Data Science and Artificial Intelligence

Machine Learning

Unsupervised learning

Lecture No. 1

By- SIDDHARTH SABHARWAL SIR

Recap of Previous Lecture

Topic — FFNN

Topic — Back propagation

Topic — Advantage & disadvantange

Topic

Topic

Turn on Slide map

Topics to be Covered

- Topic — NN Objective / Cross entropy.
- Topic — Kmeans Clustering
- Topic
- Topic
- Topic

REVISION

Back propagation

- Chain Rule of diff
- $\mathcal{L} = \frac{1}{2}(y_i - \hat{y}_i)^2$
- $w^{new} = \left(w^{old} - \eta \frac{\partial \mathcal{L}}{\partial w}\right)$

Back propagation

What is Back Propagation

- Lets learn the back propagation…

Perceptron learning

Practise

Question 2: You are training a multi-layer perceptron (MLP) with 3 hidden layers. The first hidden layer has 50 neurons, the second has 30, and the third has 20. How many total neurons are in the hidden layers?

50+30+20

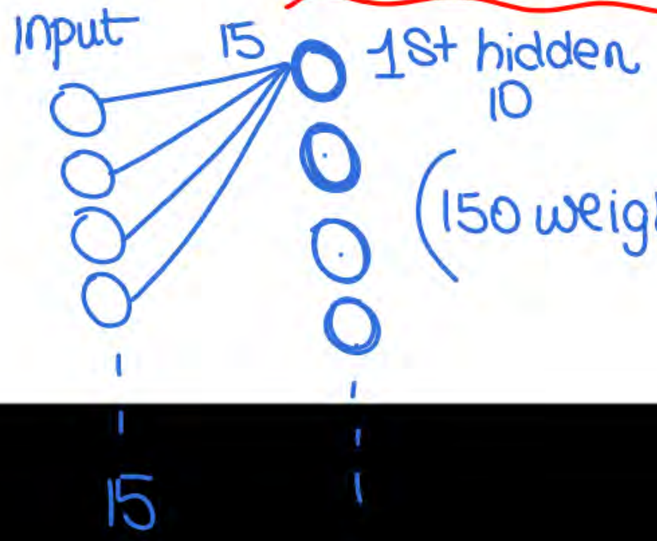
A. 50
B. 80
C. 100
D. 150

Perceptron learning

Practise

Question 3: In a feed-forward neural network, you have an input layer with 15 features. The first hidden layer has 10 neurons, and the second hidden layer has 5 neurons. How many weights are there in the connections between the input and the first hidden layer?

A. 150
B. 155
C. 165
D. 170



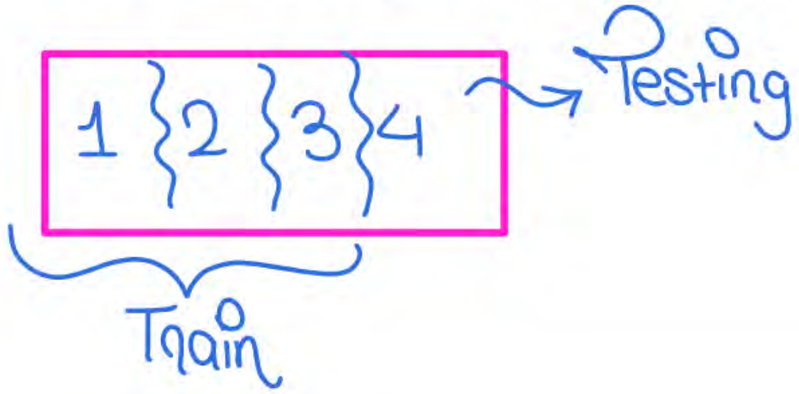
(150 weights) + 10

Perceptron learning

Practise

Question 5: In a 4-fold cross-validation, what percentage of the data is used for testing in each fold?

A. 25%
B. 30%
C. 40%
D. 50%



Perceptron learning

Practise

Question 12: You have a multi-layer perceptron (MLP) with an input layer of 40 features and a hidden layer with 30 neurons. How many weights are there connecting the input to the hidden layer, not considering bias terms?

