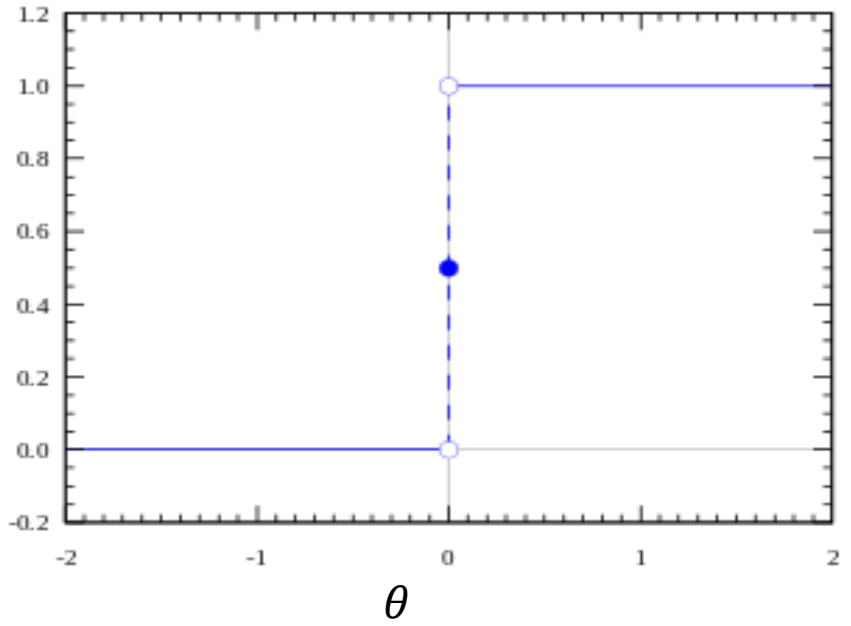
f(x) = x, for all x



Binary Step Function

$$f(x) = \begin{cases} 1; if\ f(x) \geq \theta \\ 0; if\ f(x) < \theta \end{cases}$$

- $where\ \theta\ is\ threshold$
- Single layer nets uses binary step (threshold) function.



Artificial Neural Network (ANN) Terminologies

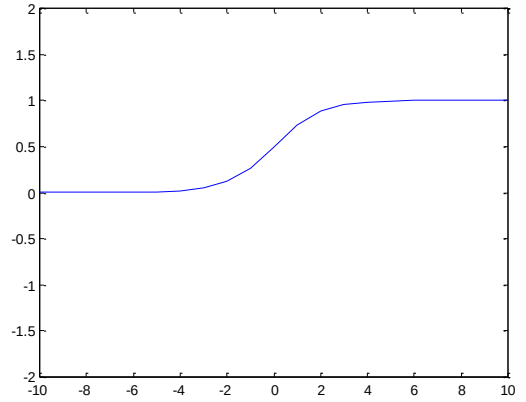
2. Activation functions

- **Sigmoidal function**
  - S-shaped curves
  - Hyperbolic & logistic functions
  - Used in multi layer nets
  - Example: back propagation net
  - Types
    - Binary Sigmoidal Function
    - Bipolar Sigmoidal Function

Binary Sigmoidal (Logistic) Function

It ranges between 0 to 1.

$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$



Where σ – steepness parameter.

Artificial Neural Network (ANN) Terminologies

2. Activation functions

- Bipolar Sigmoidal Function
  - Range: +1 and -1
  - Called hyperbolic tangent function

$$b(x) = 2f(x) - 1$$

$$b(x) = 2 \times \frac{1}{1 + \exp(-\sigma x)} - 1$$

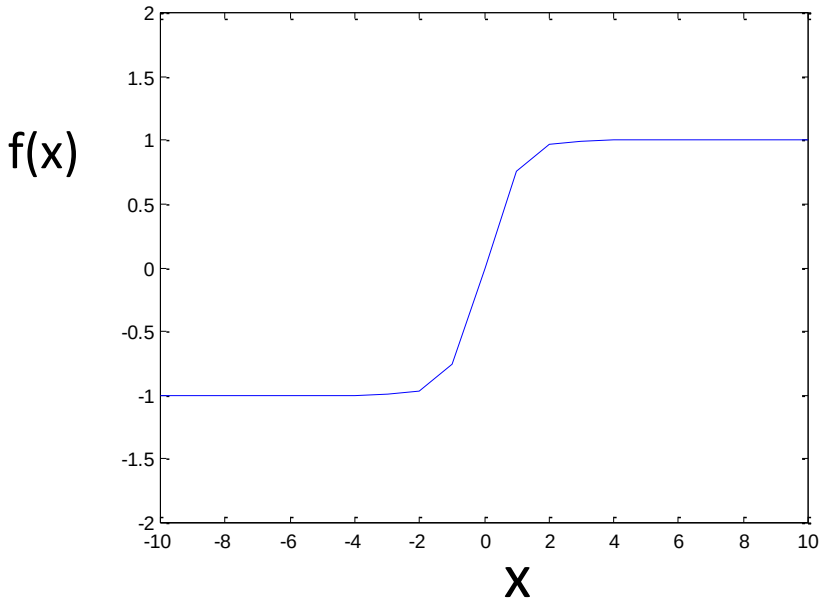
$$b(x) = \frac{2 - 1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)} - 1$$

$$b(x) = \frac{1 - \exp(-\sigma x)}{1 + \exp(-\sigma x)}$$

In binary Sigmoidal Function

$$f(x) = \frac{1}{1 + \exp(-\sigma x)}$$

Where σ – steepness parameter.



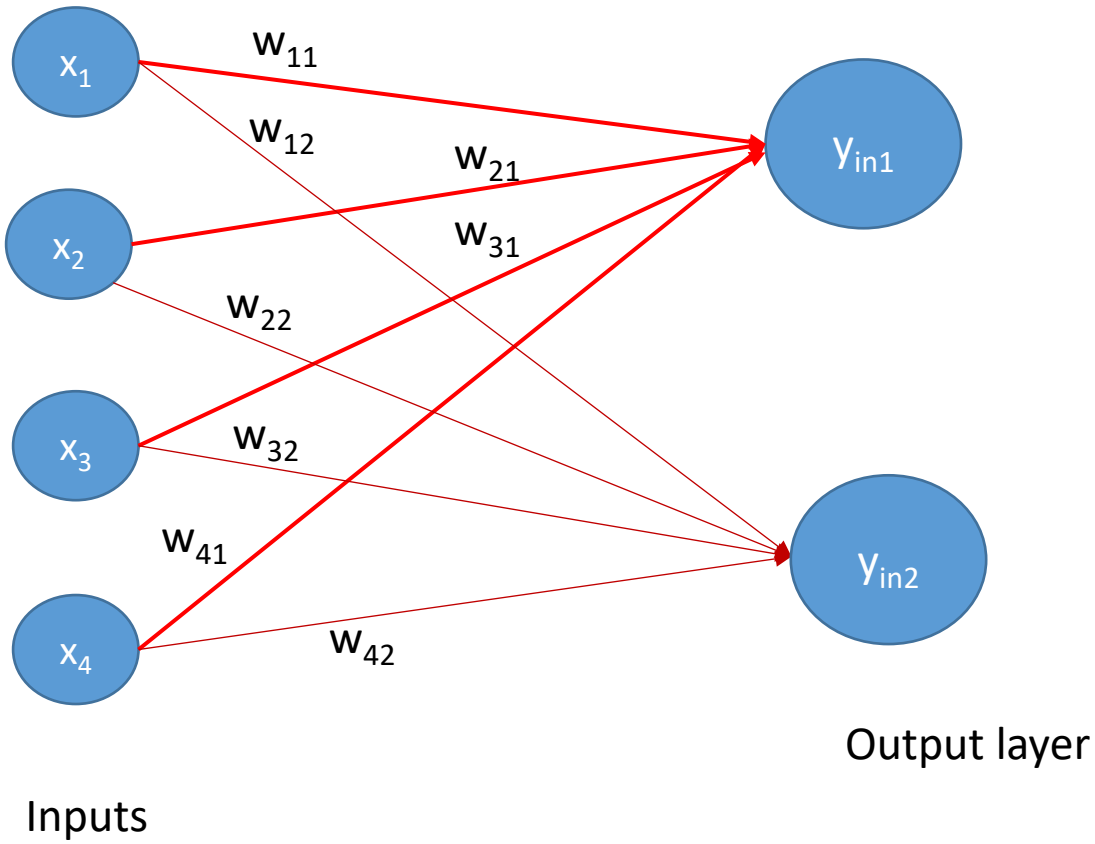
Artificial Neural Network (ANN) Terminologies

3. Calculation of Net Input

- If the weights are given as $W = (w_{ij})$ in a matrix form

- The net input to output $y_{inj} = x_i * w_{ij}$

$$y_{inj} = \sum_{i=1}^{n} x_i \, w_{ij}$$

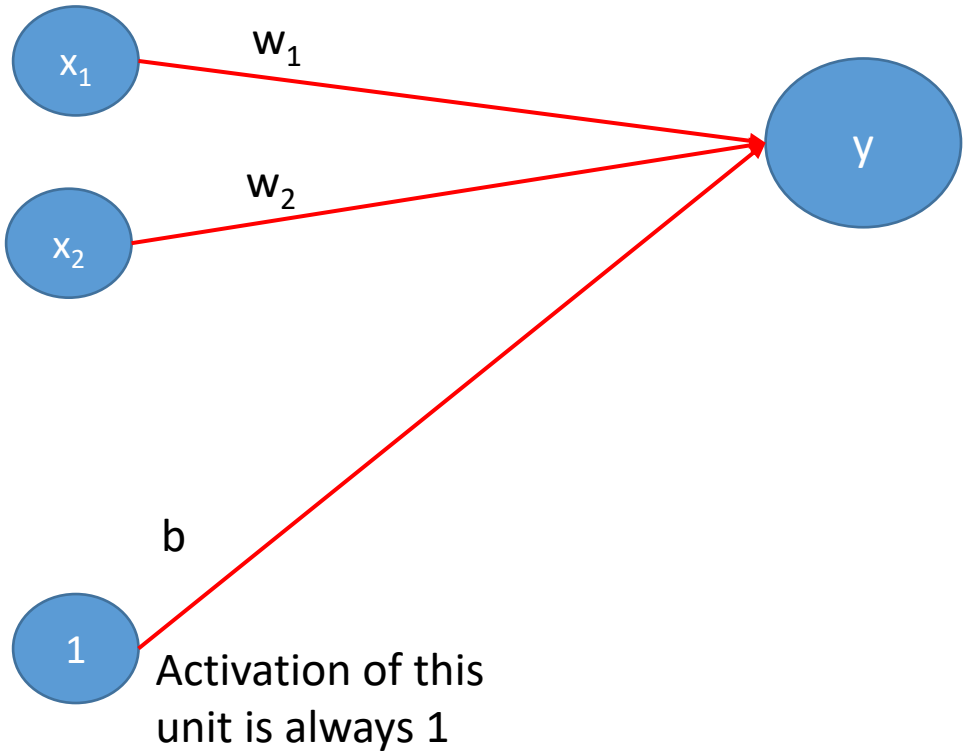


Artificial Neural Network (ANN) Terminologies

4. Bias

- Weight on a connection from a unit whose activation is always 1.
- Increasing bias increase net input.

$$\text{Net} = b + \sum_{i=1}^{n} x_i \, w_i$$



Artificial Neural Network (ANN) Terminologies

5. Threshold

- Used in calculating the activations of the given net
- Based on the threshold, output is calculated.
- Activation function is based on the value of θ

$$y = f(Net) = \begin{cases} +1;\ if\ Net \geq \theta \\ -1; if\ Net < \theta \end{cases}$$

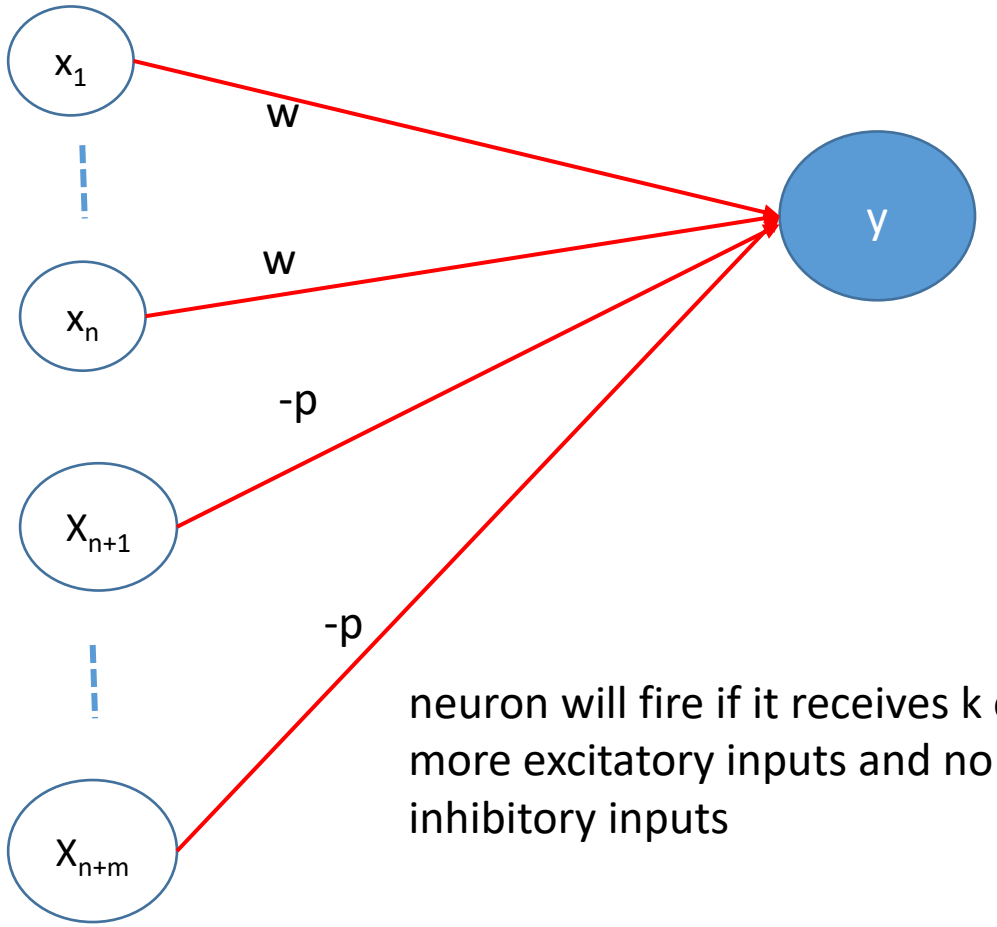
$where\ \theta\ and\ \theta_j\ are\ thresholds$

McCulloch-Pitts Neuron Model

- Here, y is McCulloch-Pitts neuron
- Receives signal from any number of neurons
- Connection weights from x_1 .. x_n are excitatory, denoted by w.
- Connection weights from x_{n+1} .. X_{n+m} are inhibitory, denoted by –p.
- McCulloch-Pitts neuron y has the activation function

$$f(y_{in}) = \begin{cases} 1; if\ f(y_{in}) \geq \theta \\ 0; if\ f(y_{in}) < \theta \end{cases}$$

- Where, θ – threshold and y_{in} - net input to y



neuron will fire if it receives k or more excitatory inputs and no inhibitory inputs

Examples

- Generate the output of logic AND function by McCulloch-Pitts neuron model.
- Generate the output of logic OR function by McCulloch-Pitts neuron model.
- Generate the output of logic NOT function by McCulloch-Pitts neuron model.
- Generate the output of logic XOR function by McCulloch-Pitts neuron model.

Perceptron Networks

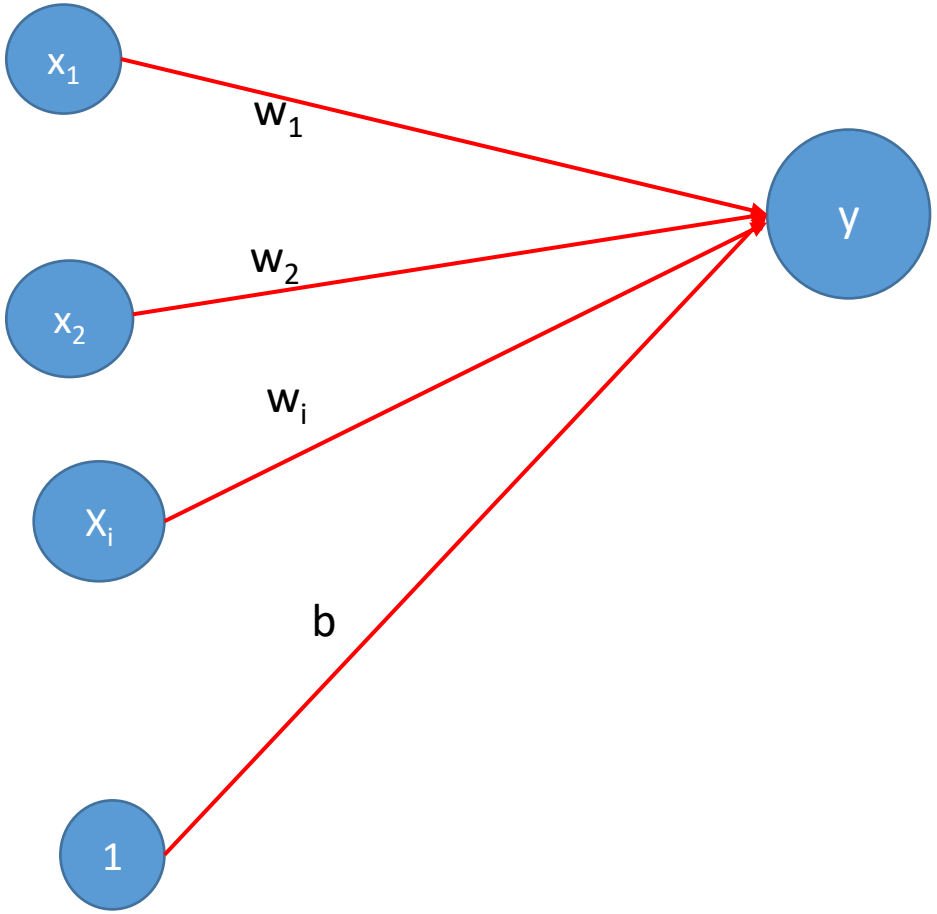
- Frank Rosenblatt; Minsky and Papert - limitations
- Learning rule: iterative weight adjustment (powerful technique)
- Training in perceptron will continue until no error occurs.
- This net solve – classification problem
- Types of perceptron:
  - Single Layer perceptron
  - Multilayer perceptron

Single Layer Perceptron

Algorithm

1. Initialize weights and bias (set to ZERO) and set the learning rate α (0 to 1)
2. While stopping condition is not TRUE, repeat steps 3 to 7
3. For each training pair S: t do steps 4 to 6
4. Set activation of input layers $x_i = s_i$ for all i = 1 to n

Architecture



Single Layer Perceptron Architecture

5. Compute output response

$$y_{in} = b + \sum_{i=1}^{n} x_i \, wi$$

Activation function

$$Y = f(y_{in}) = \begin{cases} +1 & if \; y_{in} > \theta \\ 0 & if \; -\theta \le y_{in} \le \theta \\ -1 & if \; y_{in} < \theta_j \end{cases}$$

6. If the output response is not equal to target, then update the weights and bias

If $t \neq y$

$$w_{i(new)} = w_{i(_{old})} + \alpha t x_i$$

$$b_{(new)} = b_{(_{old})} + \alpha t$$

else

$$w_{i(new)} = w_{i(_{old})}$$

$$b_{(new)} = b_{(_{old})}$$

7. Test for stopping condition

Gradient Descent

- In machine learning, gradient descent is a technique that evaluate and update the weights in every iteration to minimize the error of a model on our training data.

  - Training instances to the input to the model one at a time.
  - Model make a prediction for the training instances
  - Calculate the error
  - Update weights to reduce the error for next prediction

$$w_{i(new)} = w_{i(_{old})} + \alpha t x_i$$

Example

- Develop a perceptron for AND, OR and NOT function with bipolar inputs and targets.

Multilayer Networks

- Feed forward Networks
  - Output is calculated for every input to the network
  - No feedback

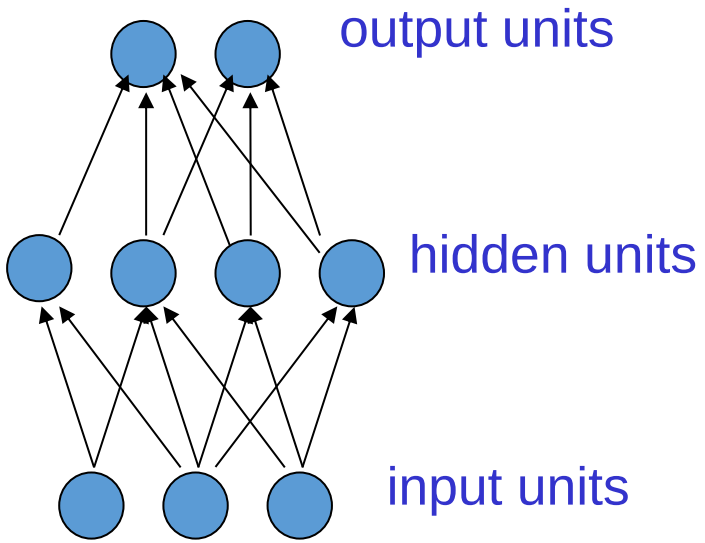
  - Example: Back Propagation Network

- Back Propagation Network (BPN)
  - Multilayer ANN
  - Using gradient descent based delta learning rule known as back propagation (for errors)
  - Efficiently changing the weights with **differentiable activation function** units to learn a training set of input-output examples.

Back Propagation Network (BPN)

Architecture

- These are the commonest type of neural network in practical applications.
  - The first layer is the input and the last layer is the output.
  - If there is more than one hidden layer, we call them "**deep**" neural networks.
- They compute a series of transformations that change the similarities between cases.
  - The activities of the neurons in each layer are a non-linear function of the activities in the layer below.