40x30

A. 1200
B. 30
C. 40
D. 70

Perceptron learning

Practise

not in syllabus

The primary difference between FFNN and recurrent neural networks (RNN) is:

- A) FFNNs have memory while RNNs do not.
- B) RNNs have memory while FFNNs do not.
- C) FFNNs are used for time-series data.
- D) RNNs can only have one hidden layer.

Not in Syllabus

the o/p of some neurons @ higher layers is fed back as input to the neurons of lower layers.

Perceptron learning

Practise

In a feedforward neural network, the information moves:

- A) Forward and backward.
- B) Only forward, from input to output.
- C) In a cyclic manner.
- D) In any random direction.

Perceptron learning

Practise

A feedforward neural network:

- A) Allows connections to form cycles.
- B) Does not allow connections to form cycles.
- C) Is the same as a recurrent neural network.
- D) Only consists of one hidden layer.

Perceptron learning

Practise

The universal approximation theorem states that:

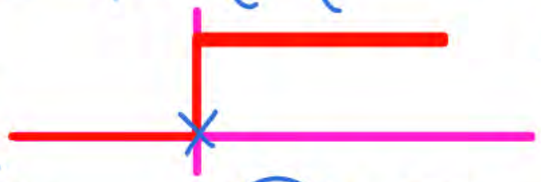
- A) A single-layer perceptron can approximate any function.
- B) An MLP with at least one hidden layer can approximate any continuous function given enough neurons.
- C) An MLP with two hidden layers can solve the XOR problem.
- D) MLPs are only useful for classification tasks.

Perceptron learning

Practise

In a multi-layer perceptron, the activation function used in hidden layers is typically:

- A) Linear → not linear bcoz hidden layer ka kaam transformation
- B) Step → not diff fxn



diff = 0,

- C) Non-linear (e.g., ReLU, Sigmoid, Tanh) → Vanishing gradient $\frac{\partial L}{\partial w} \approx 0$
- D) None of the above

Perceptron learning

Practise

Which of the following is true about single-layer perceptrons?

→ input layer & o/p layer

- A) They can solve XOR problems.
- B) They can have multiple hidden layers.
- C) They are the building blocks of deep neural networks.
- D) They can only solve linearly separable problems.

Perceptron learning

Practise

The learning rule used by the single-layer perceptron is known as:

- A) Hebbian Learning
- B) Gradient Descent
- C) Delta Rule
- D) Backpropagation

not Robust to outlier

- MSE ⇒ $\frac{1}{2N}\sum_{i=1}^{N}(y_i-\hat{y}_i)^2$
- MAE ⇒ $\frac{1}{N}\sum_{i=1}^{N}|y_i-\hat{y}_i|$

Problem any outlier point → Outlier effect our algo.

⇒ we can say that

loss ⇒ $\frac{1}{N}\sum|y_i-\hat{y}_i|$ $|y_i-\hat{y}_i|>\delta$

$\frac{1}{2N}\sum(y_i-\hat{y}_i)^2$ $|y_i-\hat{y}_i|<\delta$

Logistic Reg $\Rightarrow$ MLE $\Rightarrow \left(\prod_{i=1}^{N} (P)^{y_i} (1-P)^{1-y_i}\right)^{max}$

- P = Probab that point belong to class 1
- So if $y_i = 1$ then max P for that point
- But if $y_i = 0$ then max $(1-P)$

$$\Rightarrow Max\left\{\log\left(\pi P^{Y_i}(1-P)^{1-y_i}\right)\right\}$$

$$\Rightarrow \sum \log\left(P^{y_i} \cdot (1-P)^{1-y_i}\right)$$

$$\Rightarrow \sum \left\{y_i \log P + (1-y_i)\log(1-P)\right\}^{\underline{\underline{max}}}$$

Perceptron learning

What is cross entropy loss function

1.The cross-entropy loss function, also known as log loss, is a commonly used loss function in classification problems, particularly in binary and multiclass classification tasks involving neural networks. It measures the performance of a classification model whose output is a probability value between 0 and 1.