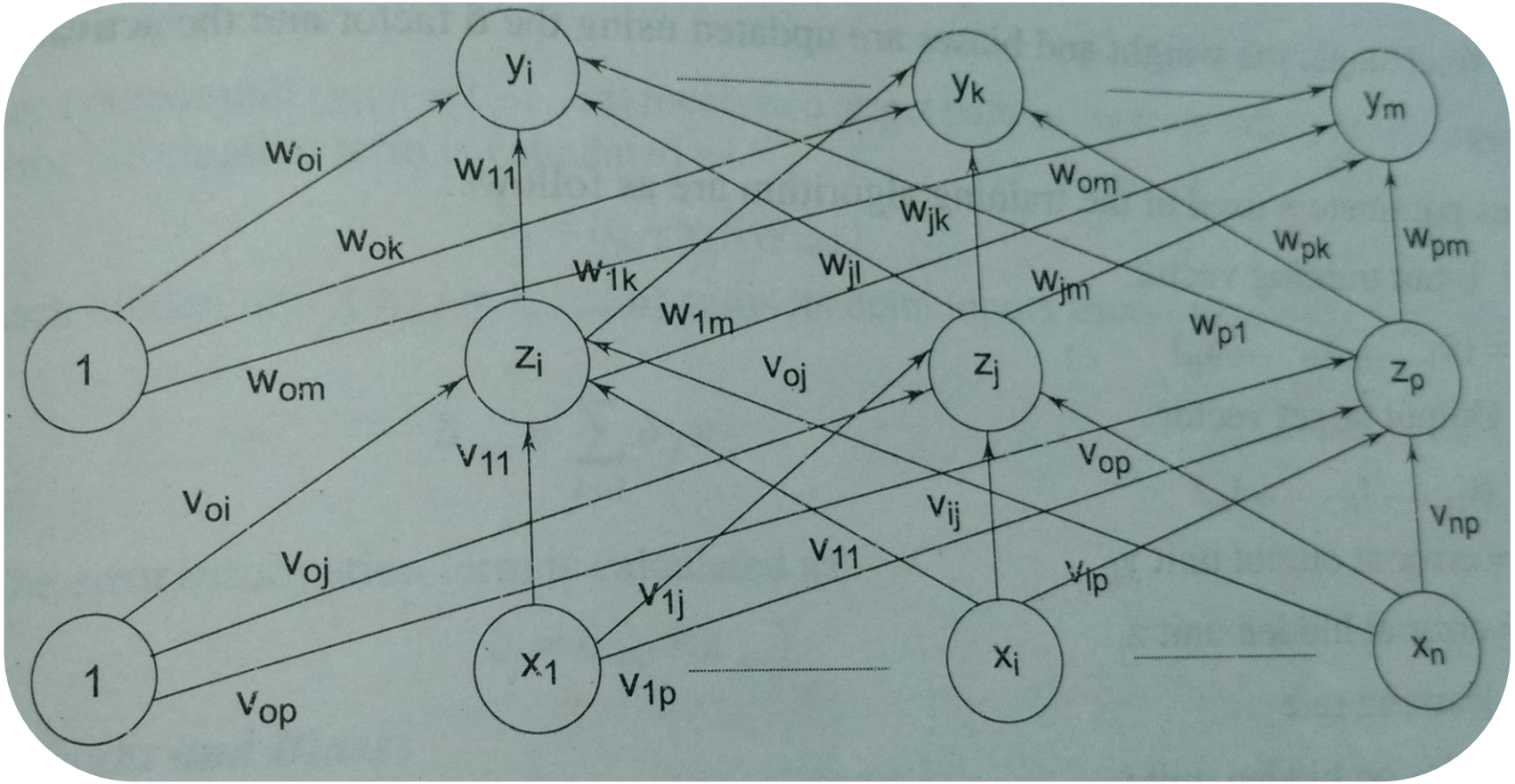
During back propagation learning

- Signals sent in reverse direction also.

Increase in number of hidden layer

- Minimize error
- Increase computational complexity
- Increase time taken for convergence.

Back Propagation Network (BPN)



Parameters

- The various parameters used in the training algorithm are

x: Input training vector

$x = (x_1, \ldots, x_i, \ldots, x_n)$

t: Output target vector

$t = (t_1, \ldots, t_k, \ldots, t_m)$

δ_k = error at output unit y_k

δ_j = error at hidden unit z_j

α = learning rate

V_{oj} = bias on hidden unit j

z_j = hidden unit j

w_{ok} = bias on output unit k

y_k = output unit k.

Training Algorithm

Step 1: Initialize weight to small random values.

Step 2: While stopping condition is false, do steps 3 – 10

Step 3: For each training pair do steps 4 - 9

Training Algorithm

Initialization of Weights

Step 1: Initialize weight to small random values.

Step 2: While stopping condition is false, do Steps 3–10

Step 3: For each training pair do Steps 4–9

Feed Forward

Step 4: Each input unit receives the input signal x_i and transmits this signals to all units in the layer above i.e. hidden units

Step 5: Each hidden unit (z_j, $j = 1, \ldots, p$) sums its weighted input signals

$$z_{-inj} = v_{oj} + \sum_{i=1}^{n} x_i v_{ij}$$

applying activation function

$$Z_j = f(z_{inj})$$

and sends this signal to all units in the layer above i.e output units.

Training Algorithm

Step 6: Each output unit (y_k, $k=1, \ldots, m$) sums its weighted input signals

$$y_{-ink} = w_{ok} + \sum_{j=1}^{p} z_j w_{jk}$$

and applies its activation function to calculate the output signals.

$$Y_k = f(y_{-ink})$$

Training Algorithm

Back Propagation of Errors

Step 7: Each output unit (y_k, $k = 1, \ldots, m$) receives a target pattern corresponding to an input pattern, error information term is calculated as

$$\delta_k = (t_k - y_k) f(y_{-ink})$$

Step 8: Each hidden unit (z_j, $j = 1, \ldots, n$) sums its delta inputs from units in the layer above

$$\delta_{-inj} = \sum_{k=1}^{m} \delta_k w_{jk}$$

The error information term is calculated as

$$\delta_j = \delta_{-inj} f(z_{-inj})$$

Training Algorithm

Updation of weights and biases

Step 9: Each output unit (y_k, $k = 1, \ldots, m$) updates its bias and weights ($j = 0, \ldots, p$)

The weight correction term is given by

$$\Delta W_{jk} = \alpha \delta_k z_j$$

and the bias correction term is given by

$$\Delta W_{ok} = \alpha \delta_k$$

Therefore, $W_{jk}(\text{new}) = W_{jk}(\text{old}) + \Delta W_{jk}$, $W_{ok}(\text{new}) = W_{ok}(\text{old}) + \Delta W_{ok}$

Each hidden unit (z_j, $j = 1, \ldots, p$) updates its bias and weights ($i = 0, \ldots n$)

The weight correction term

$$\Delta V_{ij} = \alpha \delta_j x_i$$

The bias correction term

$$\Delta V_{oj} = \alpha \delta_j$$

Therefore, $V_{ij}(\text{new}) = V_{ij}(\text{old}) + \Delta V_{ij}$, $V_{oj}(\text{new}) = V_{oj}(\text{old}) + \Delta V_{oj}$

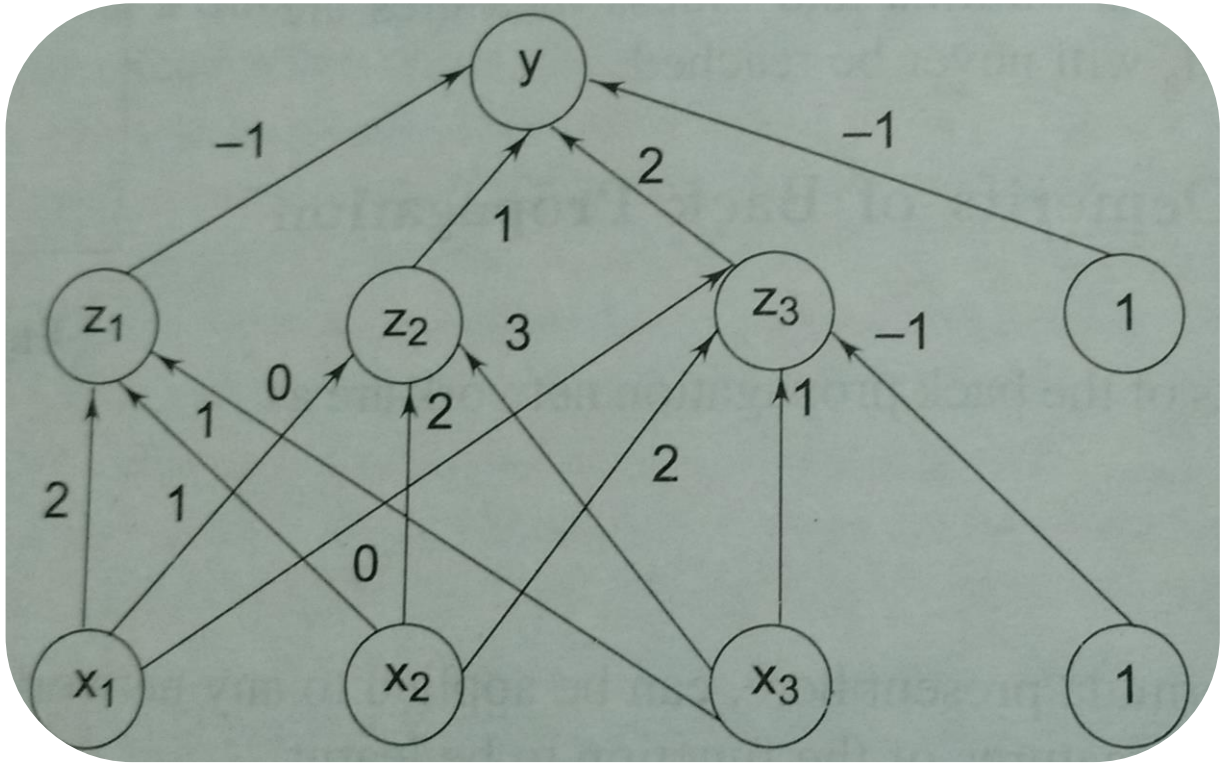
Training Algorithm

Step 10: Test for stopping condition:

The stopping condition may be the minimization of errors, number of epochs etc.

Example

- Find the new weights when the network given in the figure is presented the input pattern [0.6 0.8 0] and the target output is 0.9. use the learning rate α = 0.3 and use binary sigmoid activation function.

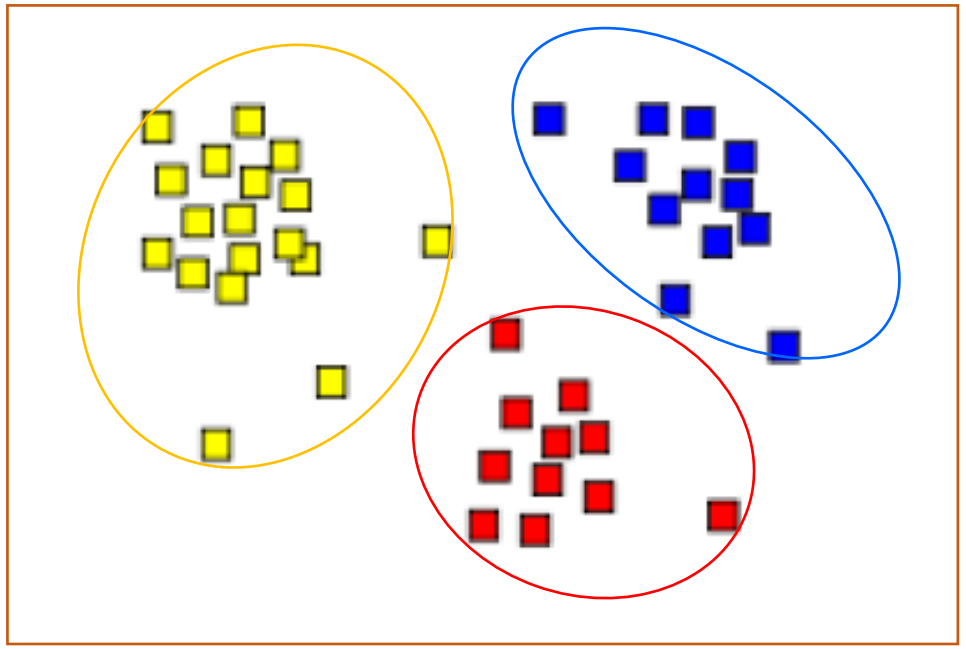


Clustering in Machine Learning

What is clustering?

- The organization of unlabeled data into similarity groups called clusters.
- Data items within a cluster are "*similar*"
- Data items between clusters are "*dissimilar*"

- Clustering is a unsupervised learning method
- Used for statistical data analysis in many fields.



Types of Clustering

Clustering can be divided into two subgroups

- **Hard Clustering:**
  - In hard clustering, each data point may belongs to a cluster completely or not.
  - For example, each customer is put into one group out of the 10 groups.

- **Soft Clustering**:
  - In soft clustering, instead of putting each data point into a separate cluster, a probability or likelihood of that data point to be in those clusters is assigned.
  - For example, each costumer is assigned a probability to be in either of 10 clusters of the retail store.

Types of clustering algorithms

- In fact, there are more than 100 clustering algorithms available. Here we discuss only few of the algorithms are used popularly used:
- **Connectivity models:**
  - These models are based on the notion that the data points closer in data space exhibit more similarity to each other than the data points lying farther away.
  - Two approached in these models
    - In the first approach, they start with classifying all data points into separate clusters & then aggregating them as the distance decreases.
    - In the second approach, all data points are classified as a single cluster and then partitioned as the distance increases.
    - These models are very easy to interpret but lacks scalability for handling big datasets. Examples of these models are hierarchical clustering algorithm and its variants.

- **Centroid models:**
  - These are iterative clustering algorithms in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters.
  - K-Means clustering algorithm is a popular algorithm that falls into this category.
  - In these models, the no. of clusters required at the end have to be mentioned beforehand, which makes it important to have prior knowledge of the dataset.
  - These models run iteratively to find the local optima.

- **Distribution models:**
  - These clustering models are based on the notion of how probable is it that all data points in the cluster belong to the same distribution (For example: Normal, Gaussian).
  - These models often suffer from overfitting.
  - A popular example of these models is Expectation-maximization algorithm which uses multivariate normal distributions.

- **Density Models:**
  - These methods consider the clusters as the dense region having some similarity and different from the lower dense region of the space.
  - These methods have good accuracy and ability to merge two clusters.
  - Example *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)*, *OPTICS (Ordering Points to Identify Clustering Structure)* etc.

Applications of Clustering in Different Fields

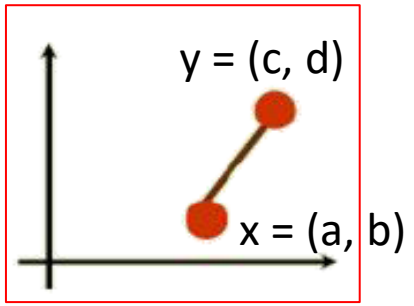
- **Marketing:** It can be used to characterize & discover customer segments for marketing purposes.
- **Biology:** It can be used for classification among different species of plants and animals.
- **Libraries:** It is used in clustering different books on the basis of topics and information.
- **Insurance:** It is used to acknowledge the customers, their policies and identifying the frauds.
- **City Planning:** It is used to make groups of houses and to study their values based on their geographical locations and other factors present.
- **Earthquake studies:** By learning the earthquake affected areas we can determine the dangerous zones.

Dissimilarity (Distance) Measures

Euclidean: Take the square root of the sum of the squares of the differences of the coordinates

For example, if x=(a, b) and y=(c, d), the Euclidean distance between x and y is

$$d = \sqrt{(a-c)^2 + (b-d)^2}$$



In general, the Euclidean distance d (x_i, y_i) is

$$d\,(x_i, y_i) = \sqrt{\sum_{k=1}^{d} (x_i - y_i)^2}$$

Clustering Techniques

- Hierarchical algorithms find successive clusters using previously established clusters.
- Types:
  - Agglomerative (“bottom-up”) or
  - Divisive (“top-down”)

  - Agglomerative algorithm begins with each element as a separate cluster and merge them into successively larger clusters

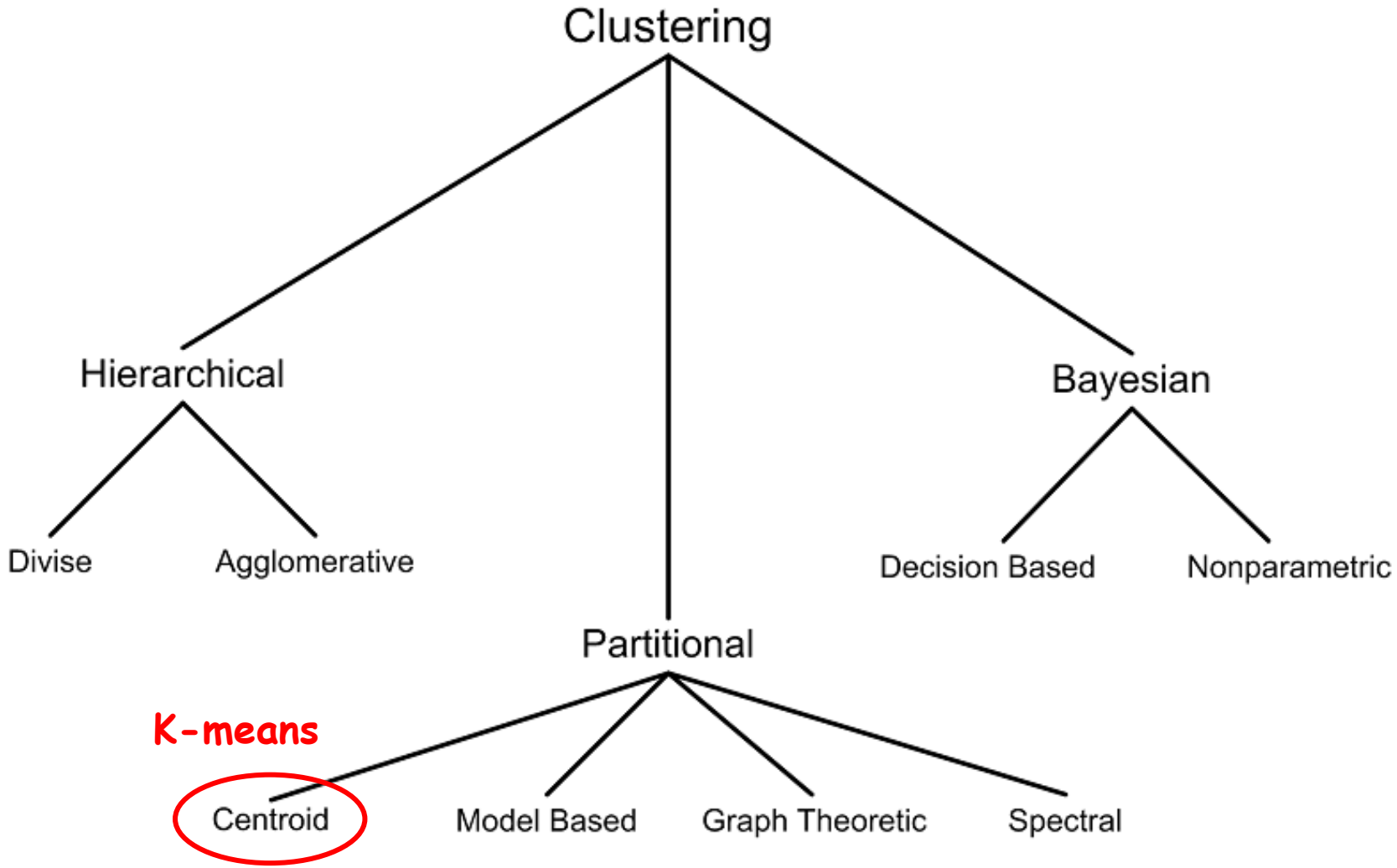
  - Divisive algorithms begin with the whole set proceed to divide it into successively smaller clusters.

Clustering Techniques

- Partitional algorithm typically determine all clusters at once, but can also be used as divisive algorithms in the hierarchical clustering.

- Bayesian algorithm try to generate a posteriori distribution over the collection of partitions of the data.

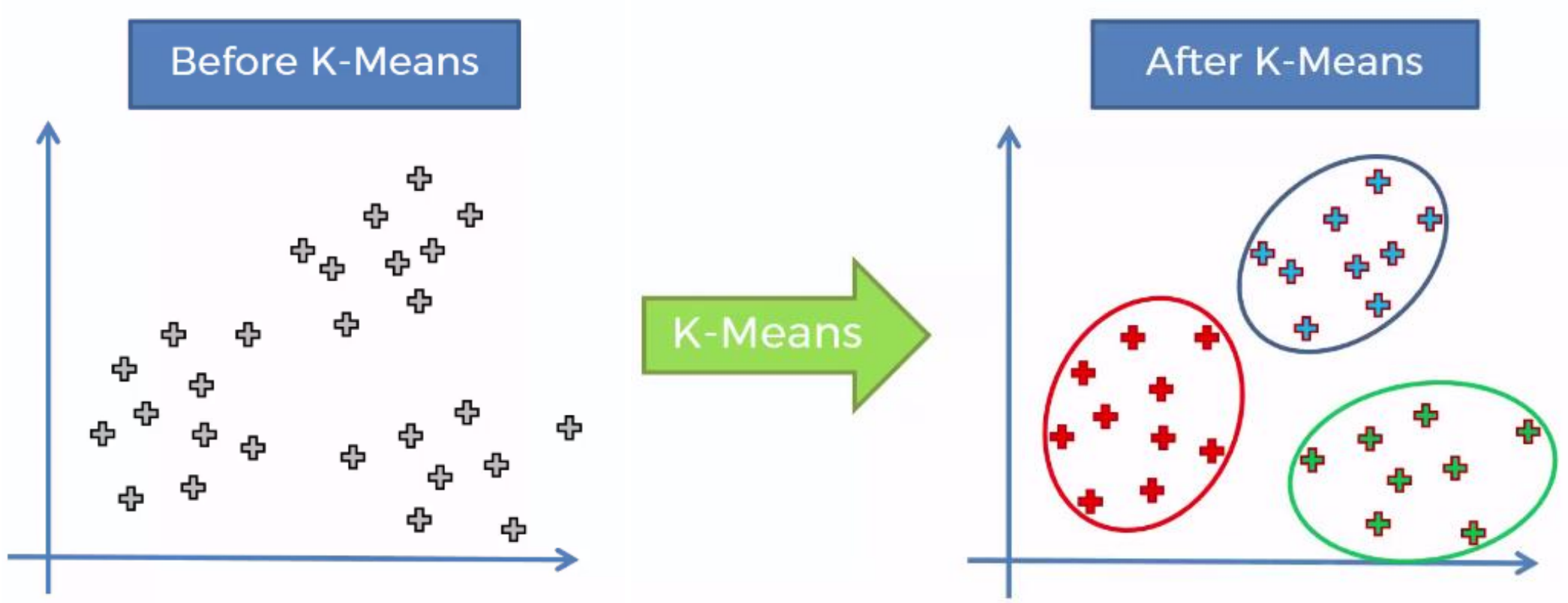
Clustering Techniques



K-Means Clustering

The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid.

K-Means Clustering



K-Means Algorithm

Step-1: Choose the number K of clusters

Step-2: Select at random k points, the centroids (not necessarily from your dataset)

Step-3: Assign each data point to the closest centroid → that forms k clusters

Step-4: Compute and place the new centroid of each cluster

Step-5: Reassign each data points to the new closest centroid.
If any reassignment took place, go to Step-4, otherwise go to Stop.