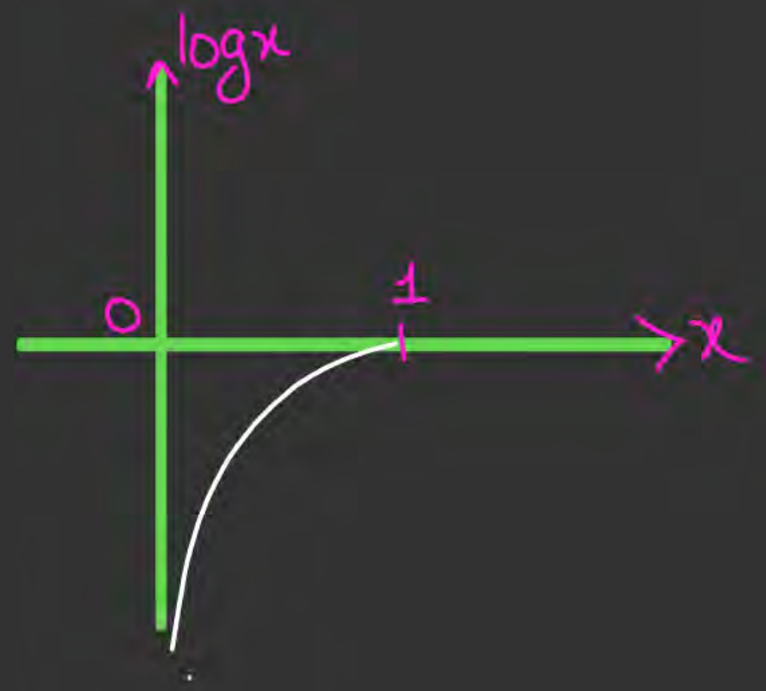
$$\boxed{L = -\left[y\log(p) + (1-y)\log(1-p)\right]}$$

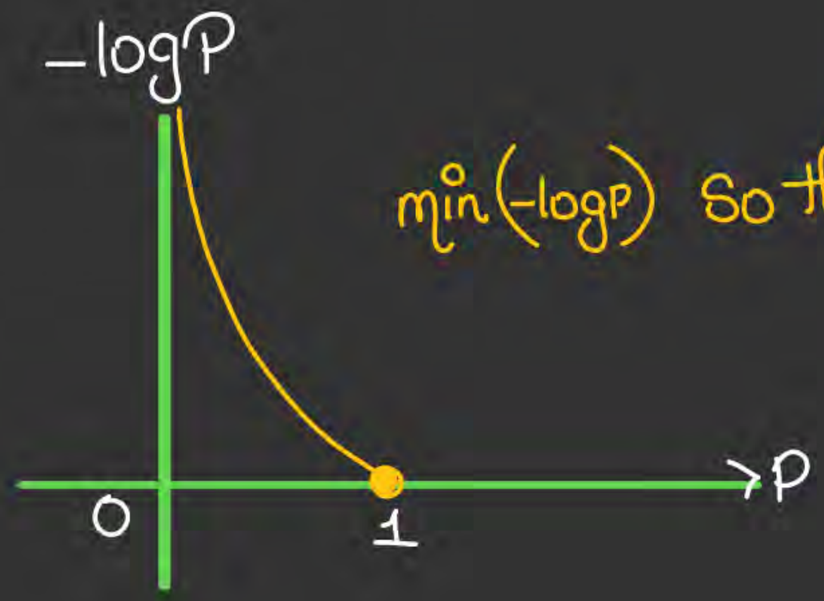
→ now minimize this.

where:

- y is the true label (0 or 1).
- p is the predicted probability of the instance being in class 1.

- So if point is of class 1 ⇒ I want $P \approx 1$
  So loss fxn tryto keep $P \Rightarrow 1$
- If point belong to Class 0, $y=0$
  min $-\log(1-p)$
  Only when $P=0$.





min(−log p) so this min when P = 1.

Clustering

Clustering Analysis

Unsupervised learning

So we want to create K subsets within the data which are similar ...

- data has no labels.
- Our task is to create K number of subsets from data such that each subset has points similar to each other.

K number of groups

- Similarity metric ⇒ euclidean distance b/w points.

Clustering Analysis

- We want that the samples within the clusters/subsets are similar…
- What will be the measure of similarity..

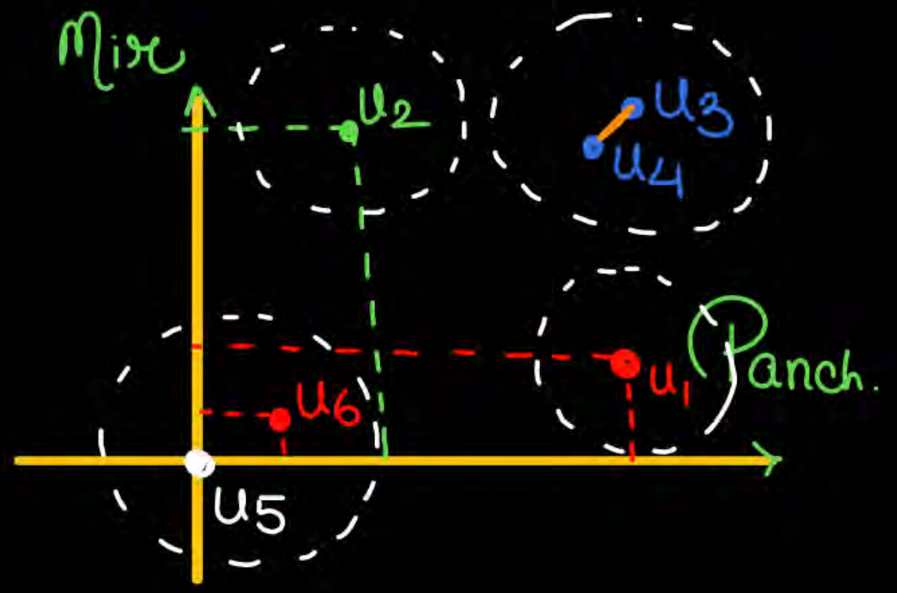
Clustering Analysis

Distance between the points of a clusters < Distance between points of different cluster

Clustering

Amazon Prime

	Mirzapur	Panchayat
U_1	1	5
U_2	5	1
U_3	5	5
U_4	4.5	4.5
U_5	0	0
U_6	0.5	0.5



Clustering

Where can we use Clustering

- Marketing and Customer Segmentation: Identifying distinct customer groups based on purchasing behavior, demographics, and preferences. Tailoring marketing strategies to different customer segments.
- Healthcare and Medicine: Grouping patients with similar symptoms or genetic profiles to better understand diseases.
- Image and Pattern Recognition: Segmenting images into different regions for object detection and recognition. Grouping similar images for indexing and retrieval in large databases.
- Social Network Analysis: Identifying communities or groups within social networks. Analyzing user behavior and interactions on social media platforms.
- Financial Analysis: Segmenting companies or stocks based on financial performance and characteristics. Identifying patterns in trading behavior and market conditions.

Clustering

Where can we use clustering ??

Lets see an example of Clustering ….. Amazon Prime

Type of cluster Analysis

- Partition bases clustering (flat)
  - K mean and K Medoid Clustering
- Hierarchical Clustering
  - Bottom Up – Agglomerative
  - Top down – Divisive

K-Means Algorithms

- K is the number of clusters (thus K is a Hyperparameter)