Stop: your Model is ready

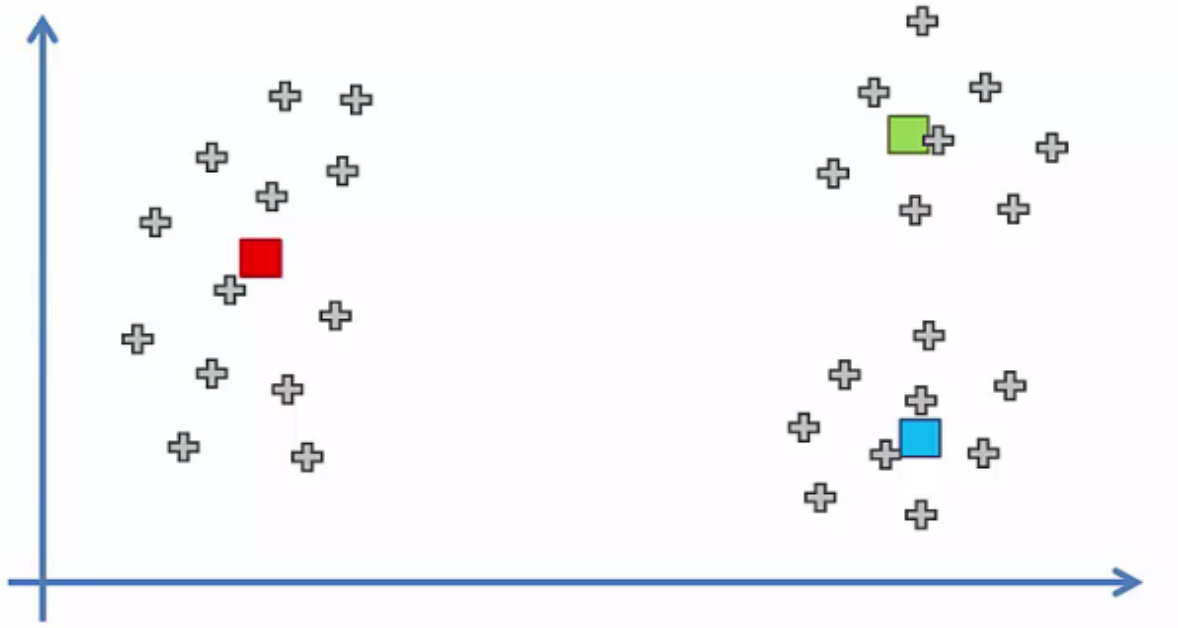
K-Means Clustering: Example-1

Objects	Attribute 1 (X):weight index	Attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

- These objects belong to two groups of medicine (cluster 1 and cluster 2). K=2
- Determine which medicines belong to cluster 1 and which medicines belong to the other cluster 2.

Random Initialization of Centroids

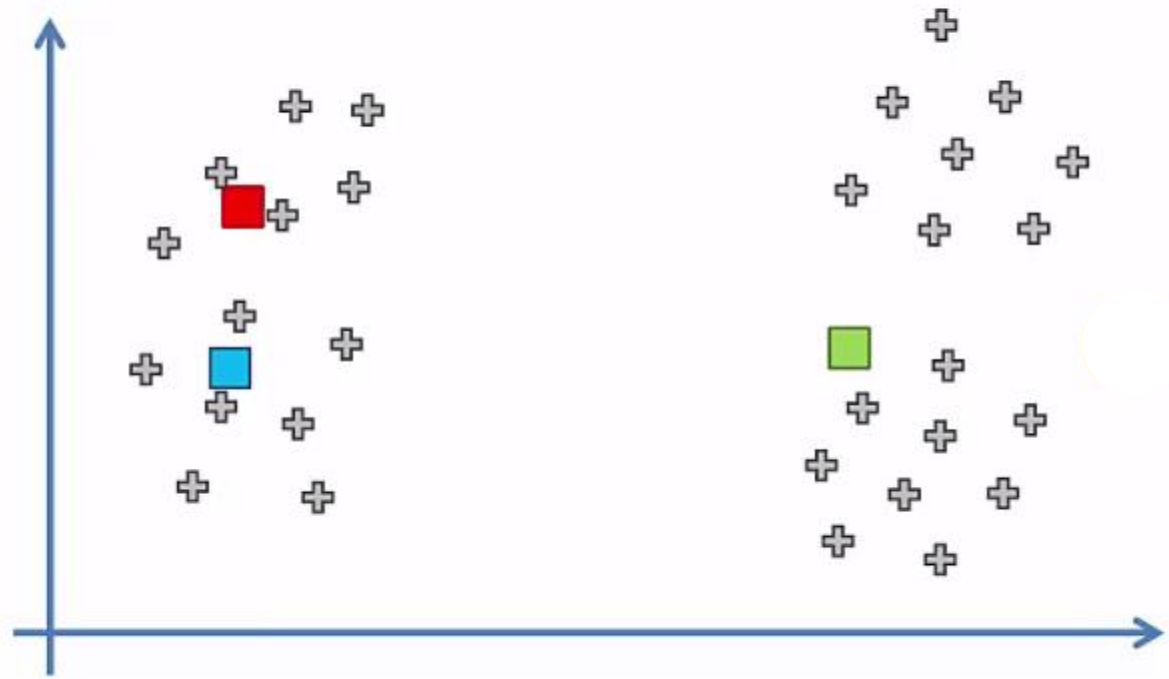
If I chose correct random initialization of centroids …



…this correct random initialisation would lead us to…

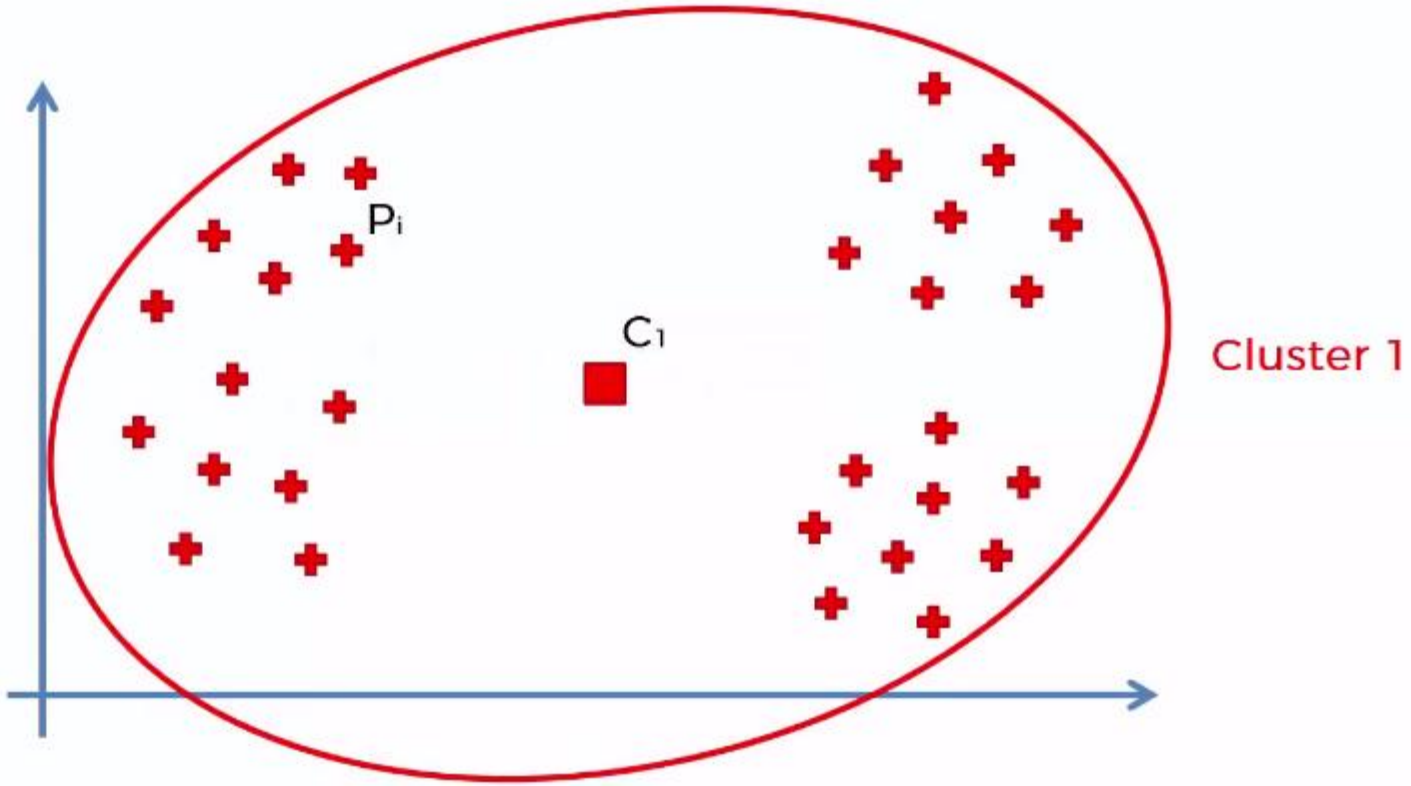
What will happen if we choose a bad random initialization?

STEP 2: Select at random K points, the centroids (not necessarily from your dataset)

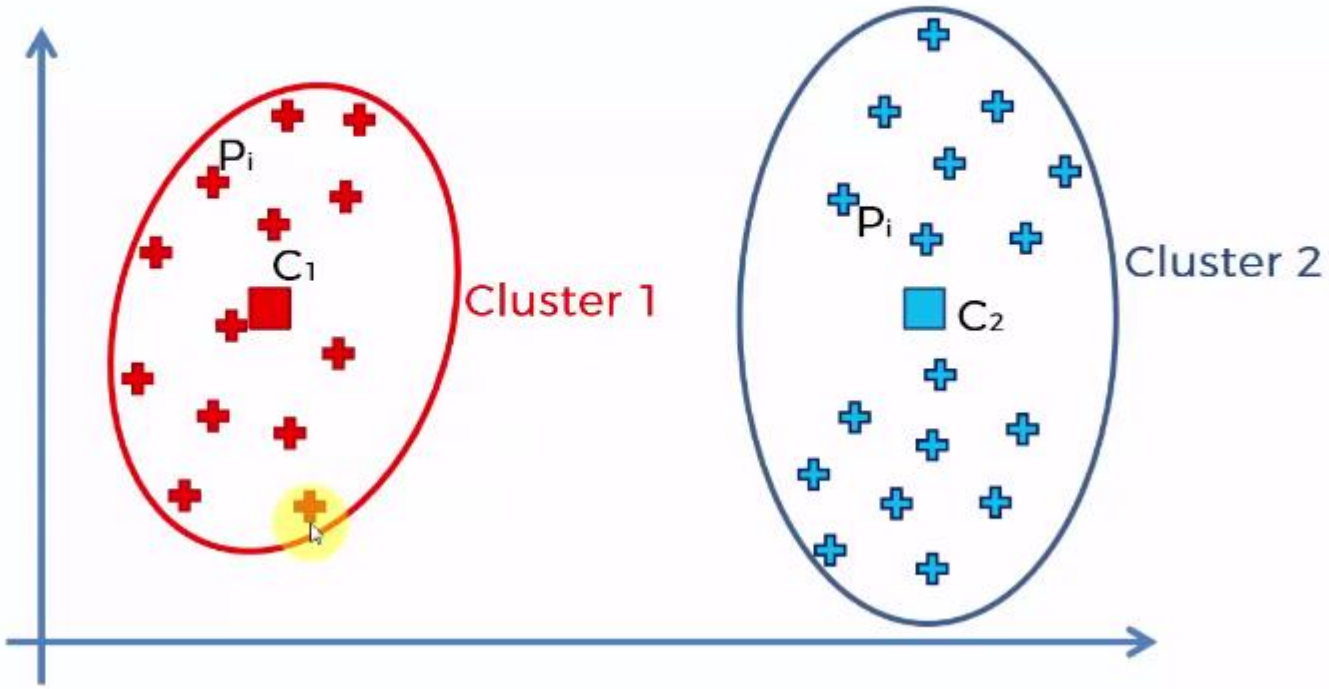


Choosing the Right Number of Clusters

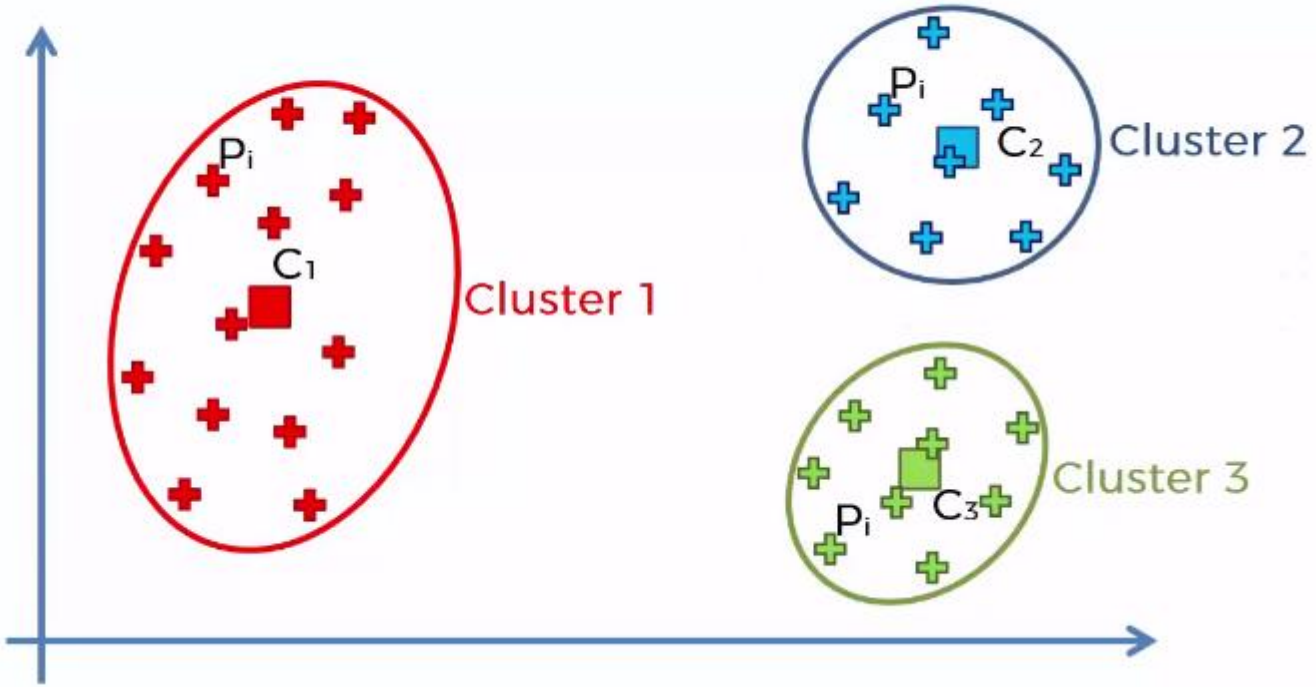
→ Within Cluster Sum of Square (WCSS)



$$\mathbf{WCSS_1} = \sum_{\mathbf{p_i \text{ in cluster } 1}} \mathbf{distance}\,(\mathbf{P_i}, \mathbf{C1})^2$$

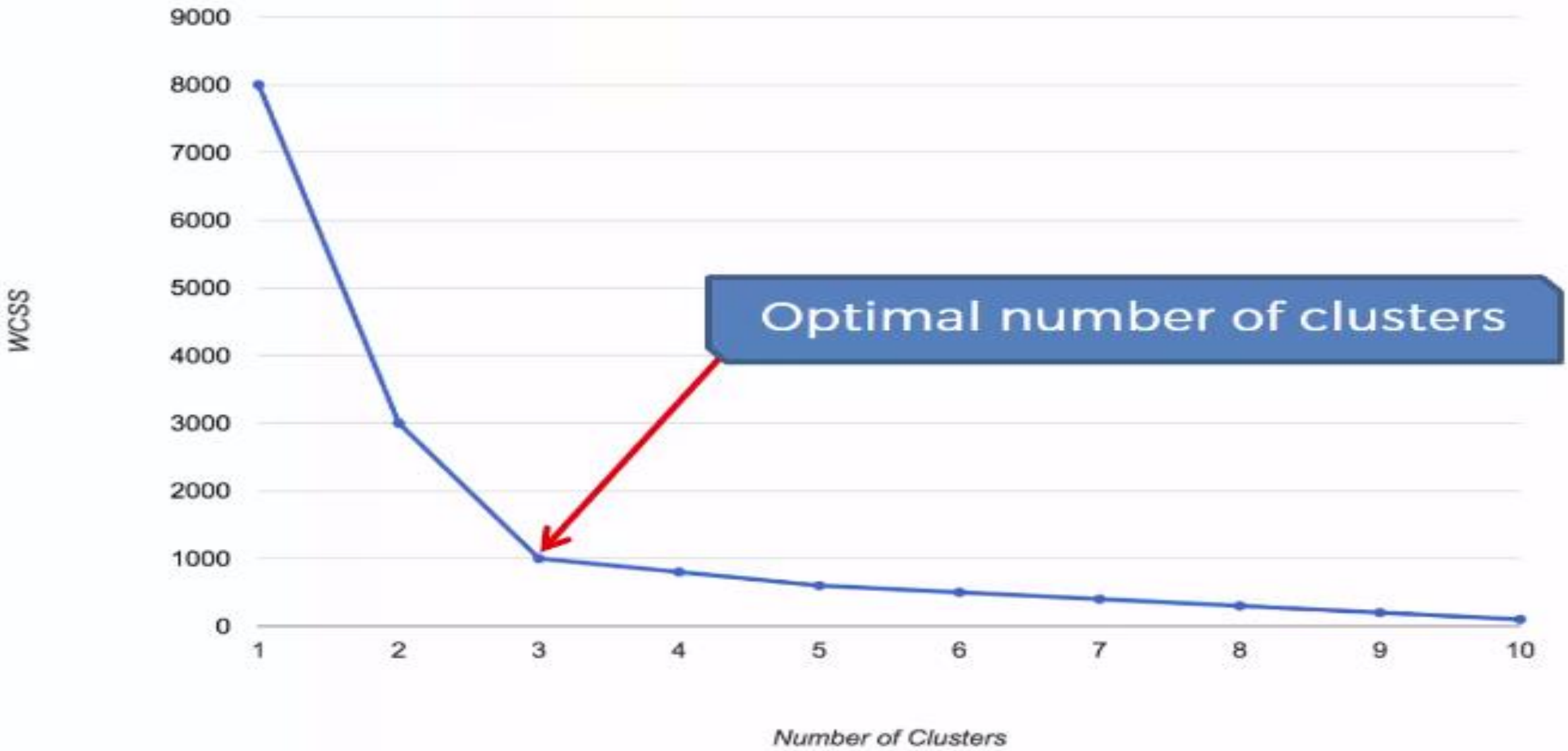


$$\text{WCSS} = \sum_{p_i \text{ in cluster 1}} \text{distance}\,(P_i, C1)^2 + \sum_{p_i \text{ in cluster 2}} \text{distance}\,(P_i, C_2)^2$$



$$\text{WCSS} = \sum_{p_i \text{ in cluster 1}} \text{distance}\,(P_i, C1)^2 + \sum_{p_i \text{ in cluster 2}} \text{distance}\,(P_i, C_2)^2 + \sum_{p_i \text{ in cluster 3}} \text{distance}\,(P_i, C_3)^2$$

The Elbow Method



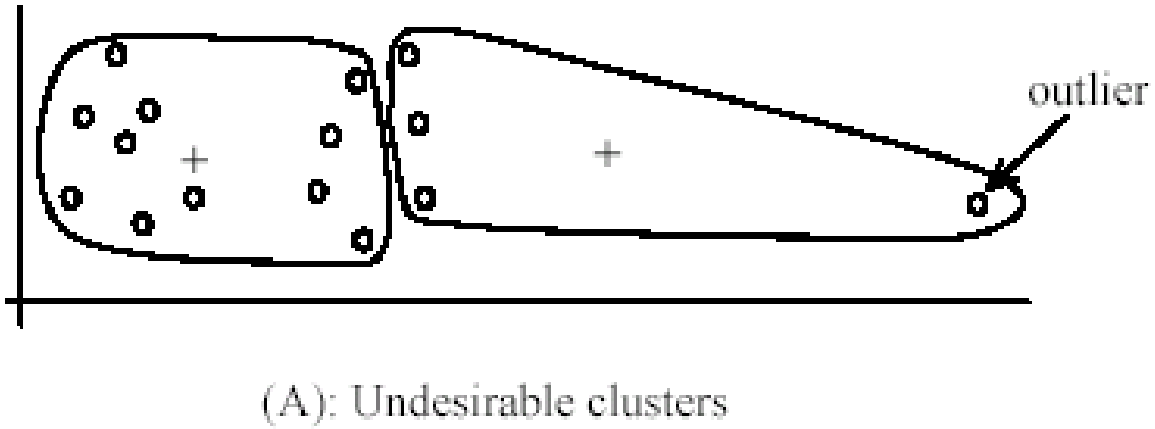
Strengths of K-means

- **Simple**: easy to understand and to implement
- **Efficient:** Time complexity: $O(tkn)$
  - n-is the number of data points
  - *k-is the number of clusters*
  - *t- is the number of iterations*
  - Since both k and t are small. k-means is considered a linear algorithm.
- K-means is the most popular clustering algorithm.

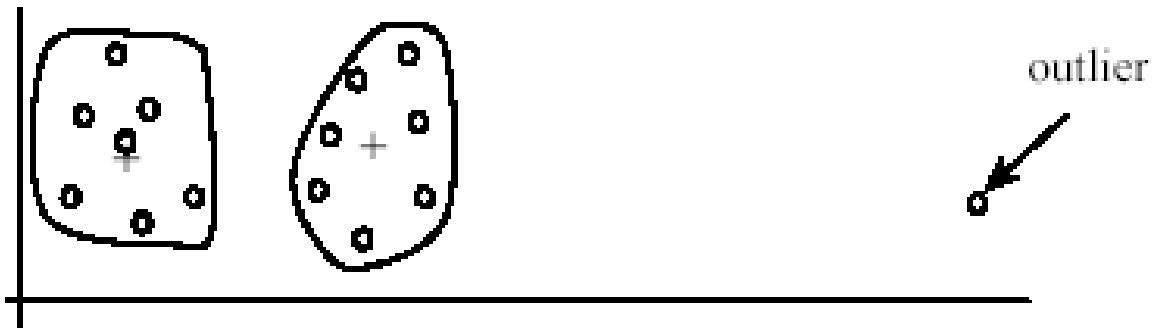
Weaknesses of K-means

- The user needs to specify k.
- The algorithm is sensitive to **outliers**
  - Outliers are data points that are **very far away from other data points**.
  - Outliers could be **errors** in the data recording or some **special data points** with very different values.

Outliers



(A): Undesirable clusters

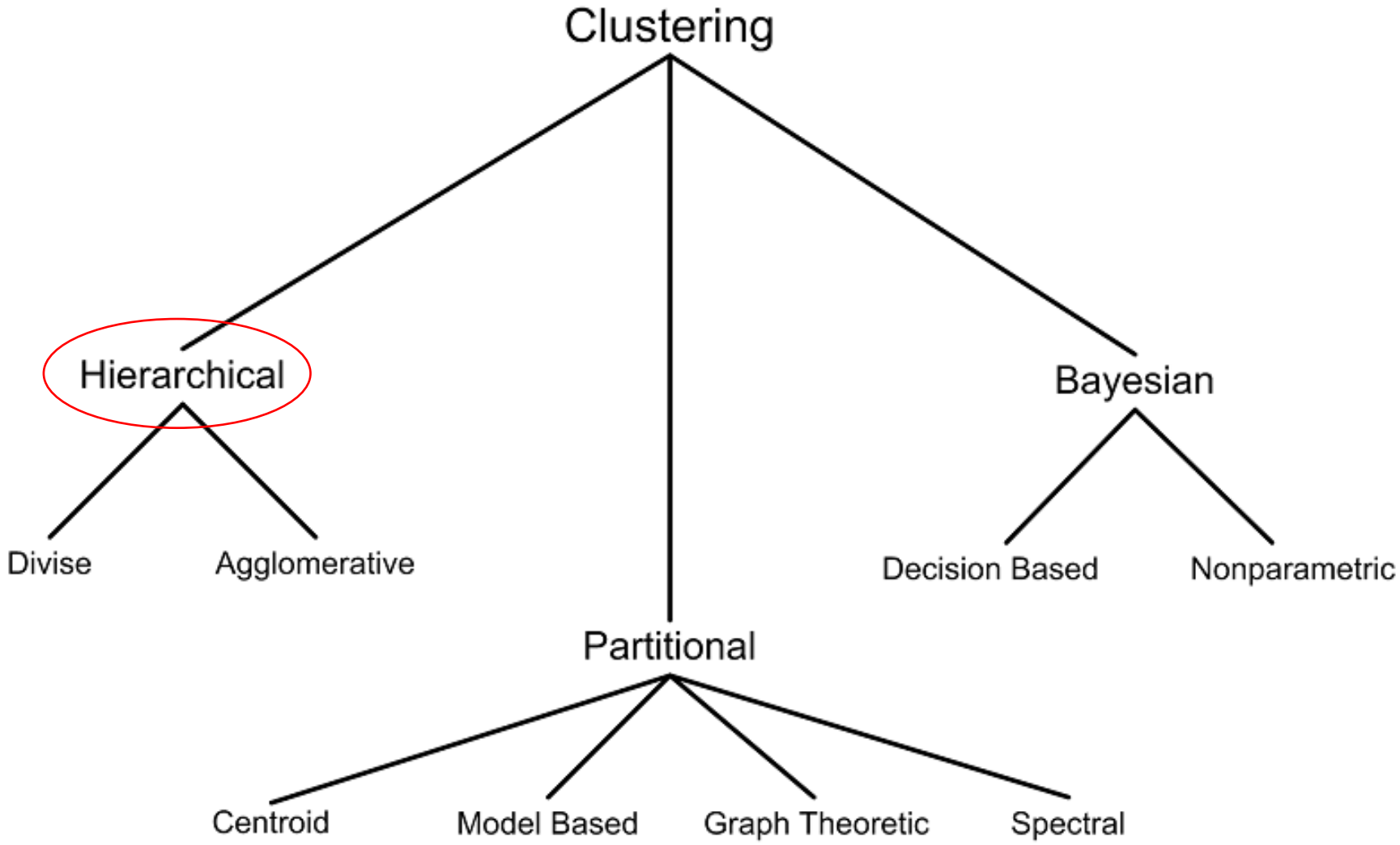


(B): Ideal clusters

Handling with outliers

- Remove some data points that are much further away from the centroids
- To be safe, monitor these possible outliers over a few iterations and then decide to remove them.

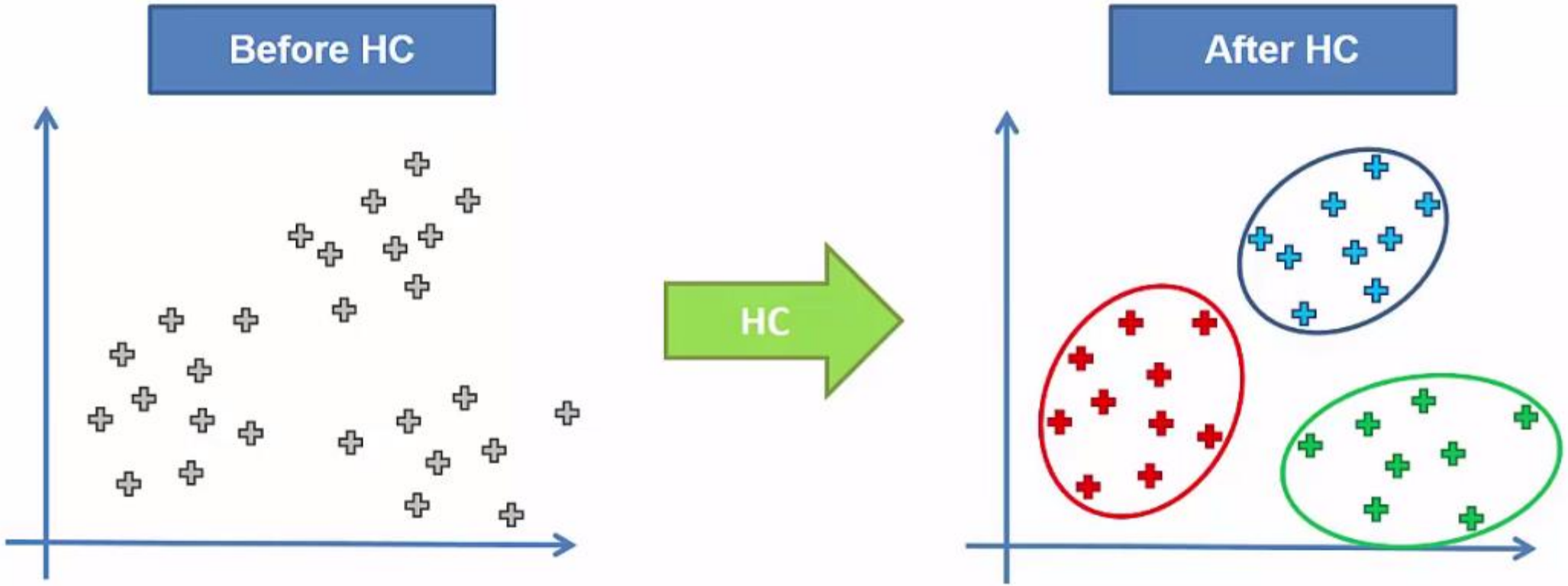
Hierarchical Clustering



Hierarchical Clustering

In Machine Learning

What HC does for you



Same as K-Means but different process

Hierarchical Clustering

- Two types
  - Agglomerative (bottom-up)
  - Divisive (top-down)

- Agglomerative Algorithm
  - Begin with each data element as a separate cluster and merge them into successively larger cluster

- Divisive Algorithm
  - Begins with the whole set and proceed to divide it into successive smaller clusters

Agglomerative HC

STEP 1: Make each data point a single-point cluster ➡ That forms N clusters

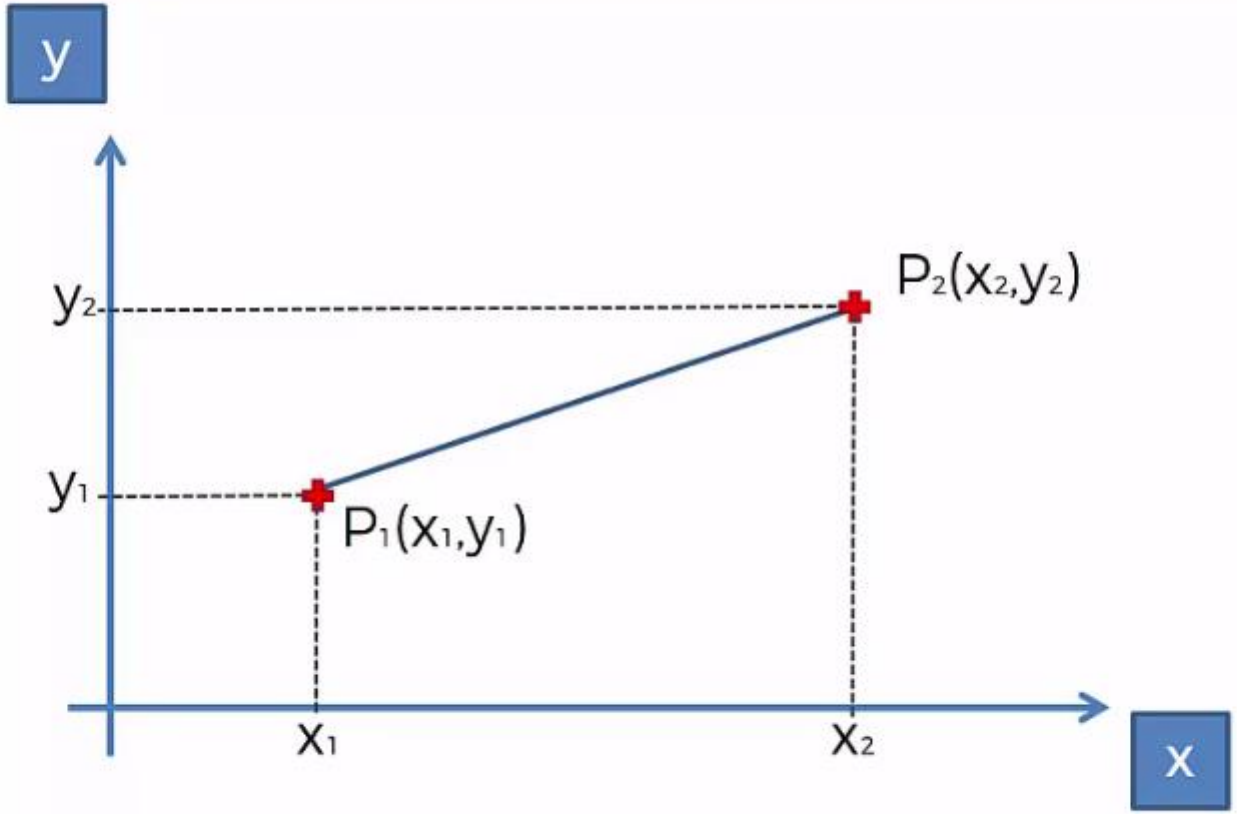
STEP 2: Take the two closest data points and make them one cluster ➡ That forms N-1 clusters

STEP 3: Take the two closest clusters and make them one cluster ➡ That forms N - 2 clusters

STEP 4: Repeat STEP 3 until there is only one cluster

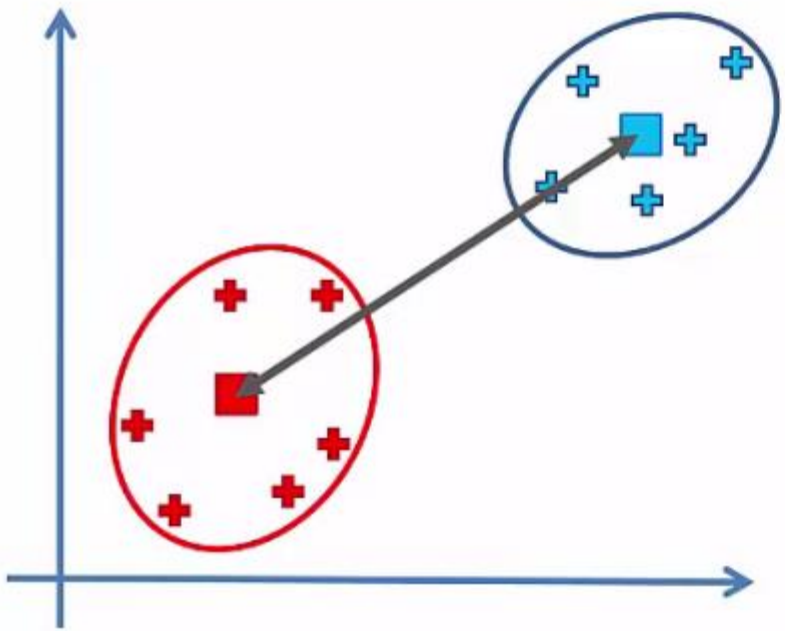
FIN

Euclidean Distance



Euclidean Distance between P_1 and $P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Distance Between Clusters

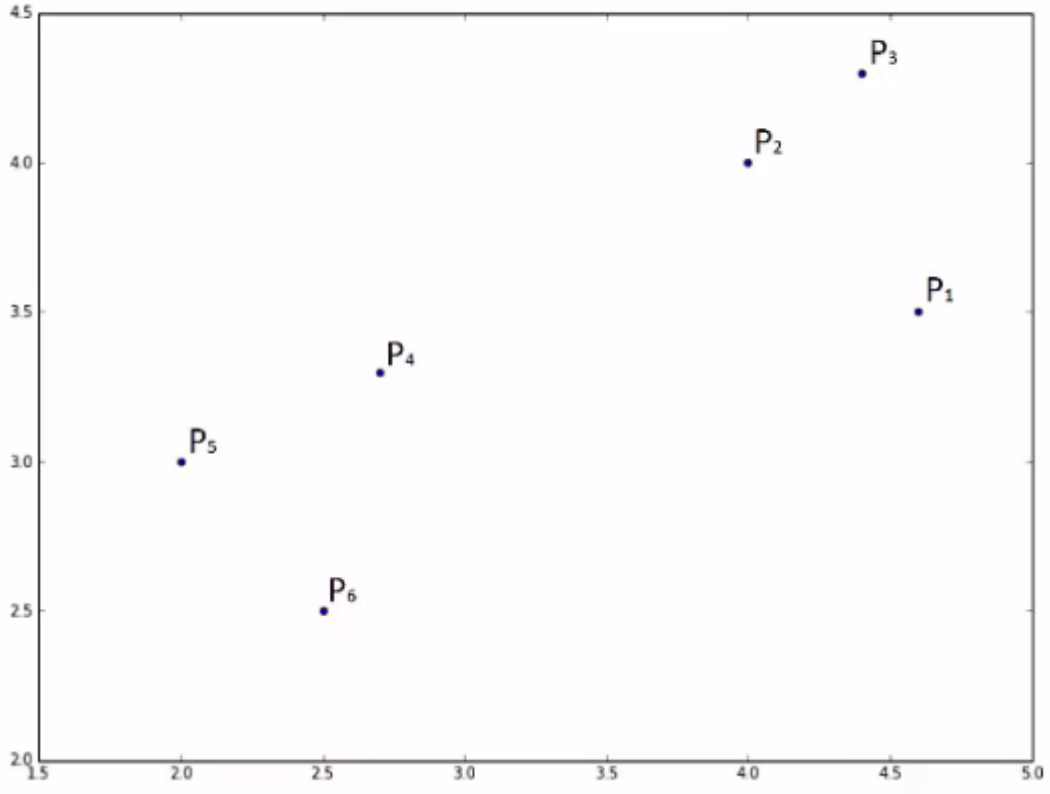


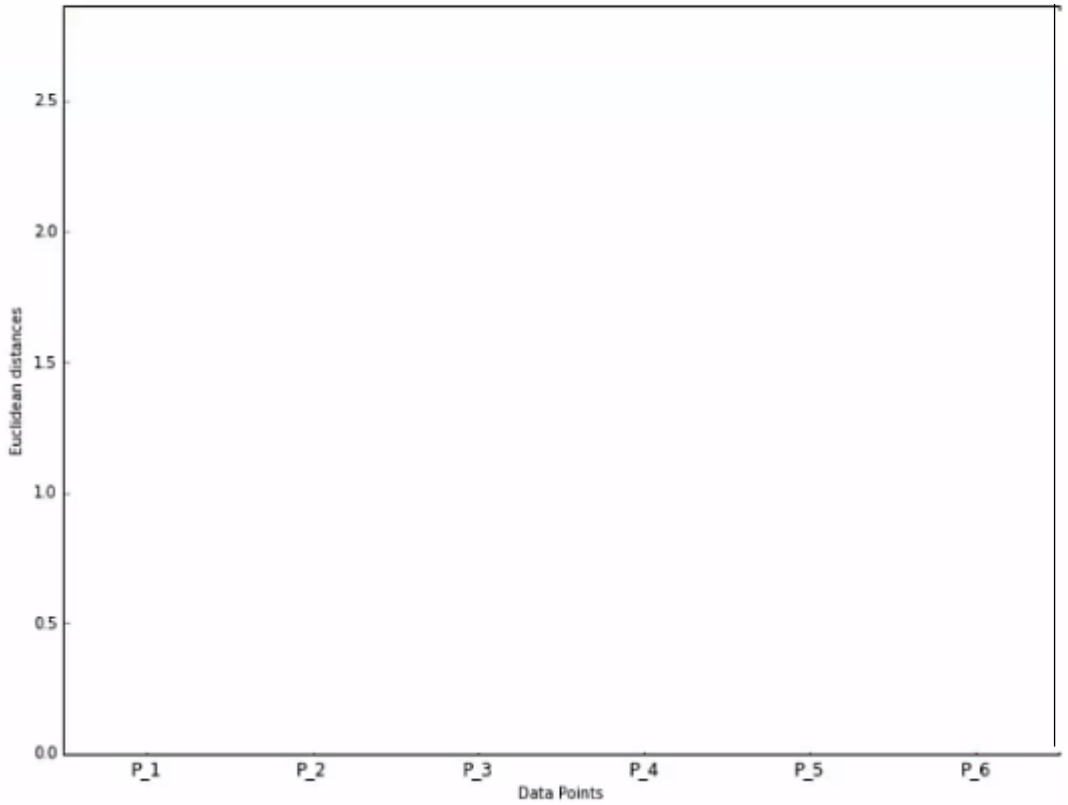
Distance Between Two Clusters:

- Option 1: Closest Points
- Option 2: Furthest Points
- Option 3: Average Distance
- Option 4: Distance Between Centroids

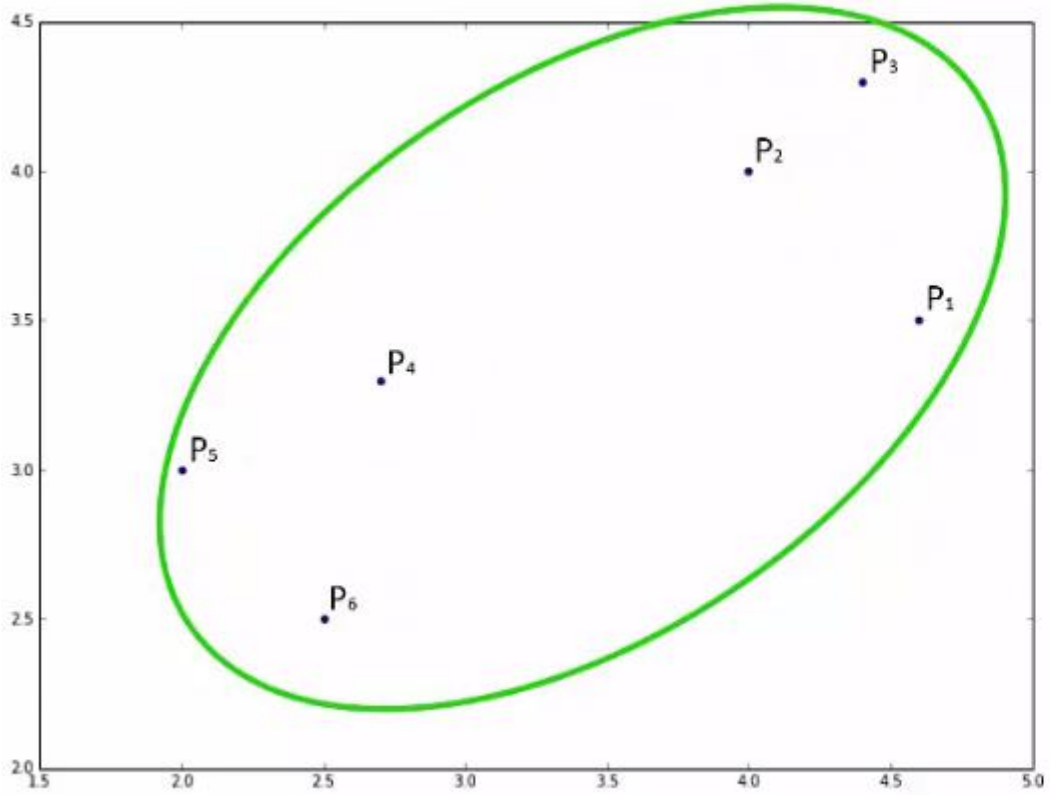
How do Dendrograms Works?