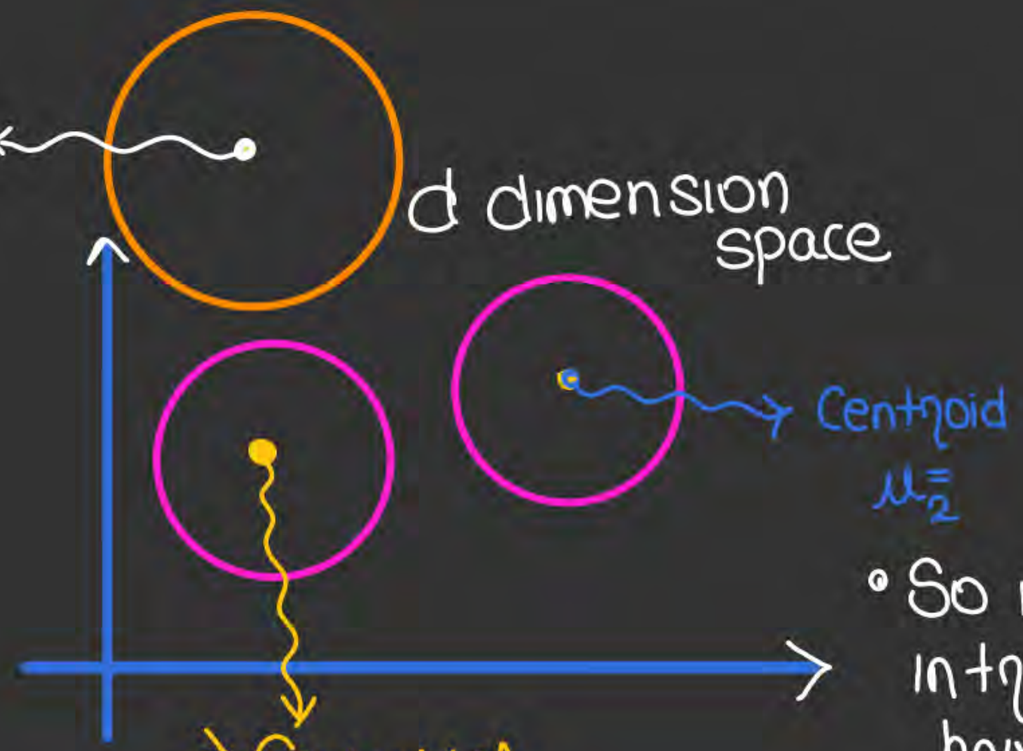
K mean Clustering

(No of clusters to be created)

- No cross Validation as Data is not labled.
- No Testing

K-Means Algorithms

- K means clustering, assigns data points to one of the K clusters depending on their distance from the center of the clusters.
- It starts by **randomly assigning** the clusters centroid in the space.
- Then each data point assign to one of the cluster based on its distance from centroid of the cluster. After assigning each point to one of the cluster, new cluster centroids are assigned.



- So in K mean clustering in training process we have to find K centroids of clusters $u_1, u_2, ---u_K$

⇒ Centroid of Cluster ⇒ μ

D dimensional

$$\mu_1 = \frac{\text{Summation of all Points in Cluster}}{\text{No of points in cluster}}$$

- we need centroids so that the cluster of any new point can be decided ⇒

How ⇒ any new point ⇒ x_{new} (D dimensional), find E.D of x_{new} from all μ,

$$\|x_{new} - \mu_1\|^2$$
$$\|x_{new} - \mu_2\|^2$$
$$\vdots$$
$$\|x_{new} - \mu_K\|^2$$

* So the centroid which has min distance will be assigned to the point

Clustering

How to find centroids ⇒ we have data without labels.

K-Means Algorithms

- **Algorithm :** we have to learn K centroids of K clusters.

→ 1. Initialize k means with random values ($\mu_1, \mu_2, \mu_3 \ldots \mu_K$ ⇒ Randomly any Values)

2. Now find eucledian distance of all points in training data with all these Randomly assigned K means or K centroids.

So we can divide points of training data into Cluster 1, Cluster 2, --- Cluster K.

$$\mu_1^{new} = \frac{\text{Sum of all points in Cluster 1}}{\text{No of points in cluster 1}}, \quad \mu_2^{new} \Rightarrow \frac{\text{Sum of all points in Cluster 2}}{\text{No of Points in Cluster 2}}$$

---- update all u's.

$$\underbrace{\left\| u_i^{new} - u_i^{old} \right\|^2}$$

Shows distance b/w new $old Centroid

$$\sum_{i=1}^{K} \left\| u_i^{new} - u_i^{old} \right\|$$ if this is $< \varphi$ then stop

Else Repeat step 2.

Clustering

Lets see an example

Point 1: (2, 10)
Point 2: (2, 5)
Point 3: (8, 4)
Point 4: (5, 8)
Point 5: (7, 5)
Initialize the centroid as
Centroïde 1 (C1): (2, 10)
Centroïde 2 (C2): (8, 4)
Find out the new centroïde after itération one...