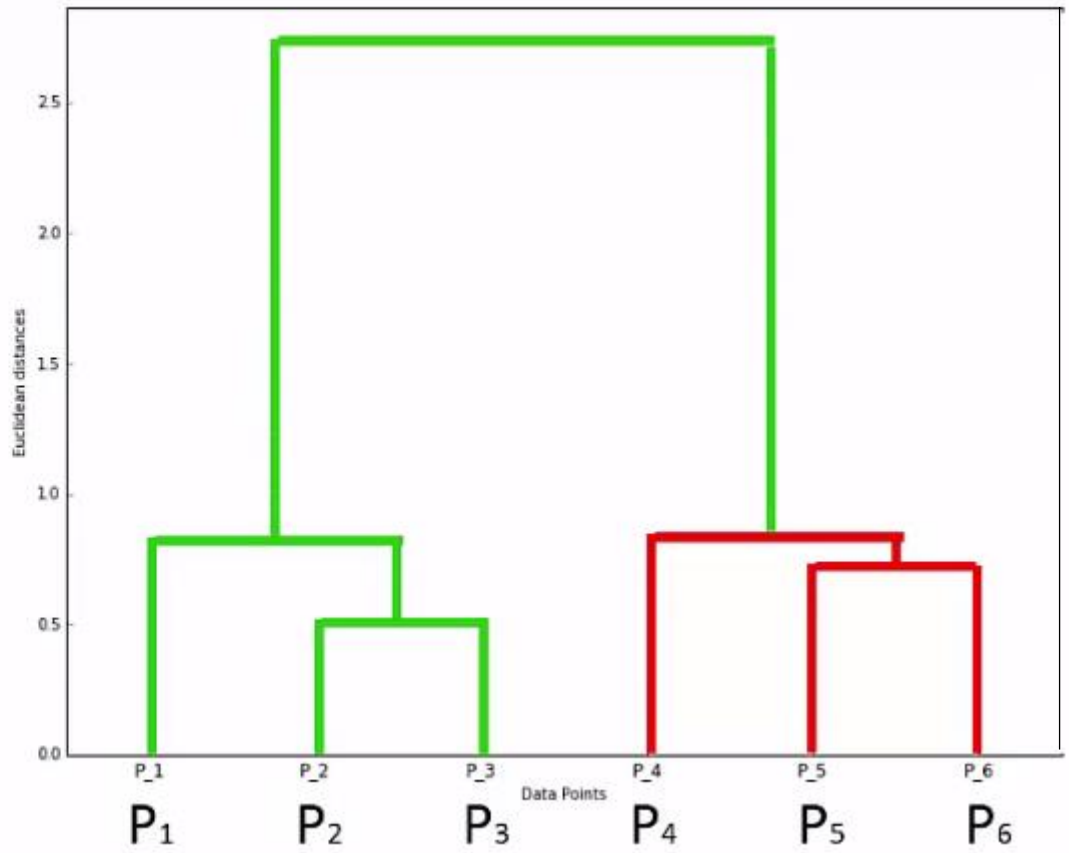
How do Dendrograms Works



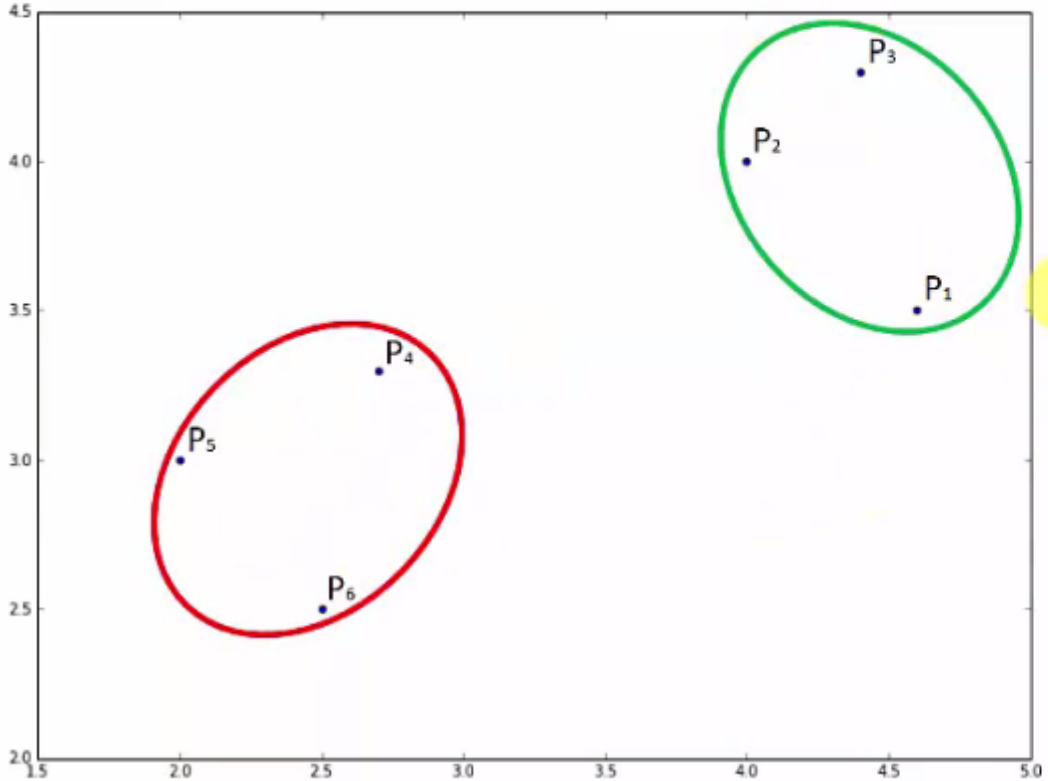


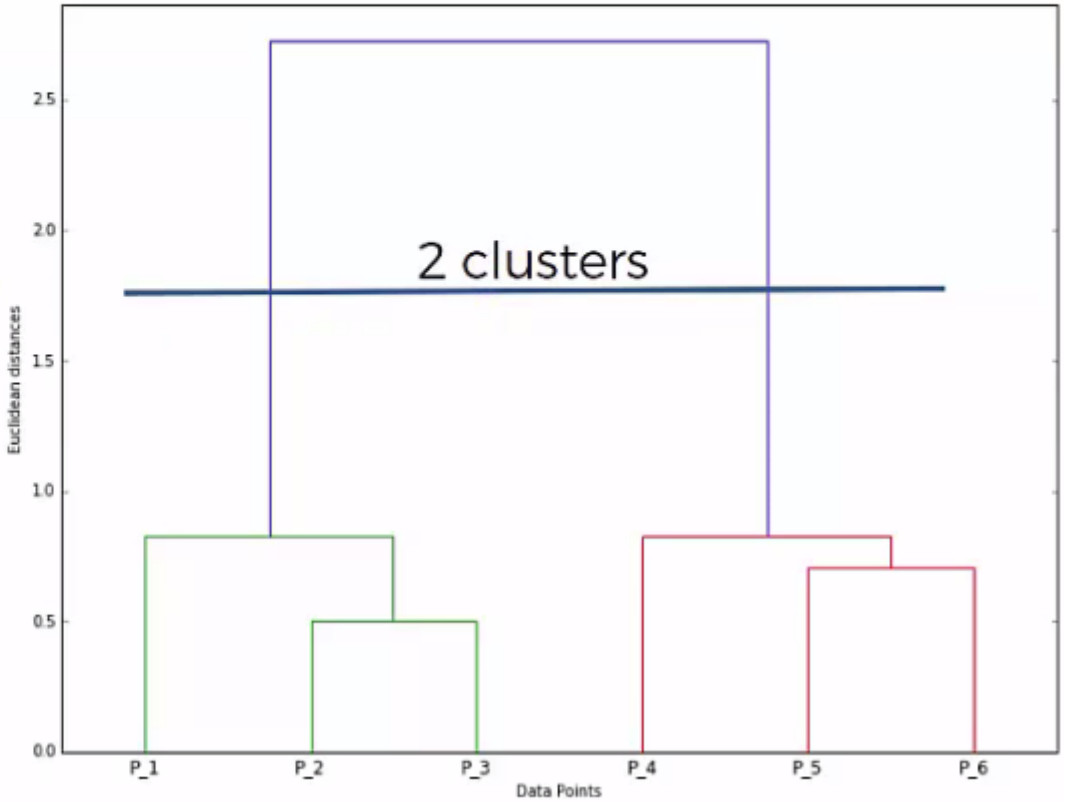
How do Dendrograms Works



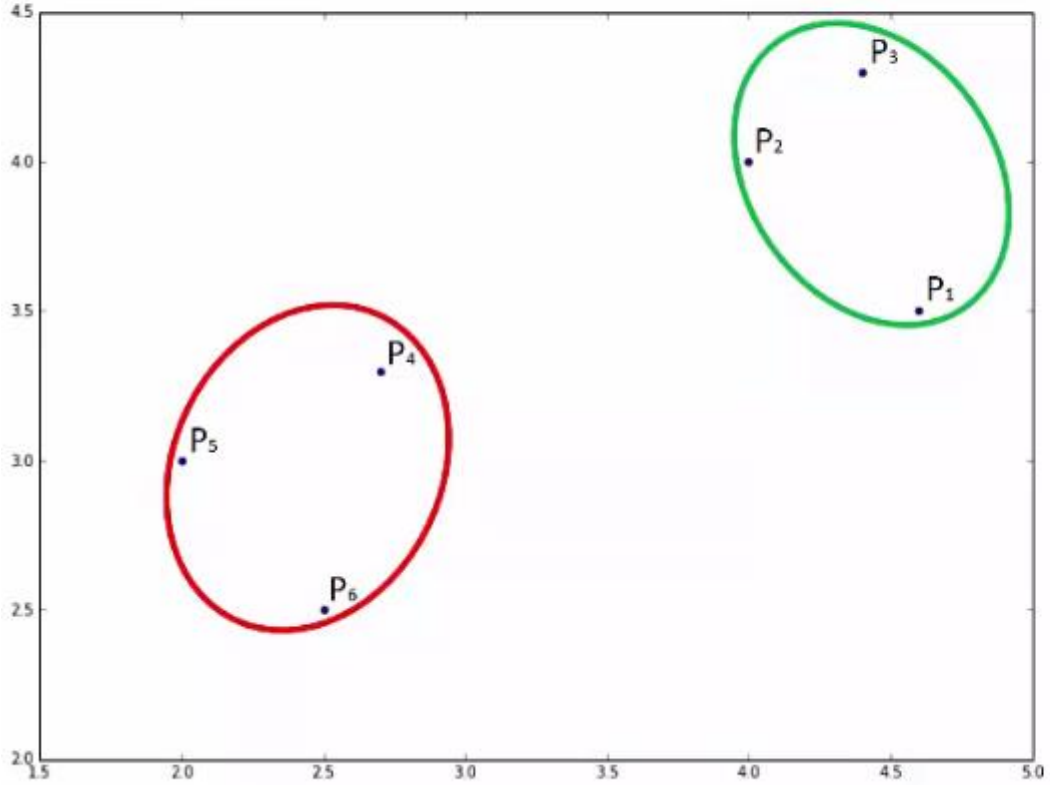


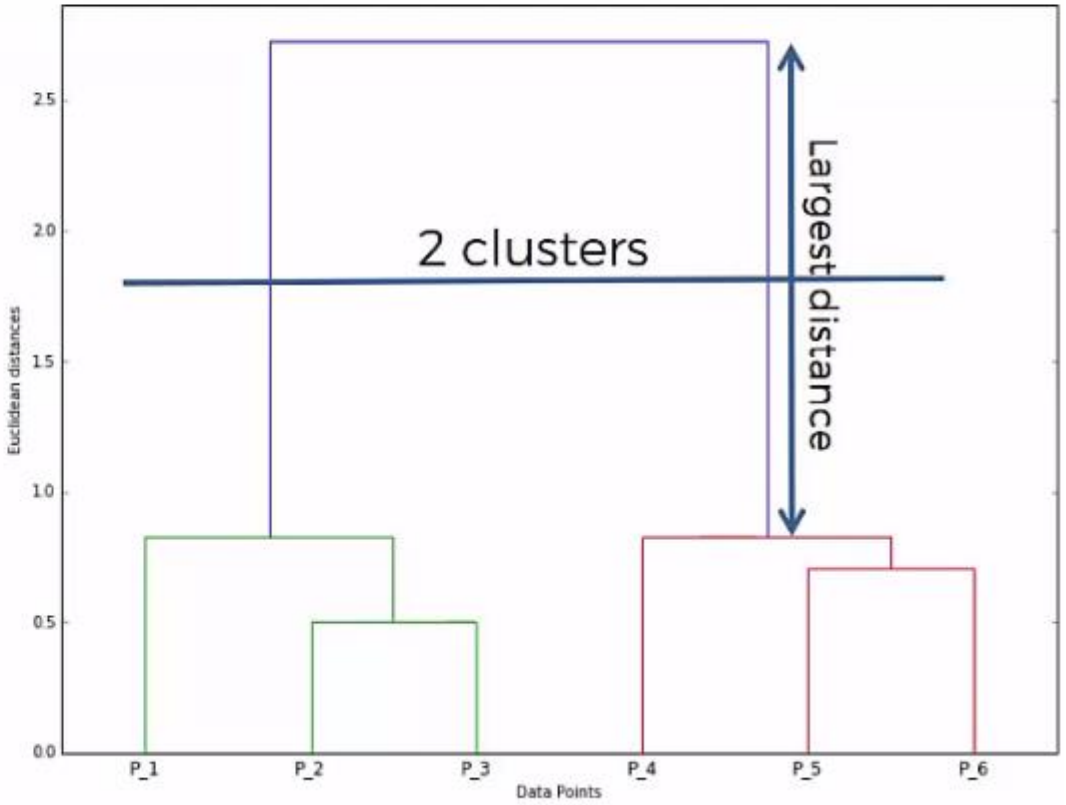
Dendrograms – Two Clusters





Dendrograms – Optimal # of Clusters





Clustering

Expectation-Maximization Algorithm

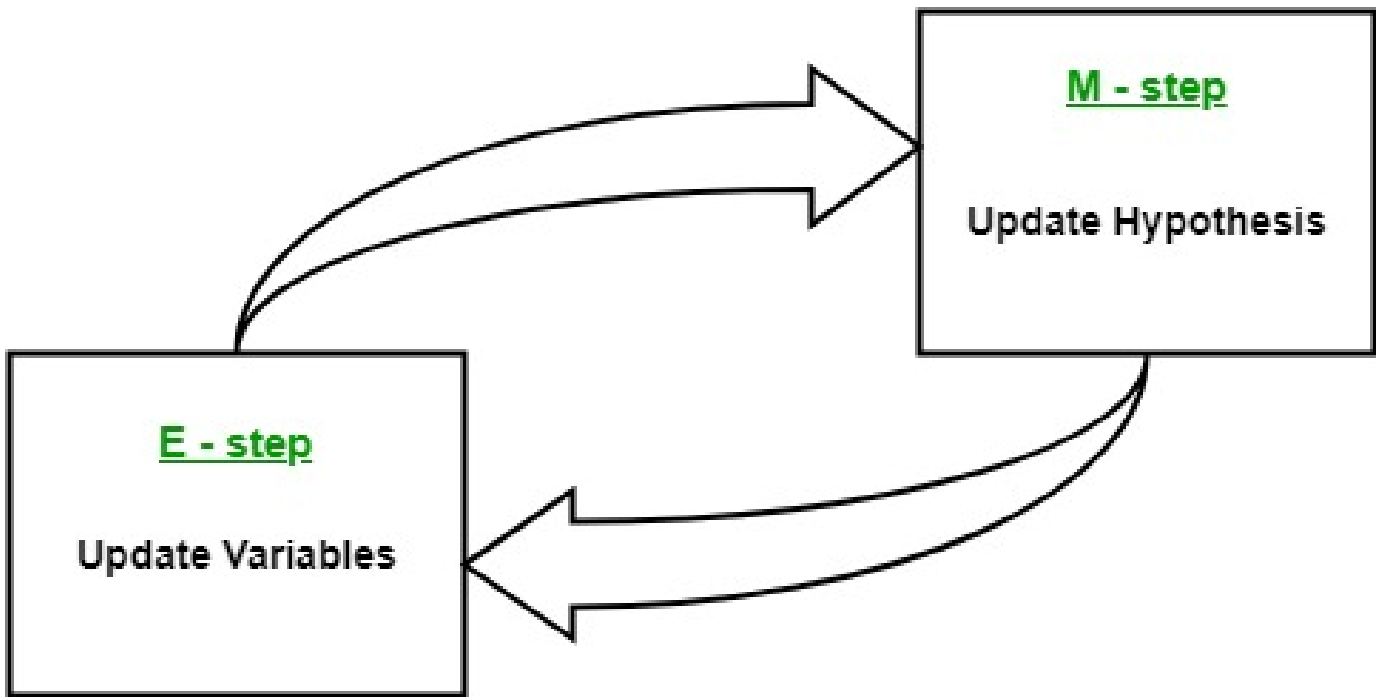
Expectation-Maximization Algorithm

This algorithm is base for many unsupervised clustering algorithms.

It was proposed in 1977 by Dempster, Nan Laird, and Donald Rubin.

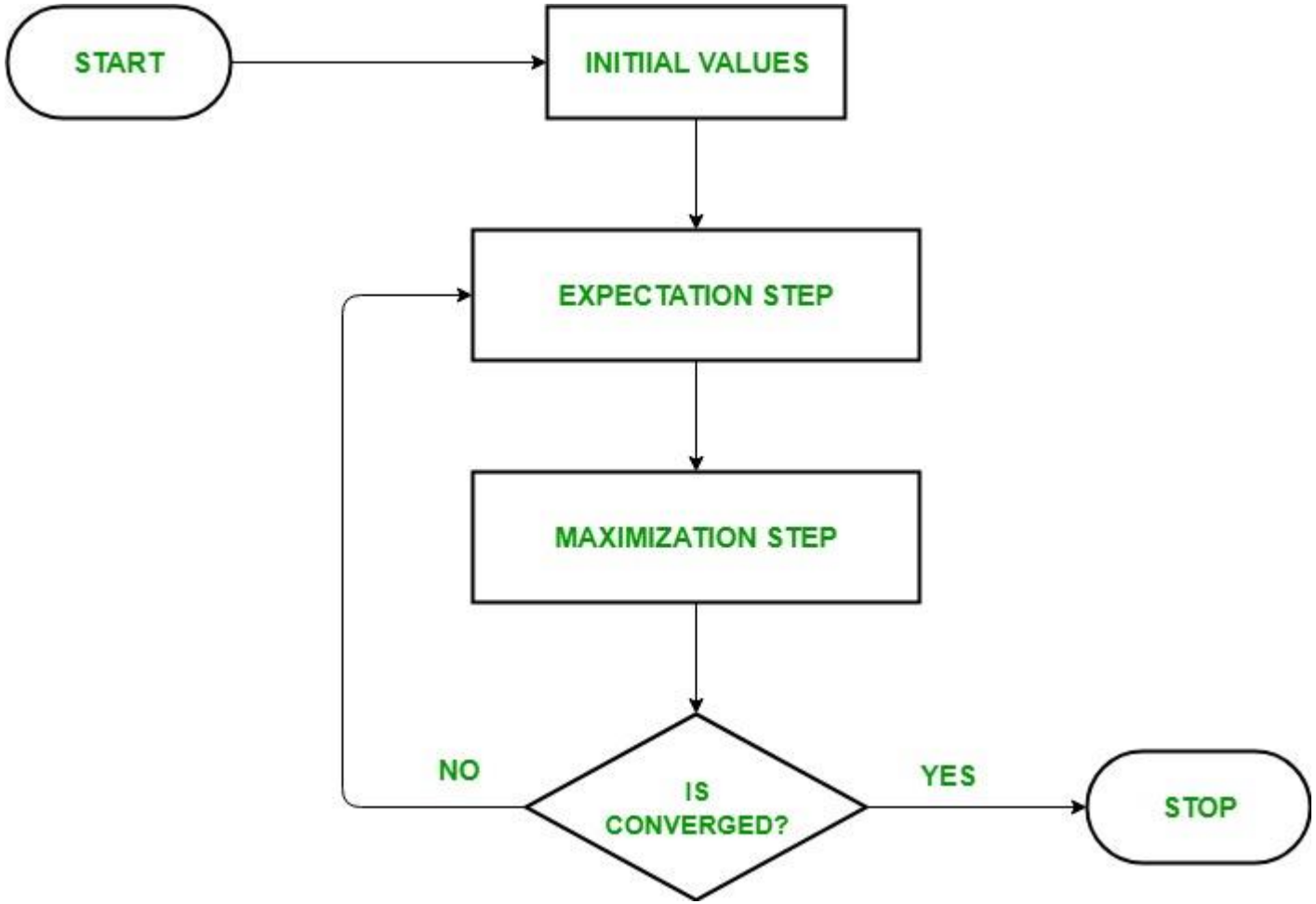
It is used to find the local maximum likelihood parameters in the cases where latent variables are involved and the data is missing or incomplete.

1. Given a set of incomplete data, consider a set of initial parameters.
2. Expectation step (E – step):
    1. Using the observed available data of the dataset, estimate (guess) the values of the missing data.
    2. The algorithm computes the latent variables i.e. expectation of the log-likelihood using the current parameter estimates.
3. Maximization step (M – step):
    1. Complete data generated after the expectation (E) step is used in order to update the parameters.
    2. The algorithm determines the parameters that maximize the expected log-likelihood obtained in the E step, and corresponding model parameters are updated based on the estimated latent variables.
4. Repeat step 2 and step 3 until convergence.


M - step
Update Hypothesis
E - step
Update Variables

Expectation-Maximization (EM) Algorithm Works:

- The Expectation-Maximization algorithm use the available observed data of the dataset to estimate the missing data and then use that data to update the values of the parameters.



Example

Let C1 and C2 be two coins.

Θ_1 be probability of getting head with C1
Θ_2 be probability of getting head with C2

Find values of Θ_1 and Θ_2 by tossing C1 and C2 for multiple times.

$$\Theta_1 = \frac{no.of\ heads\ with\ C1}{Total\ no.of\ flips\ using\ C1}$$

$$\Theta_2 = \frac{no.of\ heads\ with\ C2}{Total\ no.of\ flips\ using\ C2}$$

Example

For Coin A:

$\sum L(H) = 21.3$

$\sum L(T) = 8.6$

$\Theta_1 = 21.3/(21.3+8.6)$

$= 0.71$

For Coin B:

$\sum L(H) = 11.7$

$\sum L(T) = 8.4$

$\Theta_2 = 11.7/(11.7+8.4)$

$= 0.58$

These values of Θ_1 and Θ_2 will be sent to next iteration.

After 10 iteration:

The process will be continued until you get stable value of Θ_1 and Θ_2.

$\theta_1 = 0.80$

$\theta_2 = 0.52$

Usage of EM algorithm

It can be used to fill the missing data in a sample.

It can be used for the purpose of estimating the parameters of Hidden Markov Model (HMM).

It can be used for discovering the values of latent variables.

Advantages of EM algorithm

It is always guaranteed that likelihood will increase with each iteration.

The E-step and M-step are often easy for many problems in terms of implementation.

Disadvantages of EM algorithm

- It has slow convergence.
- It makes convergence to the local optima only.
- It requires both the probabilities, forward and backward.

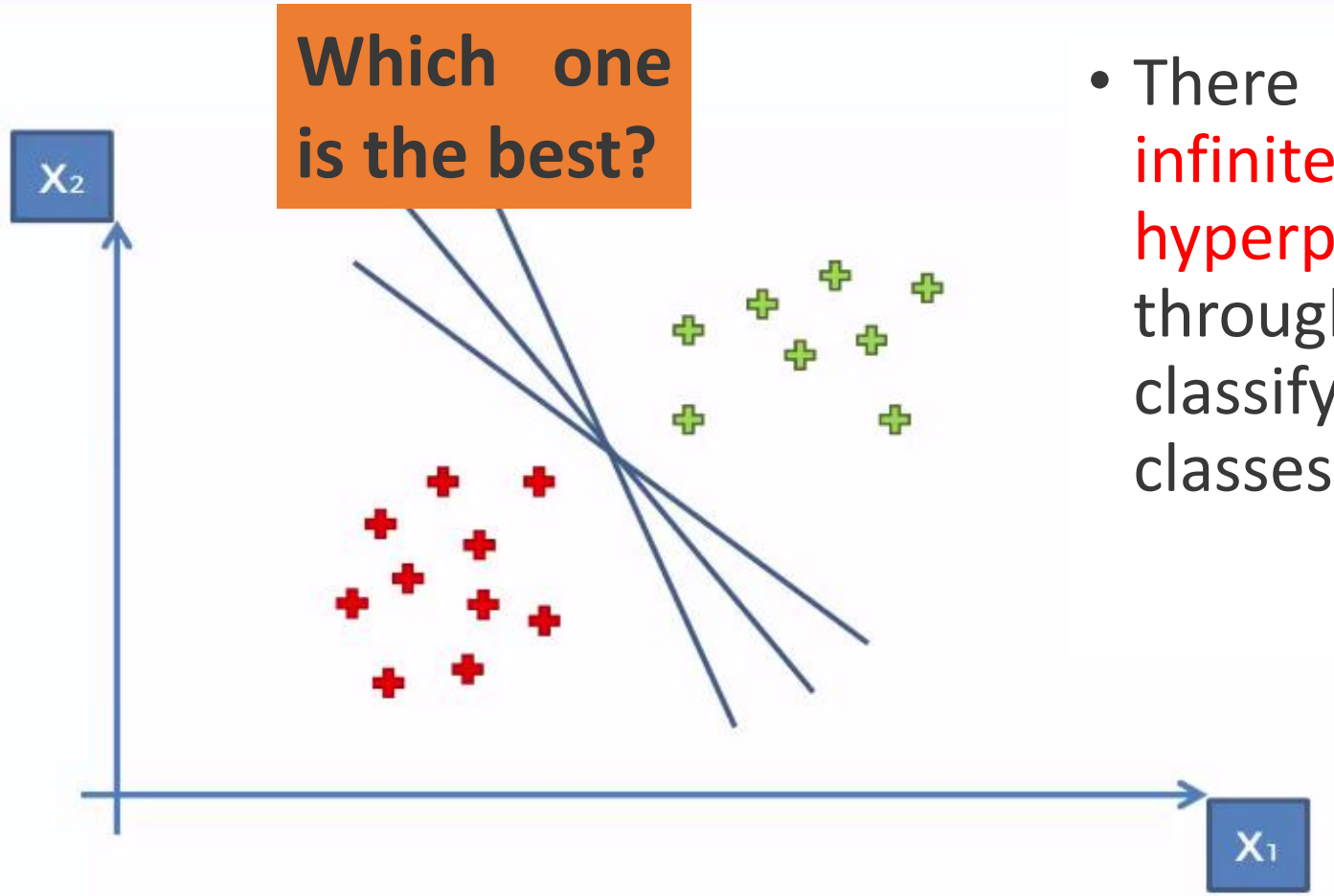
Support Vector Machine (SVM)

Machine Learning Approach

What is a Support Vector Machine(SVM)?

- It is a supervised machine learning problem where we try to find a hyperplane that best separates the two classes.

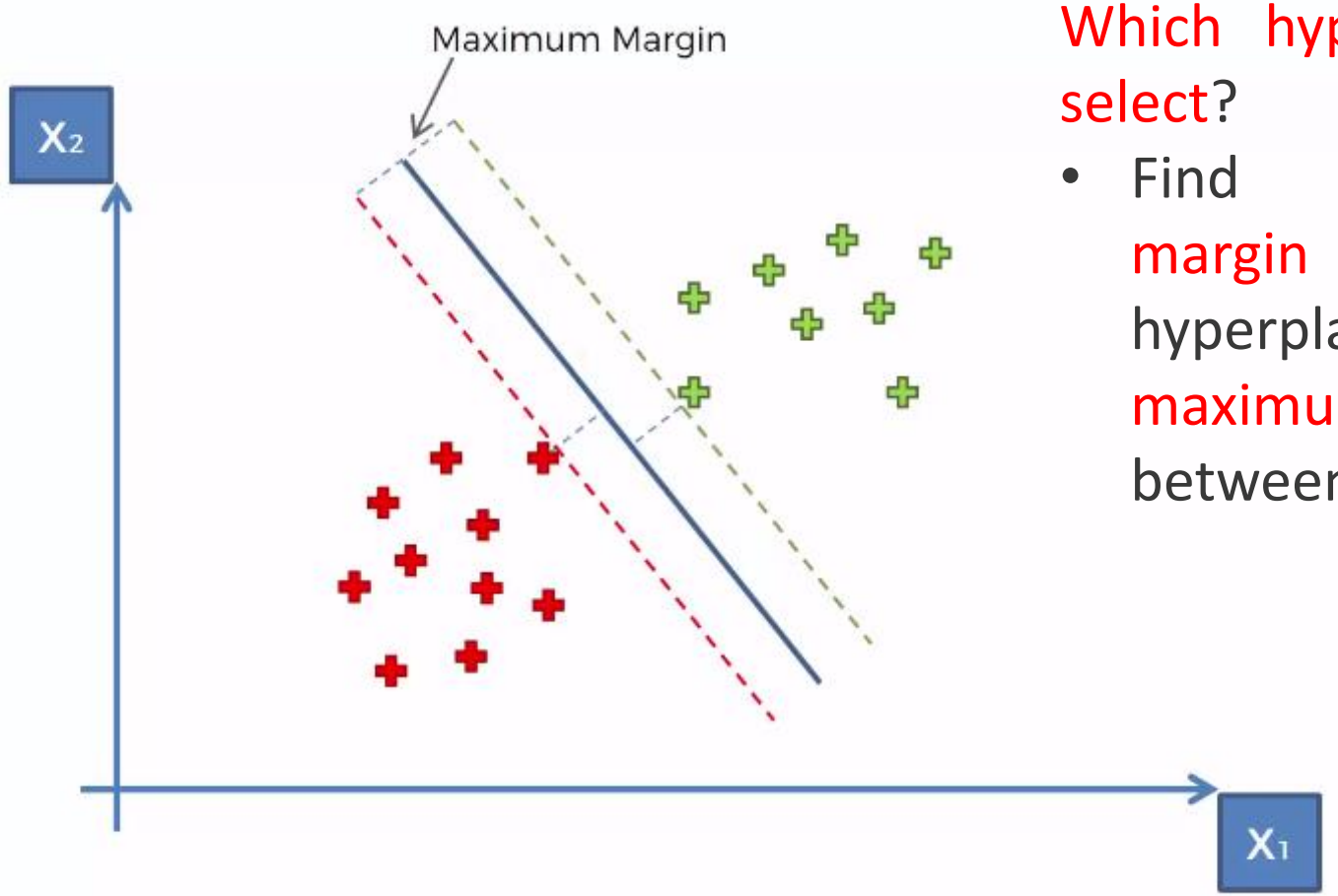
SVM



- There can be an infinite number of hyperplanes passing through a point and classifying the two classes perfectly.

Maximum Margin

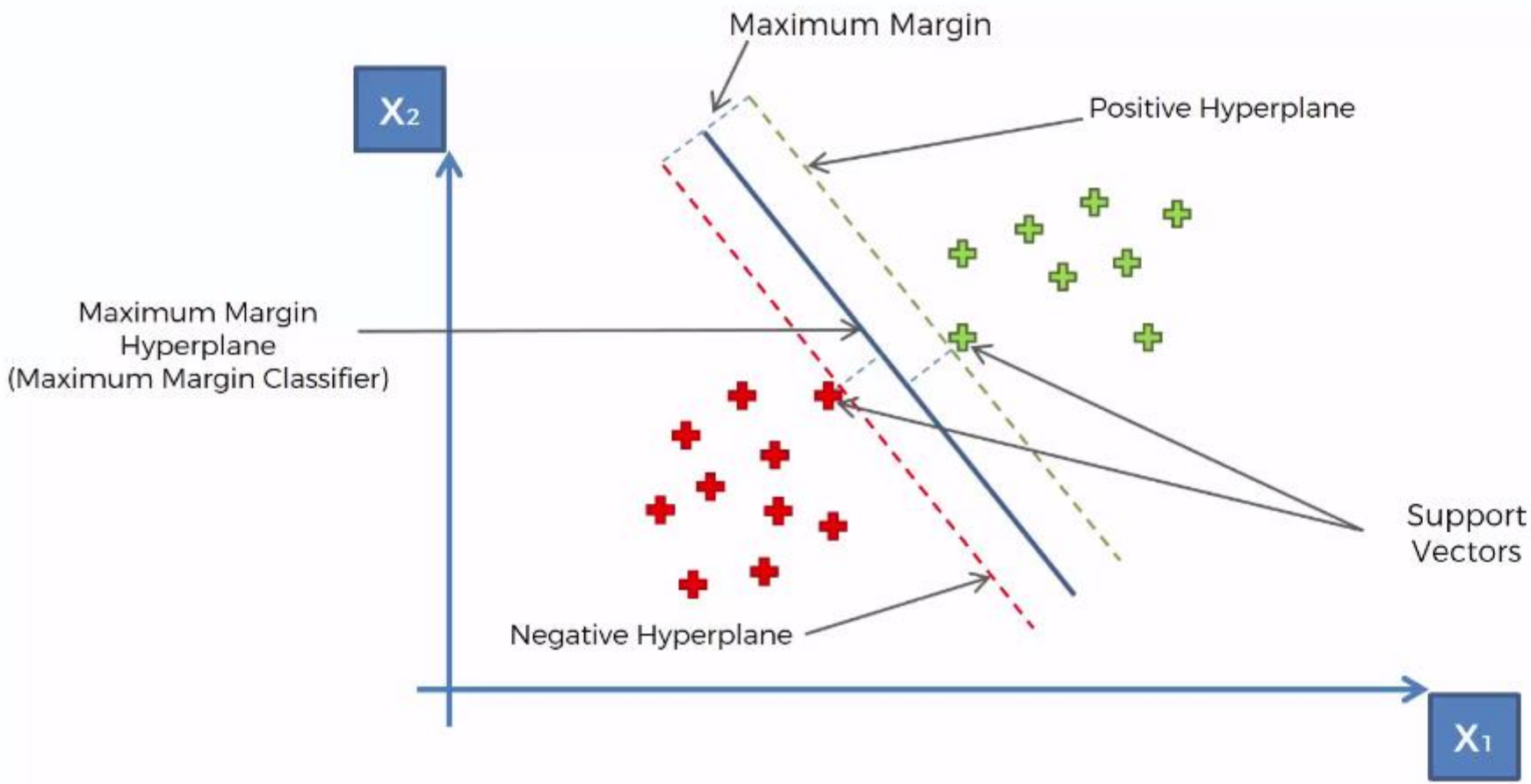
One with maximum margin is selected.



Which hyperplane does it select?

- Find the maximum margin between the hyperplanes that means maximum distances between the two classes.

Hyperplanes



Types of Support Vector Machine (SVM) Algorithms

- **Linear SVM**:
  - When the data is perfectly linearly separable only then we can use Linear SVM.
  - Perfectly linearly separable means that the data points can be classified into 2 classes by using a single straight line (if 2D).

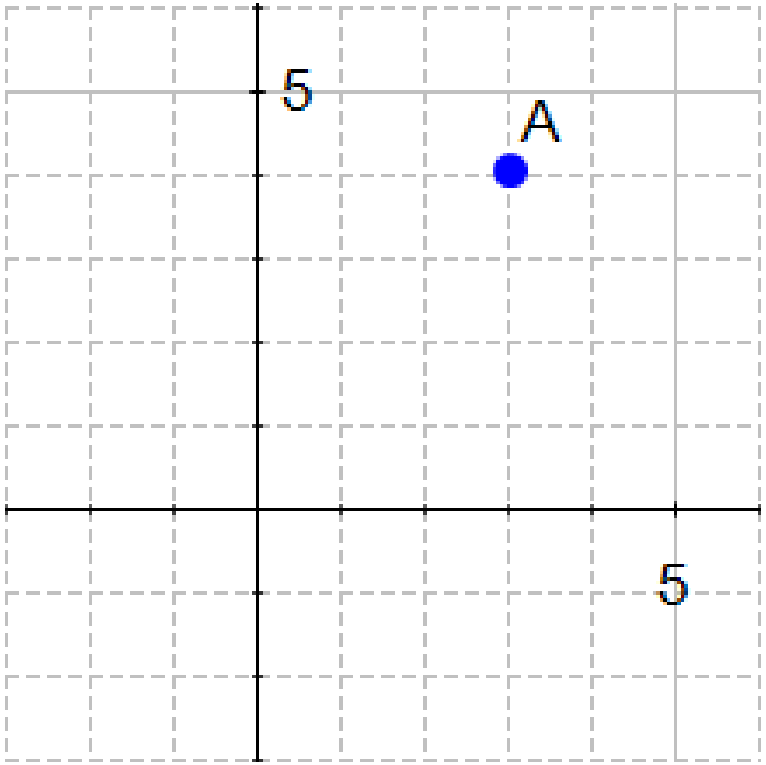
- **Non-Linear SVM**:
  - When the data is not linearly separable then we can use Non-Linear SVM, which means when the data points cannot be separated into 2 classes by using a straight line (if 2D) then we use some advanced techniques like kernel tricks to classify them.
  - In most real-world applications we do not find linearly separable datapoints hence we use kernel trick to solve them.

Mathematical Intuition Behind Support Vector Machine

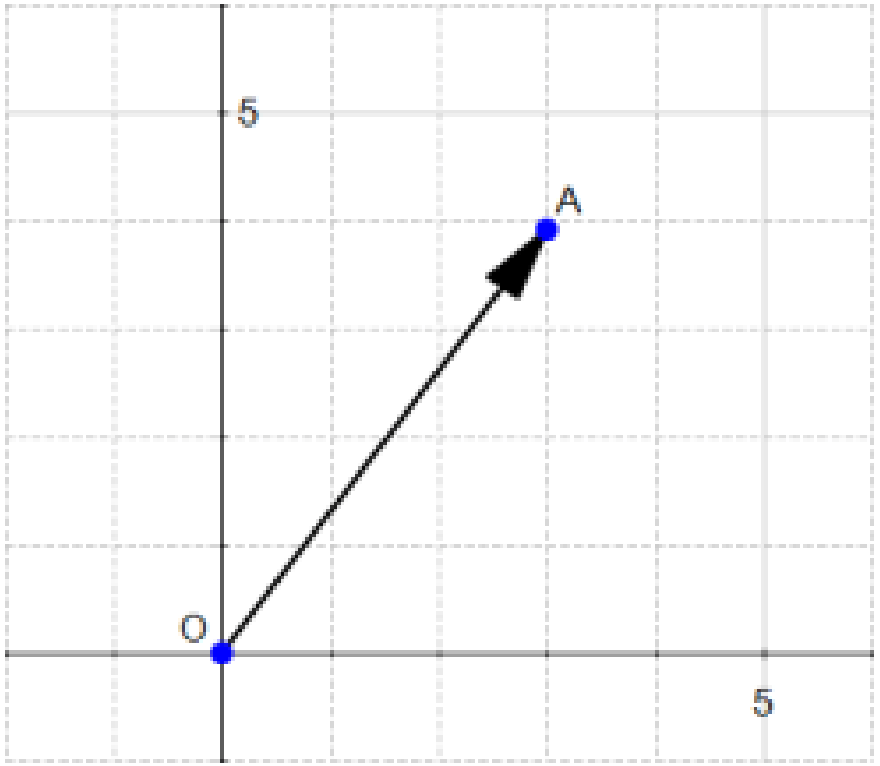
Vector

A vector is a quantity that has magnitude as well as direction and just like numbers we can use mathematical operations such as addition, multiplication.

If we define a point $A(3,4)$ in $\mathbb{R}^2$ we can plot it like this.



A vector in the plane, starting at the origin and ending at A



The magnitude

The magnitude or length of a vector X is written $\|X\|$ and is called its norm.

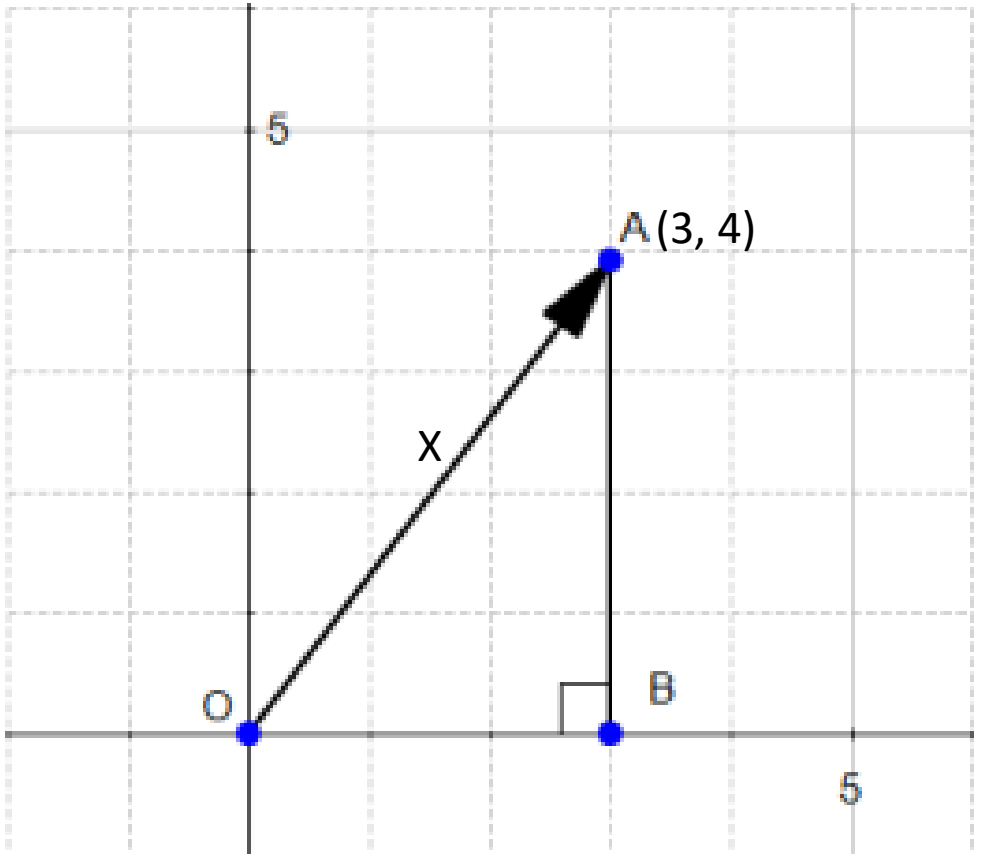
$$OA^2 = OB^2 + AB^2$$

$$OA^2 = 3^2 + 4^2$$

$$OA^2 = 25$$

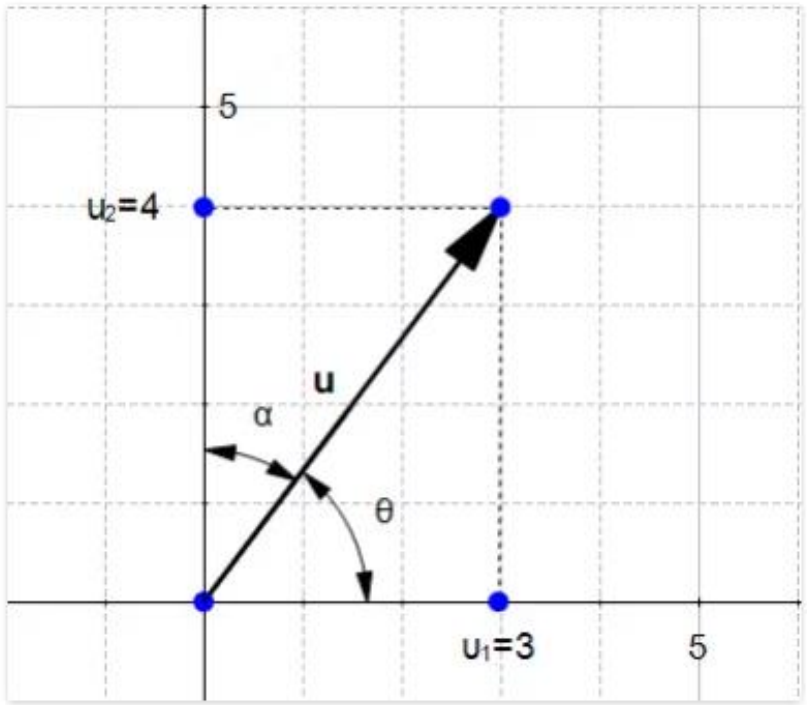
$$OA = \sqrt{25}$$

$$\|OA\| = OA = 5$$



The direction

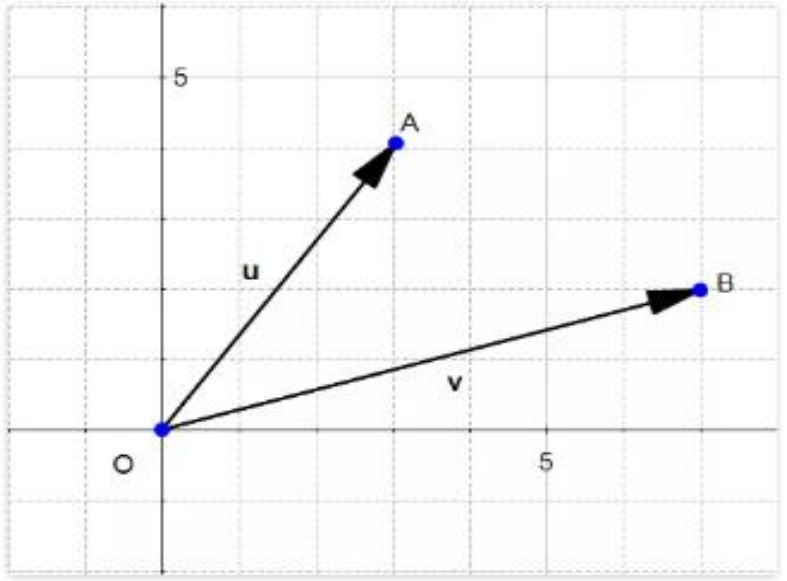
*The **direction** of a vector* $\mathbf{u}(u_1, u_2)$ *is the vector* $\mathbf{w}\left(\frac{u_1}{\|u\|}, \frac{u_2}{\|u\|}\right)$



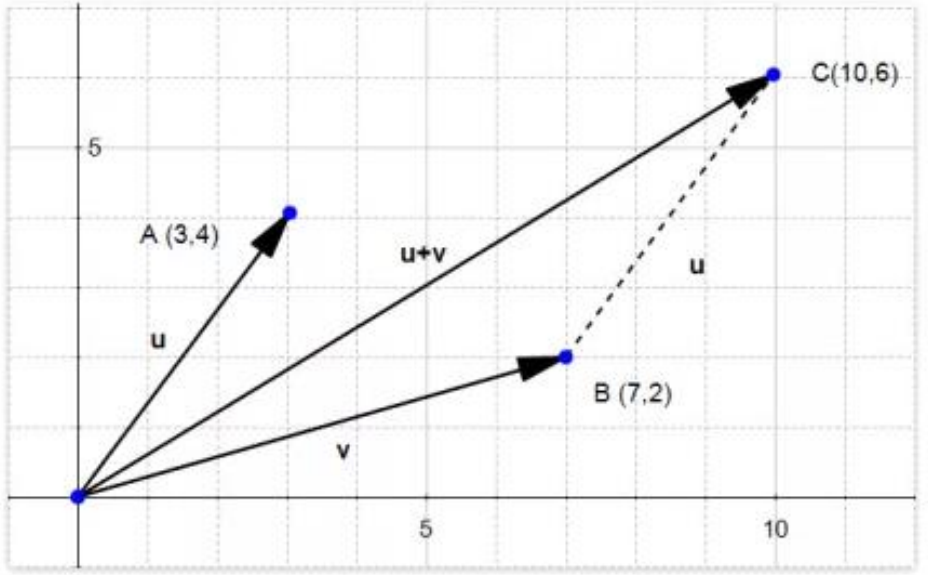
The sum of two vectors

Given two vectors $\mathbf{u}(u_1, u_2)$ and $\mathbf{v}(v_1, v_2)$ then :

$$\mathbf{u} + \mathbf{v} = (u_1 + v_1, u_2 + v_2)$$

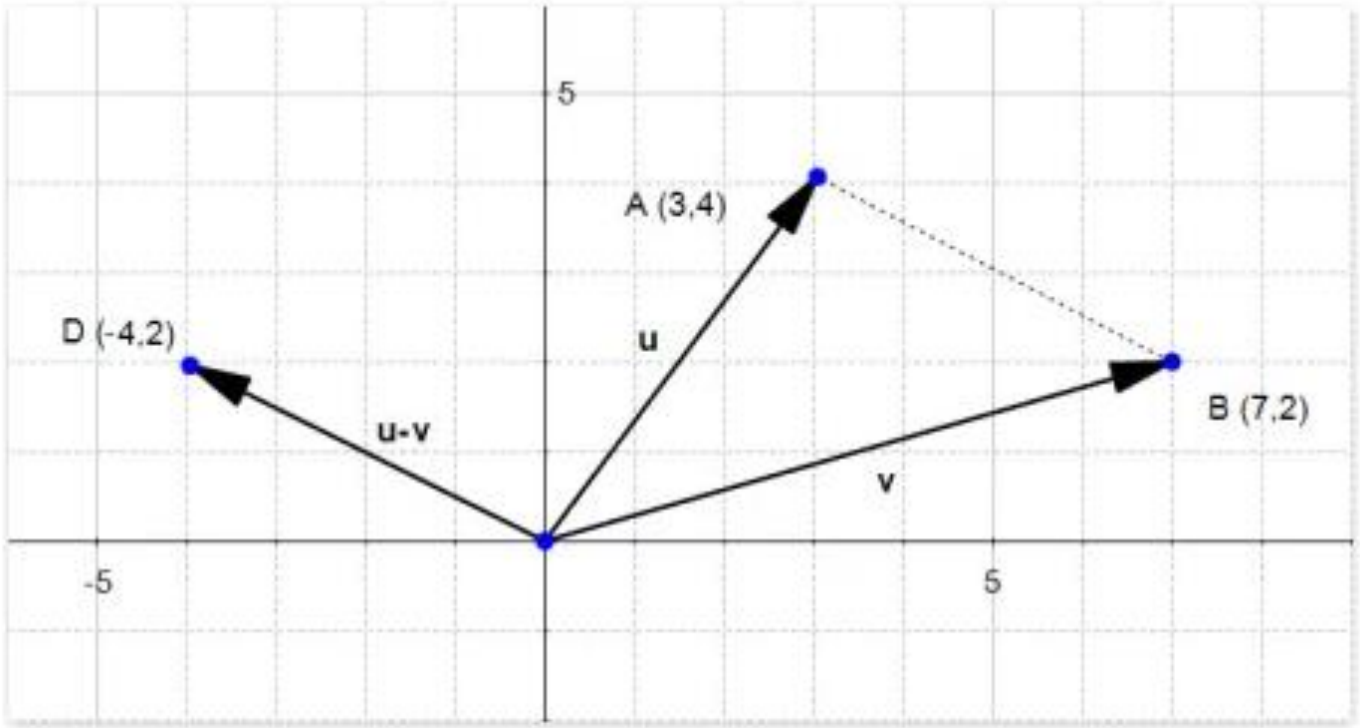


Which means that adding two vectors gives us **a third vector** whose coordinate are the sum of the coordinates of the original vectors.