Initialize C_1 (2,10) C_2 (8,4)

now find distance of all points with the centroids

C_1	1. (2,10)	0	$\sqrt{6^2+6^2} \Rightarrow \sqrt{72}$
C_1	2. (2,5)	$\sqrt{5^2}$	$\sqrt{6^2+1^2}$
C_2	3. (8,4)	$\sqrt{6^2+6^2}$	0
C_1	4. (5,8)	$\sqrt{3^2+2^2}$	$\sqrt{3^2+4^2}$
C_2	5. (7,5)	$\sqrt{5^2+5^2}$	$\sqrt{1^2+1^2}$

C_1: 2,10 | C_2: 8,4

2,5 | 7,5

5,8

$\mu_1 \Rightarrow \frac{5+2+2}{3}, \frac{8+5+10}{3}$; $\mu_2 = \frac{8+7}{2}, \frac{4+5}{2}$

$\Rightarrow \left(3, \frac{23}{3}\right)$ | $\Rightarrow 7.5, 4.5$

Initialize; C_1 (2,10) C_2 (8,4)

now find distance of all points with the centroids

C_1	1. (2,10)	0	$\sqrt{6^2+6^2} \Rightarrow \sqrt{72}$
C_1	2. (2,5)	$\sqrt{5^2}$	$\sqrt{6^2+1^2}$
C_2	3. (8,4)	$\sqrt{6^2+6^2}$	0
C_1	4. (5,8)	$\sqrt{3^2+2^2}$	$\sqrt{3^2+4^2}$
C_2	5. (7,5)	$\sqrt{5^2+5^2}$	$\sqrt{1^2+1^2}$

C_1: 2,10 | C_2: 8,4

2,5 | 7,5

5,8

$\mu_1 \Rightarrow \frac{5+2+2}{3}, \frac{8+5+10}{3}$; $\mu_2 = \frac{8+7}{2}, \frac{4+5}{2}$

$\Rightarrow \left(3, \frac{23}{3}\right)$ | $\Rightarrow 7.5, 4.5$

Iteration 1 end.

2nd iteration

Now again $\mu_1^{new} \Rightarrow$ $\mu_2^{new} \Rightarrow$

updated centroid.

$C_1 \left(3, \frac{23}{3}\right)$ $C_2 (7.5, 4.5)$

now find distance of all points with the centroids

C_1	1. (2,10)	d_1	d_2	$(d_2 > d_1)$
C_2	2. (2,5)	d_3	d_4	$d_4 < d_3$
C_2	3. (8,4)	d_5	d_6	$d_6 < d_5$
C_1	4. (5,8)	d_7	d_8	$d_7 < d_8$
C_1	5. (7,5)	d_9	d_{10}	$d_9 < d_{10}$

Step1 ⇒ Initialize u_K's ⇒ $u_1^1, u_2^1 \ldots u_K^1$

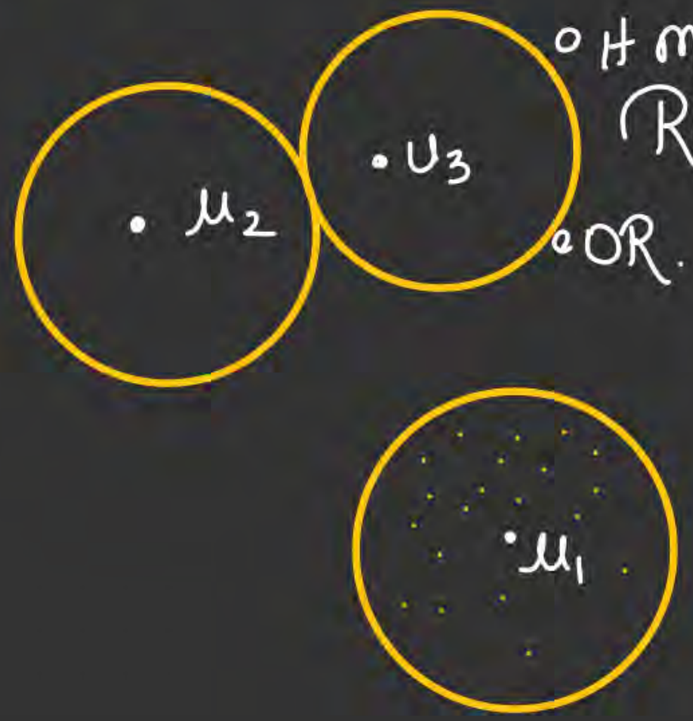
Step2 Find distance of all training data from all u_K's and assign clusters.

$$u_K^{new} \Rightarrow \frac{\sum \text{Sum of all points in cluster}}{\text{No of points in cluster}}$$

$$\sum_{i=1}^{K} \left\| u_i^{new} - u_i^{old} \right\|^2 > \varphi$$

If Less than φ Stop.