The difference between two vectors

$$\mathbf{u} - \mathbf{v} = (u_1 - v_1, u_2 - v_2)$$



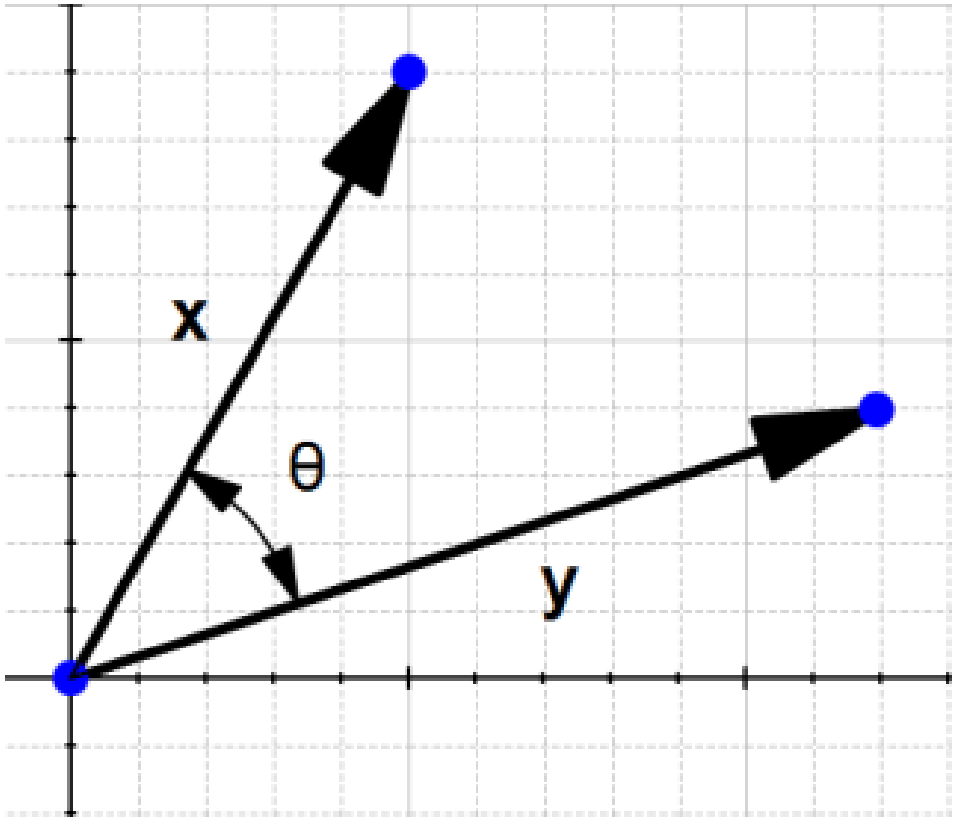
The dot product

- One **very** important notion to understand SVM is the dot product.

- If we have two vectors **x** and **y** and there is an angle **θ** between them, their **dot** product is :

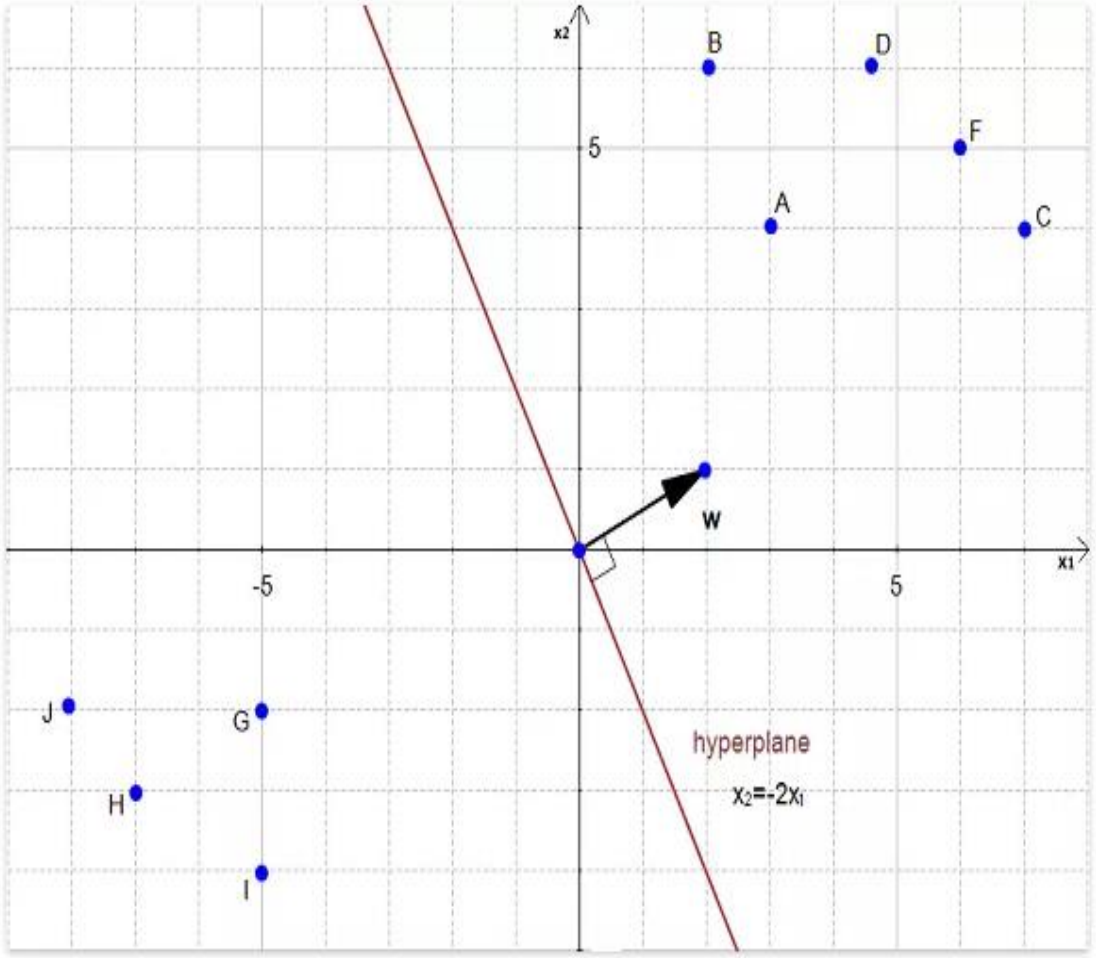
$$x \cdot y = \|x\| \, \|y\| \cos(\theta)$$

$$\cos(\theta) = \frac{adjacent}{hypotenuse}$$



Compute the distance from a point to the hyperplane

We have a Hyperplane, which separates two group of data.



Equation of the Hyperplane is $w^T x = 0$

To simplify this example, we have set $w_0 = 0$.

$$x_2 = -2x_1$$

$$x_2 + 2x_1 = 0$$

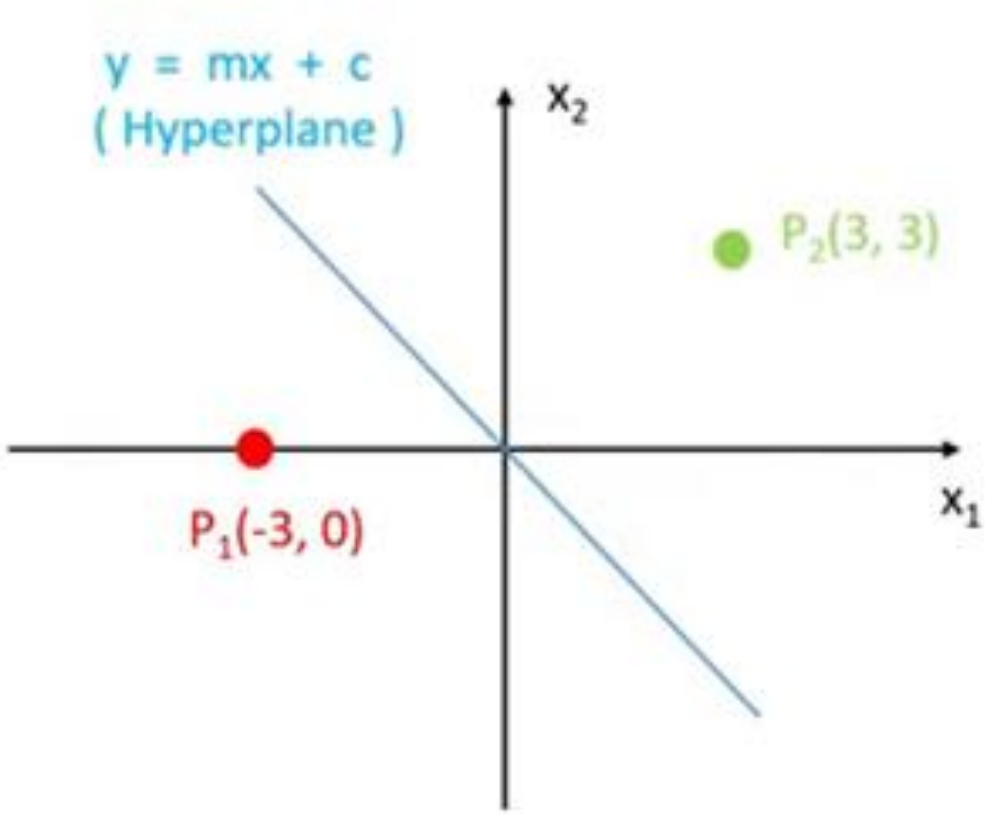
$$w^T x = 0$$

From these two vectors

$w = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ and $\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

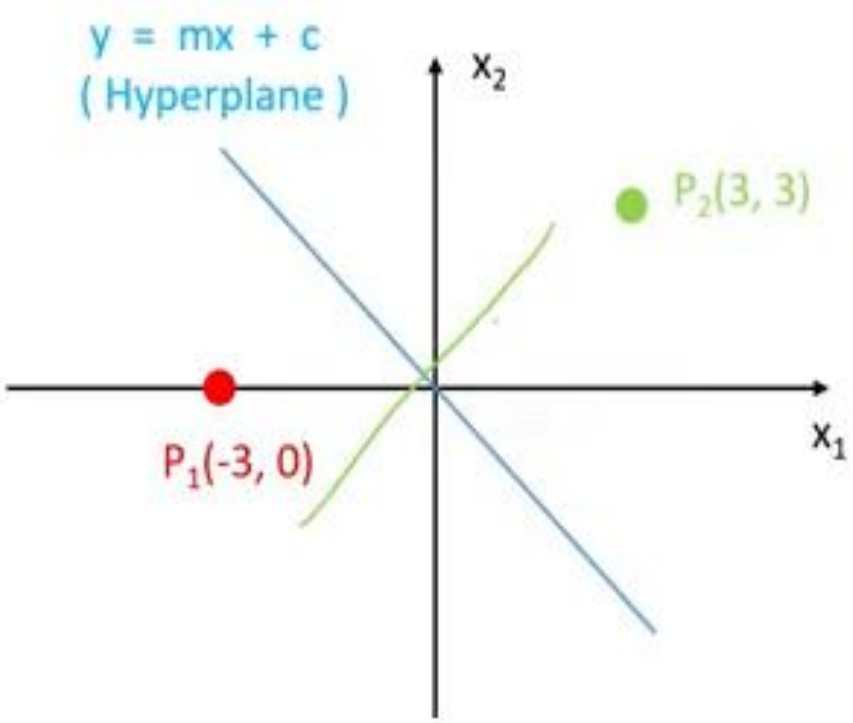
Classification of Data points

Classification of Data points: How to classify the points as +ve or –ve.



Let slope, m = - 1

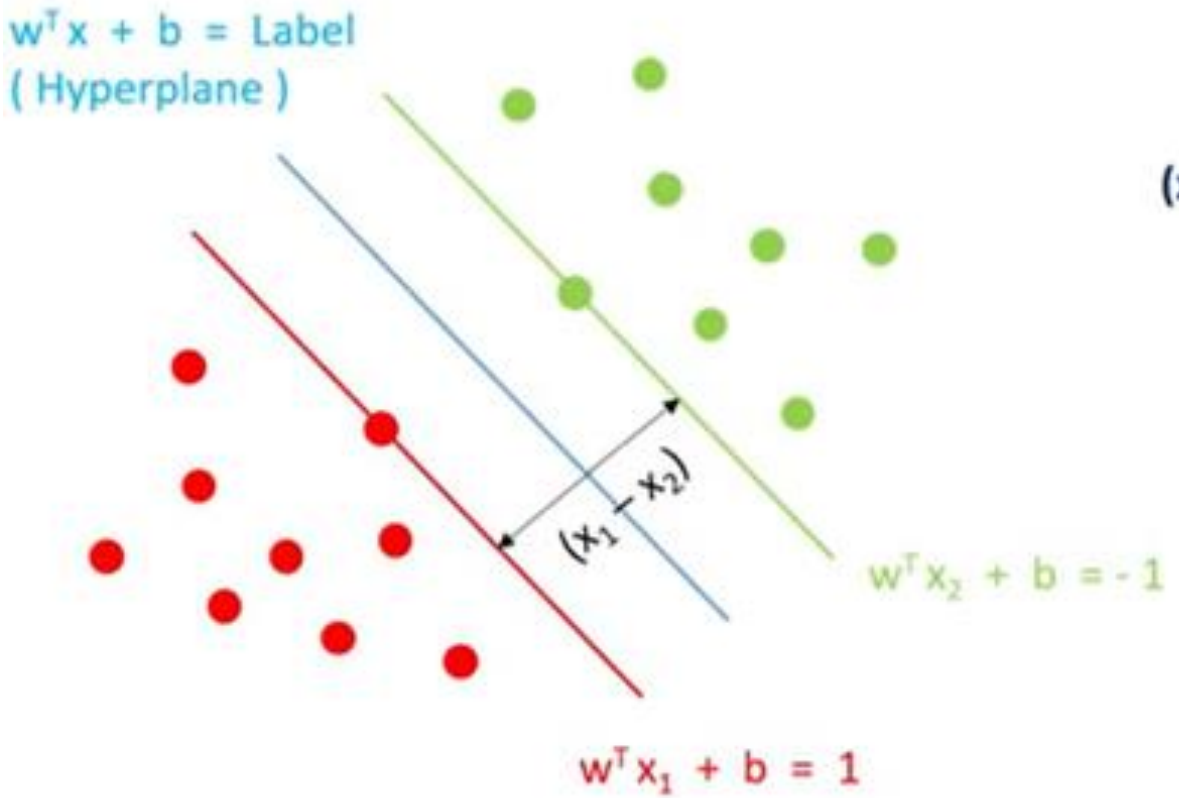
Intercept, c = 0



Let slope, m = - 1

Intercept, c = 0

Optimization for Maximum margin:



$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases} \quad \text{(Label)}$$

$$(x_1 - x_2) = \frac{2}{||w||} \quad \text{(margin)}$$

$$\max \left(\frac{2}{||w||} \right) \quad \text{Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

Soft Margin SVM

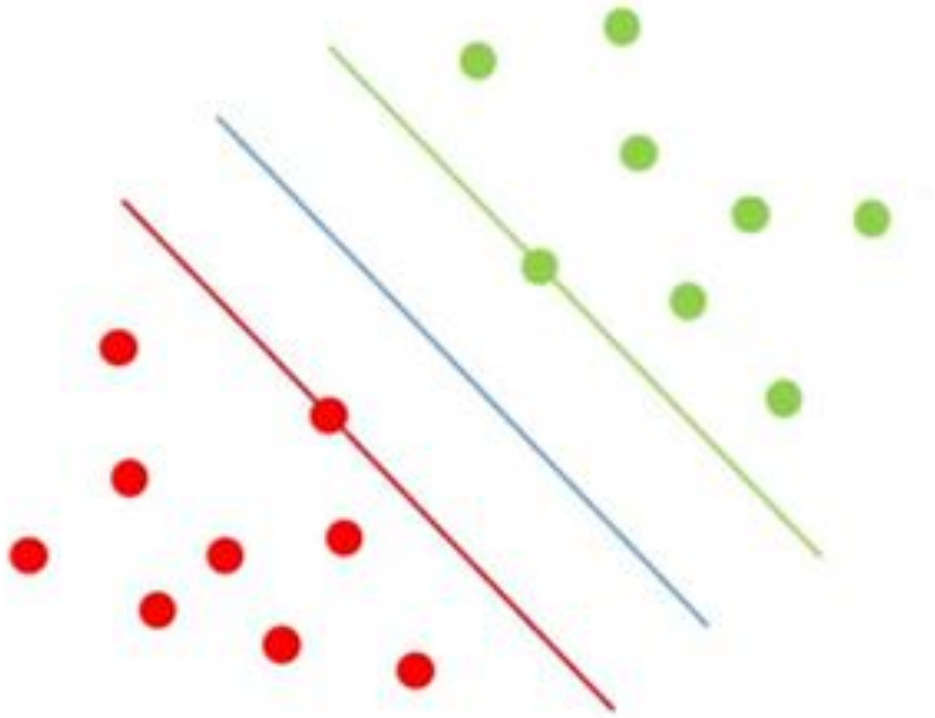
- In real-life applications, we rarely encounter datasets that are perfectly linearly separable.
- Instead, we often come across datasets that are either nearly linearly separable or entirely non-linearly separable.
- Unfortunately, the trick demonstrated above for linearly separable datasets is not applicable in these cases.
- This is soft margin classifier - Support Vector Machines (SVM) come into play.

Soft Margin SVM

- To handle non-linear data, we modify that equation in such a way that it allows few misclassifications that means it allows few points to be wrongly classified.

- We know that ***max[f(x)]*** can also be written as ***min[1/f(x)]***, it is common practice to minimize a cost function for optimization problems; therefore, we can invert the function.

Maximum margin without overfitting:



$$\max \left(\frac{2}{||w||} \right) \quad \text{Such that,}$$

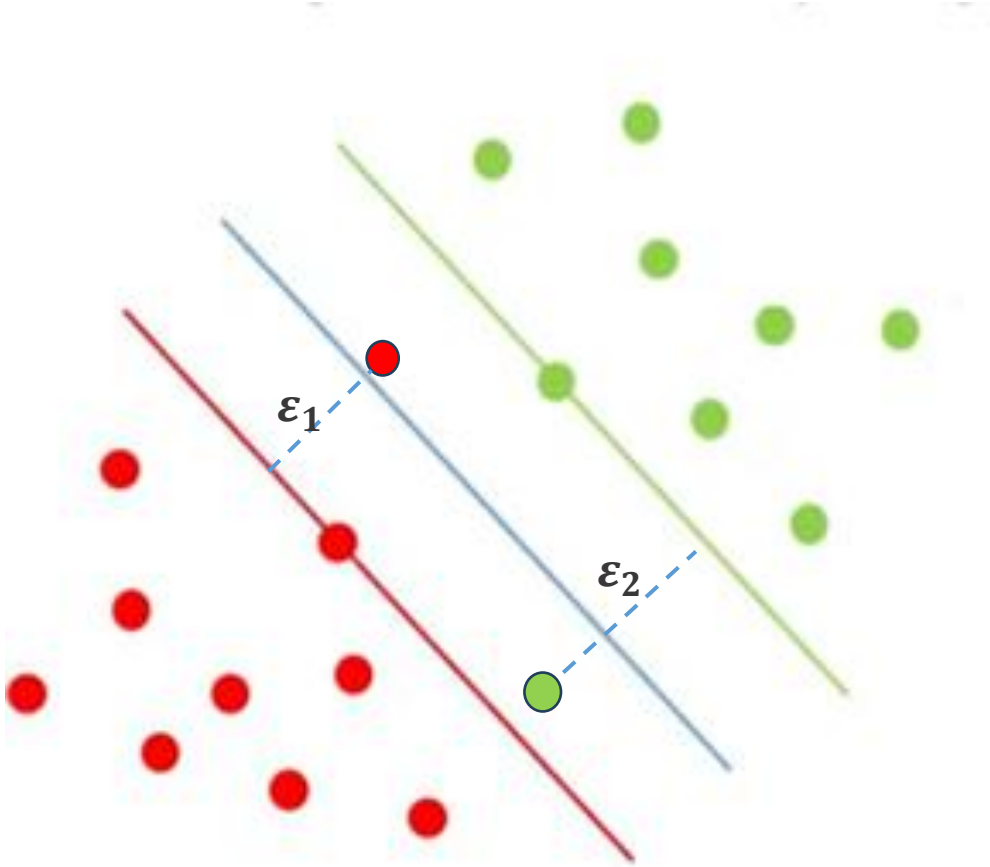
$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

This means

$$\min \left(\frac{||w||}{2} \right) + c * \Sigma \varepsilon_i$$

To make a soft margin equation we add 2 more terms to this equation which is ε and multiply that by a **hyperparameter '*c*'**

- For all the ***correctly classified*** points ε will be equal to 0 and for all the ***incorrectly classified*** points the ε is simply the distance of that particular point from its correct hyperplane.

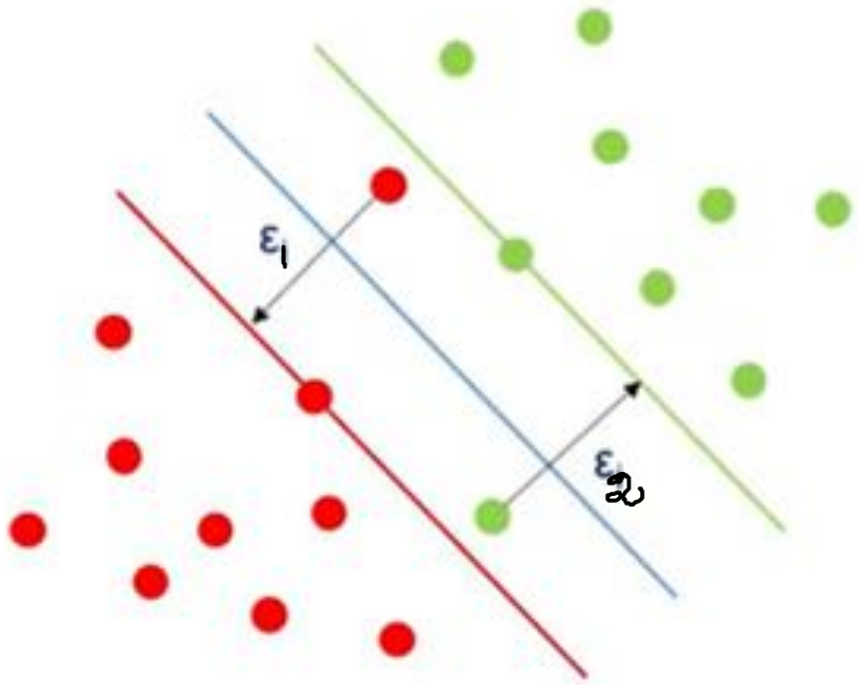


$$\max \left(\frac{2}{||w||} \right) \quad \text{Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

$$\min \left(\frac{||w||}{2} \right) + c * \sum \varepsilon_i$$

Maximum margin without overfitting:



$$\max \left(\frac{2}{||w||} \right) \quad \text{Such that,}$$

$$y_i = \begin{cases} -1, & w^T x_1 + b \leq -1 \\ 1, & w^T x_1 + b \geq 1 \end{cases}$$

$$\min \left(\frac{||w||}{2} \right) + c * \Sigma \varepsilon_i$$

c --> Number of errors = 2

ε_i --> Error magnitude

Linear Example

Suppose we are given the following positively labeled data points in $\Re^2$:

$$\left\{\begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \begin{pmatrix} 6 \\ 1 \end{pmatrix}, \begin{pmatrix} 6 \\ -1 \end{pmatrix}\right\}$$

and the following negatively labeled data points in $\Re^2$ (see Figure 1):

$$\left\{\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \end{pmatrix}\right\}$$

Linear Example

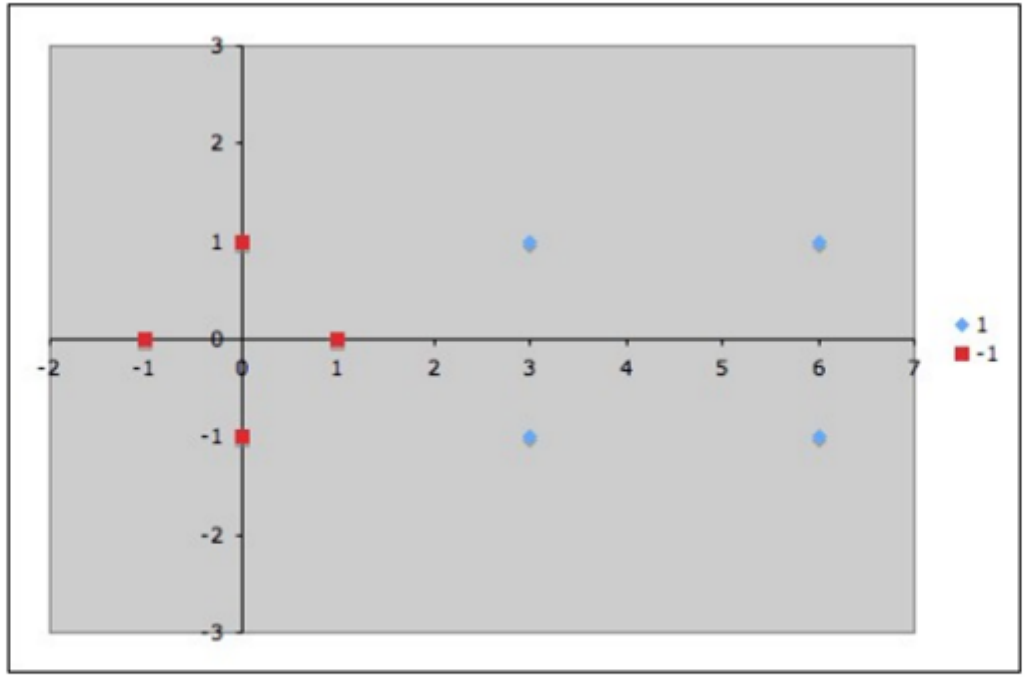


Figure 1: Sample data points in $\Re^2$. Blue diamonds are positive examples and red squares are negative examples.

Linear Example

- Discover a simple SVM that accurately discriminates the two classes.
- Since the data is linearly separable, we can use a linear SVM.
- There are three support vectors (see Figure 2)

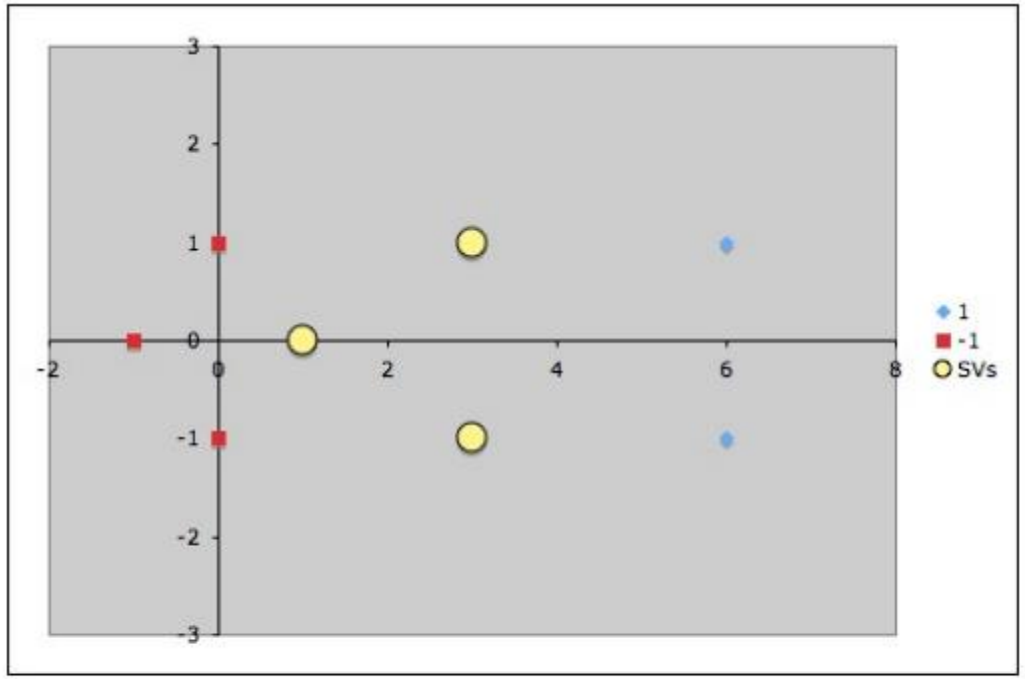
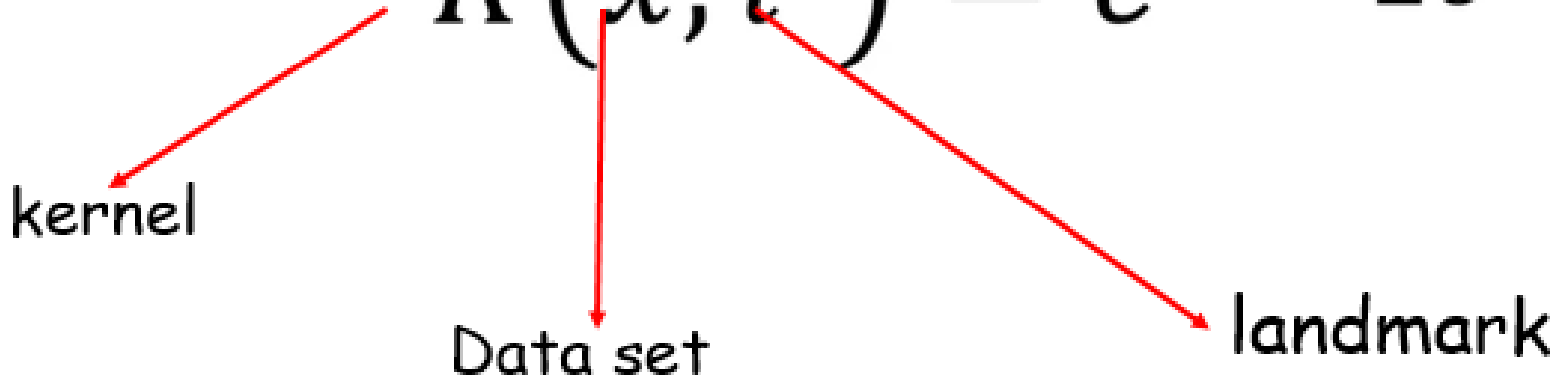


Figure 2: The three support vectors are marked as yellow circles.

The Kernel Trick

The Gaussian RBF Kernel

$$K(\vec{x}, \vec{l}^{i}) = e^{-\frac{\|\vec{x}-\vec{l}^{i}\|^2}{2\sigma^2}}$$



Types of Kernel Functions

Assignment

Kernel Methods:

- Types of kernels
- Dual Representations
- Design of Kernels

SVM

Nonlinear Example

Example

We are given the following positively labeled data points in $\Re^2$:

$$\left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix} \right\}$$

and the following negatively labeled data points in $\Re^2$ (see Figure 5):

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix} \right\}$$

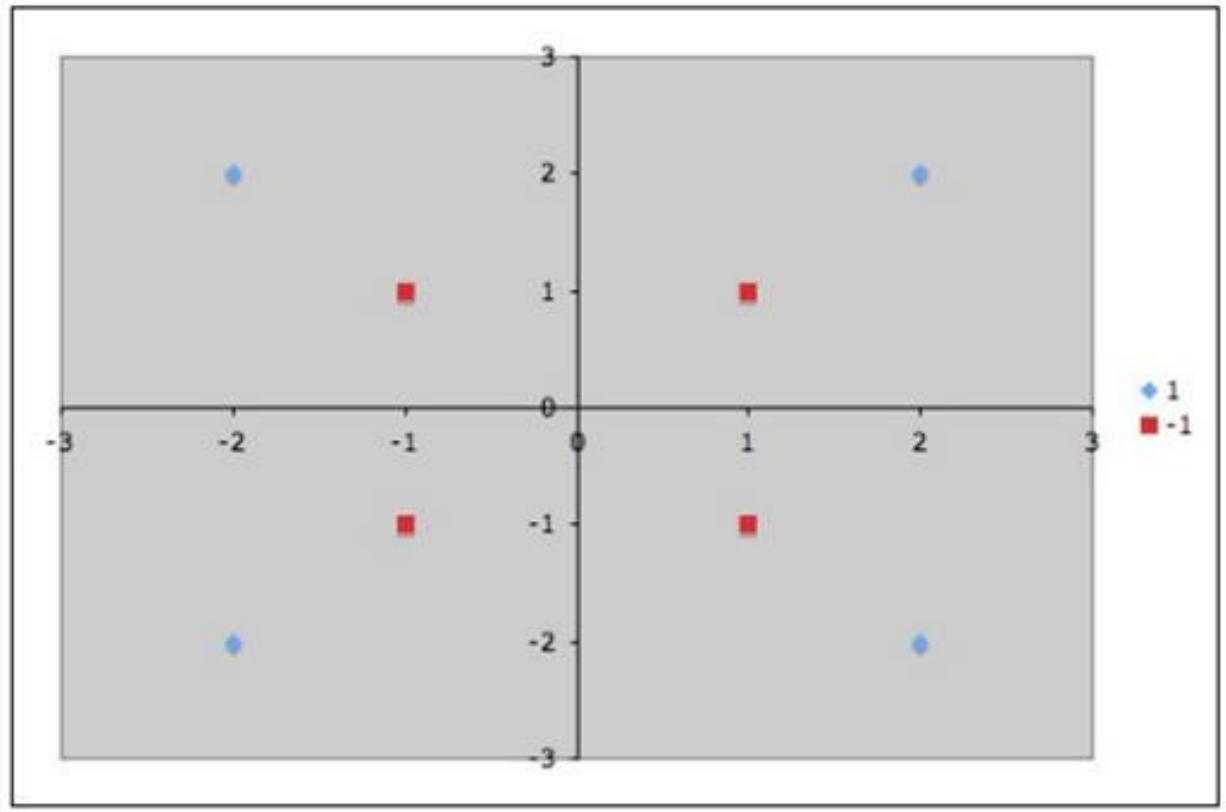


Figure 5: Nonlinearly separable sample data points in $\Re^2$. Blue diamonds are positive examples and red squares are negative examples.

Define

$$\Phi_1 \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{cases} \begin{pmatrix} 4 - x_2 + |x_1 - x_2| \\ 4 - x_1 + |x_1 - x_2| \end{pmatrix} & \text{if } \sqrt{x_1^2 + x_2^2} > 2 \\ \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} & \text{otherwise} \end{cases} \quad (1)$$

Example

- Discover a separating hyperplane that accurately discriminates the two classes.
- It is obvious that no such hyperplane exists in the input space.
- Therefore, we must use a nonlinear SVM.

$$\left\{ s_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, s_2 = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \right\}$$

We again use vectors augmented with a 1 as a bias input and will differentiate them as before. Now given the [augmented] support vectors, we must again find values for the α_i. This time our constraints are

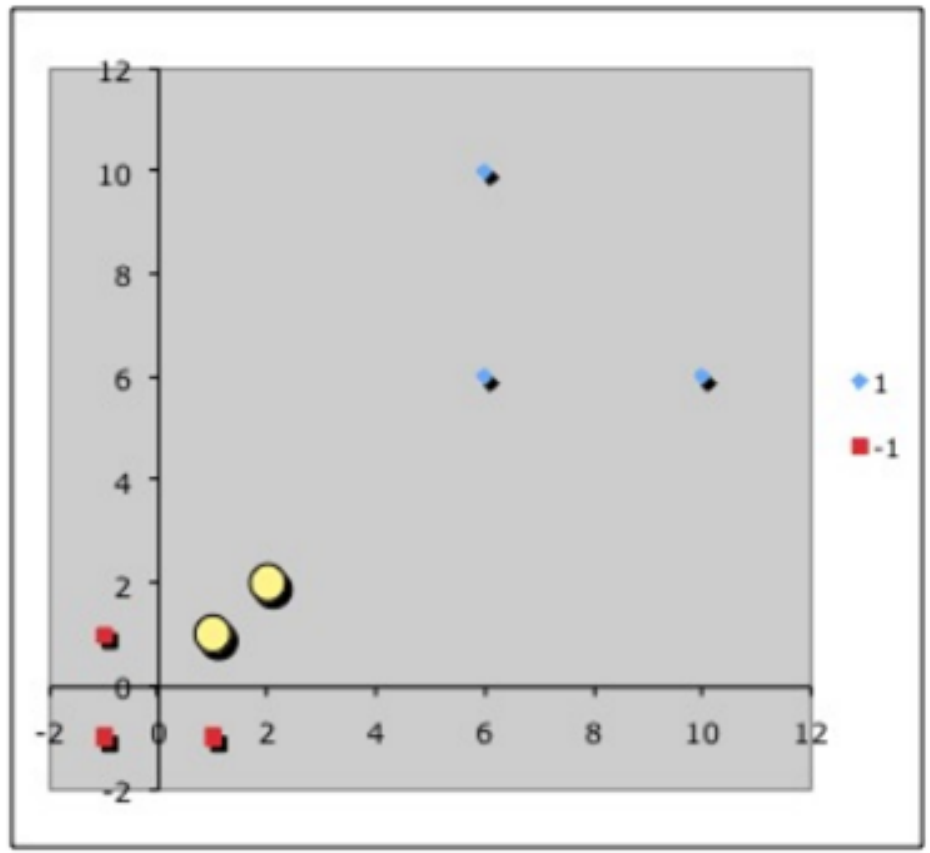


Figure 7: The two support vectors (in feature space) are marked as yellow circles.

Given Eq. 1, this reduces to

$$\begin{aligned} \alpha_1 \tilde{s_1} \cdot \tilde{s_1} + \alpha_2 \tilde{s_2} \cdot \tilde{s_1} &= -1 \\ \alpha_1 \tilde{s_1} \cdot \tilde{s_2} + \alpha_2 \tilde{s_2} \cdot \tilde{s_2} &= +1 \end{aligned}$$

(Note that even though Φ_1 is a nontrivial function, both s_1 and s_2 map to themselves under Φ_1. This will not be the case for other inputs as we will see later.)

Now, computing the dot products results in

$$\begin{aligned} 3\alpha_1 + 5\alpha_2 &= -1 \\ 5\alpha_1 + 9\alpha_2 &= +1 \end{aligned}$$

And the solution to this system of equations is $\alpha_1 = -7$ and $\alpha_2 = 4$.

Finally, we can again look at the discriminating hyperplane in input space that corresponds to these α.

$$\tilde{w} = \sum_i \alpha_i \tilde{s}_i$$

$$= -7 \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} + 4 \begin{pmatrix} 2 \\ 2 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 \\ 1 \\ -3 \end{pmatrix}$$

giving us the separating hyperplane equation $y = wx + b$ with $w = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ and $b = -3$. Plotting the line gives the expected decision surface (see Figure 8).

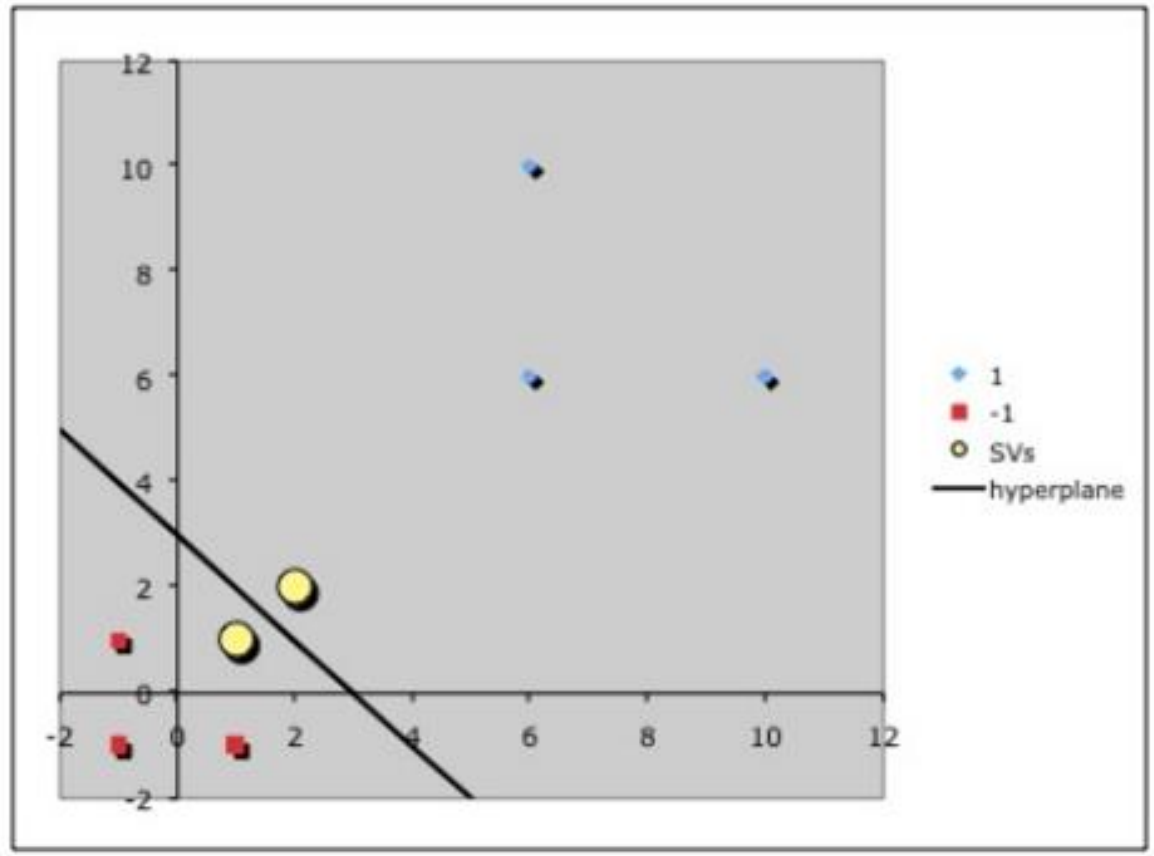
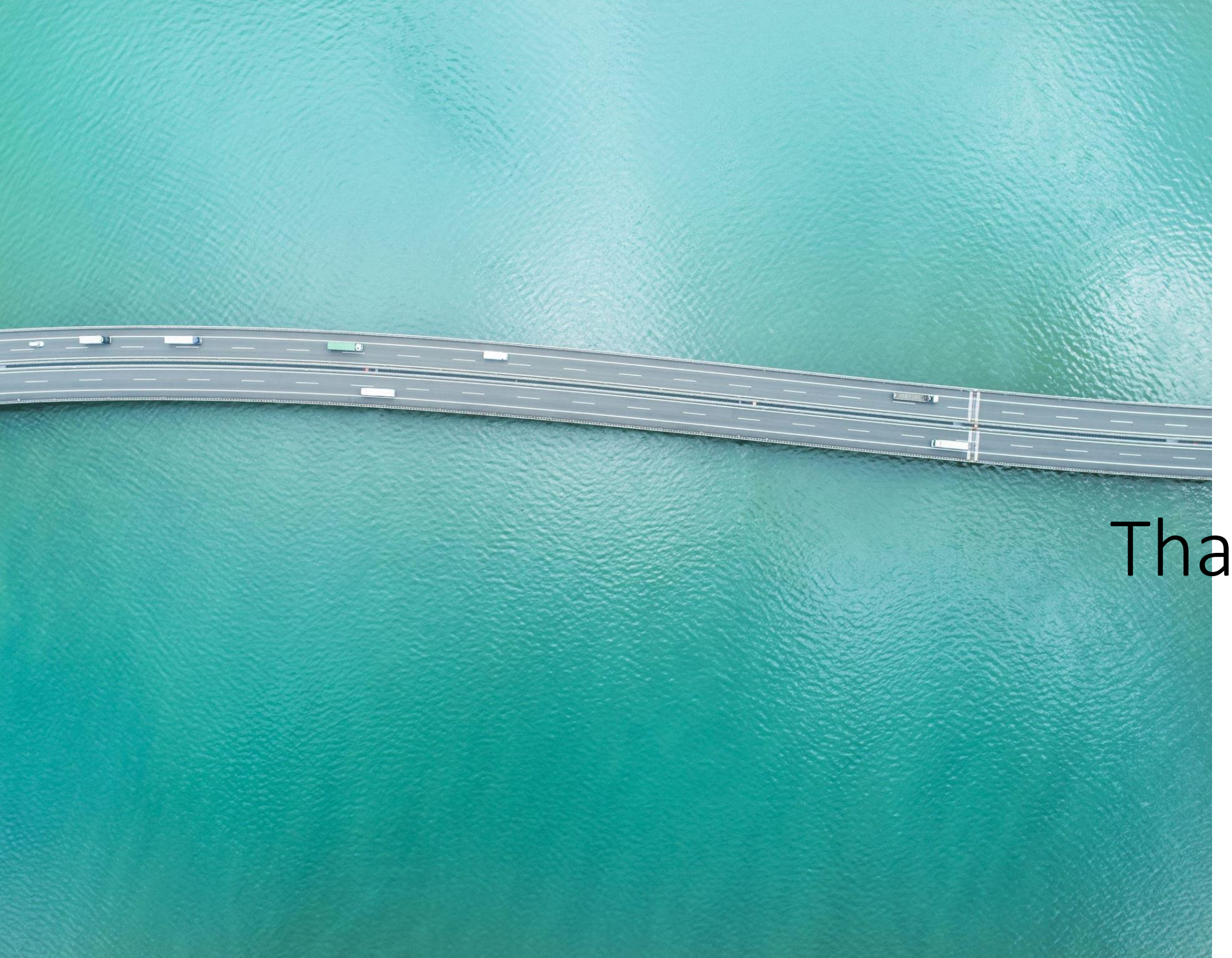


Figure 8: The discriminating hyperplane corresponding to the values $\alpha_1 = -7$ and $\alpha_2 = 4$



Thank you