- What will happen when iteration ⇓



- It means now centroids are Ready
- OR. Now the points in cluster are not Changing ⇒ mean end Result of K mean clustering ⇒ Clusters of points which has distance from their less than distance from any other centroid.

Lets see an example

Given a set of observations (x1, x2, ..., xn), where each observation is a D-dimensional real vector, k-means clustering aims to partition the n observations into k (≤ n) sets S = {S1, S2, ..., Sk} so as to minimize the within-cluster sum of squares (WCSS) (i.e. variance).

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

where $\boldsymbol{\mu}_i$ is the mean (also called centroid) of points in S_i, i.e.

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$$

K-Means Clustering

- The objective function in k-means is the WCSS (within cluster sum of squares).
- After each iteration, the WCSS decreases and so we have a nonnegative monotonically decreasing sequence.
- This guarantees that the k-means **always converges**, but not necessarily to the global optimum.

Objective of K Means clustering

- Grouping similar data points: K-means aims to identify patterns in your data by grouping data points that share similar characteristics together. This allows you to discover underlying structures within the data.
- Minimizing within-cluster distance: The algorithm strives to make sure data points within a cluster are as close as possible to each other, as measured by a distance metric (usually Euclidean distance). This ensures tight-knit clusters with high cohesiveness.
- Maximizing between-cluster distance: Conversely, k-means also tries to maximize the separation between clusters. Ideally, data points from different clusters should be far apart, making the clusters distinct from each other.

Advantages and Limitations

- Advantages:
- Simplicity: K-means is easy to understand and implement.
- Scalability: It can handle large datasets efficiently.
- Speed: The algorithm is computationally efficient, especially with K-means++ initialization.

Advantages and Limitations

- Limitations:
- Predefined K: The number of clusters K must be specified in advance.
- Initial Centroid Sensitivity: The final clustering can depend on the initial placement of centroids, potentially leading to different results.
- Cluster Assumption: K-means assumes clusters are spherical and equally sized, which may not always be the case.
- Outliers: Sensitive to outliers and noise in the data.

How to find the Best K ?

- Choosing the right number of clusters is crucial for effective clustering. Several methods are used to determine the optimal K:
- Elbow Method: Plot the sum of squared errors (SSE) for different values of K. Look for an "elbow" point where the SSE reduction slows down significantly. This point often represents the optimal number of clusters.
- Silhouette Analysis: Calculate the silhouette score for different values of K. The silhouette score measures how similar each point is to its own cluster compared to other clusters. Choose the K with the highest average silhouette score.

Silhouette Score

- Silhouette Score
- The Silhouette Score is a metric used to evaluate the quality of clustering results. It measures how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The score ranges from -1 to 1, where:
- +1 indicates that the object is well clustered.
- 0 indicates that the object is on or very close to the decision boundary between two neighboring clusters.
- -1 indicates that the object is misclassified and assigned to the wrong cluster.

Silhouette Score

$$\text{Silhouette Score} = \frac{b-a}{\max(a,b)}$$

Where:

- a is the mean distance between the sample and all other points in the same cluster.
- b is the mean distance between the sample and all points in the nearest cluster (the cluster that minimizes this mean distance).

Silhouette Score

- **Steps to Compute Silhouette Score**
- **Calculate Cohesion (a)**: For each point iii in a cluster A, calculate the average distance to all other points in the same cluster. This is the intra-cluster distance.
- **Calculate Separation (b)**: For each point iii in a cluster A, calculate the average distance to all points in the nearest cluster B. This is the nearest-cluster distance.
- **Compute Silhouette Score for each point**: Use the formula given above.
- **Average Silhouette Score**: Calculate the average silhouette score for all points to get an overall assessment of clustering quality

Silhouette Score

- Data Points and Cluster Assignments:
- Cluster 1: [(2, 10), (2, 5), (5, 8)]
- Cluster 2: [(8, 4), (7, 5)]

Silhouette Score

- K Means clustering performs best data is well separated.
- When data points overlapped this clustering is not suitable.
- K Means is faster as compare to other clustering technique. It provides strong coupling between the data points.
- K Means cluster do not provide clear information regarding the quality of clusters.
- Different initial assignment of cluster centroid may lead to different clusters.
- Also, K Means algorithm is sensitive to noise. It may have stuck in local minima.

THANK - YOU