Unsupervised Machine Learning

Unsupervised Machine learning → unlabeled data

Goal: To discover patterns or relationships with the data, such as grouping similar instances together into clusters or finding low dimensional representations.

Example: Clustering, Dimensionality Reduction, Anomoly detection

Clustering:

- Making group of similar data points
- It is used to uncover the underlying structure of the data and is often used in fields such as pattern recognition, image analysis and market research.

Application:

a) For customer segmentation:

→ You can cluster your customer based on their purchase.

↓

useful for recommender system

b) For data analysis:

→ Perform analysis on each cluster.

c) Anamoly Detection

→ Any instance which has a low affinity to all cluster is likely to be anamoly detection

d) Dimensionality Reduction

e) For search engine

f) To segment an image

Approaches for Clustering

a) Agglomerative:-

First considers all points as individual clusters and then finds out the similarity betn two points, puts them into a cluster.

↓

Then, It goes on finding similar points & clusters until there is only one cluster left ie 'all points belong to big cluster' → bottom up approach

b) Divisive

- It is oposite of Agglomerative appioach

first consider all points as a cluster

↓ in subsequent steps

Find out the points/cluster which are least similar to each other.

↓

bigger cluster into smaller one $\xrightarrow{\text{Continve till}}$ cluster as many as datapoints

R

Mo Tu We Th Fr Sa Su

Memo No.

Date / /

K-Means Clustering

- Unsupervised Clustering algorithm
- Proposed by Stuart lloyd at 'Bell labs'

K mean is a clustering approach in which the data is grouped into K distinct non overlapping clusters based on their distances from the K-centres.

Theory:

1> let $C1, C2, C3 \ldots C_K$ be the K clusters

2> Then, we can write,

$C1 \cup C2 \cup \ldots C_K =$ each datapoints

3> Also,

$C_K \cap C_{K'} = \phi$ for all $K \neq K'$

This means, clusters are non overlapping.

3> The idea behind the K-Means clustering approach is that within cluster variation among the point should be minimum. The with-in-cluster variance is denoted by W(C_K). Hence, according to the statement above, we need to minimize this varience for all the clusters.

Mathematically,

$$\underset{C_1 \dots C_K}{\text{minimize}} \left\{ \sum_{K=1}^{K} W(C_K) \right\}$$

5> The next step is to define the criterion for measuring the within cluster varience. Generally, the criterion is Euclidean Distances between the datapoints.

$$W(C_K) = \frac{1}{|C_K|} \sum_{i,i' \in C_K} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

→ It calcubtes the mean of variance in each cluster.

So, ultimately, our goal is to minimize :-

$$\boxed{\underset{C_1 \ldots C_K}{\text{minimize}} \left\{ \sum_{K=1}^{K} \frac{1}{|C_K|} \cdot \sum_{i,i' \in C_K} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \right\}}$$

Algorithm:

1. Randomly assign K centres.
2. Calculate the distance of all the points from all the K centres and allocates the points to cluster based on the shortest distance. The model's inertia is the mean squared distance bet^n each instance & it's closest centroid. The goal is to have low inertia.
3. Once all points are assigned to cluster, recompute the centroids.

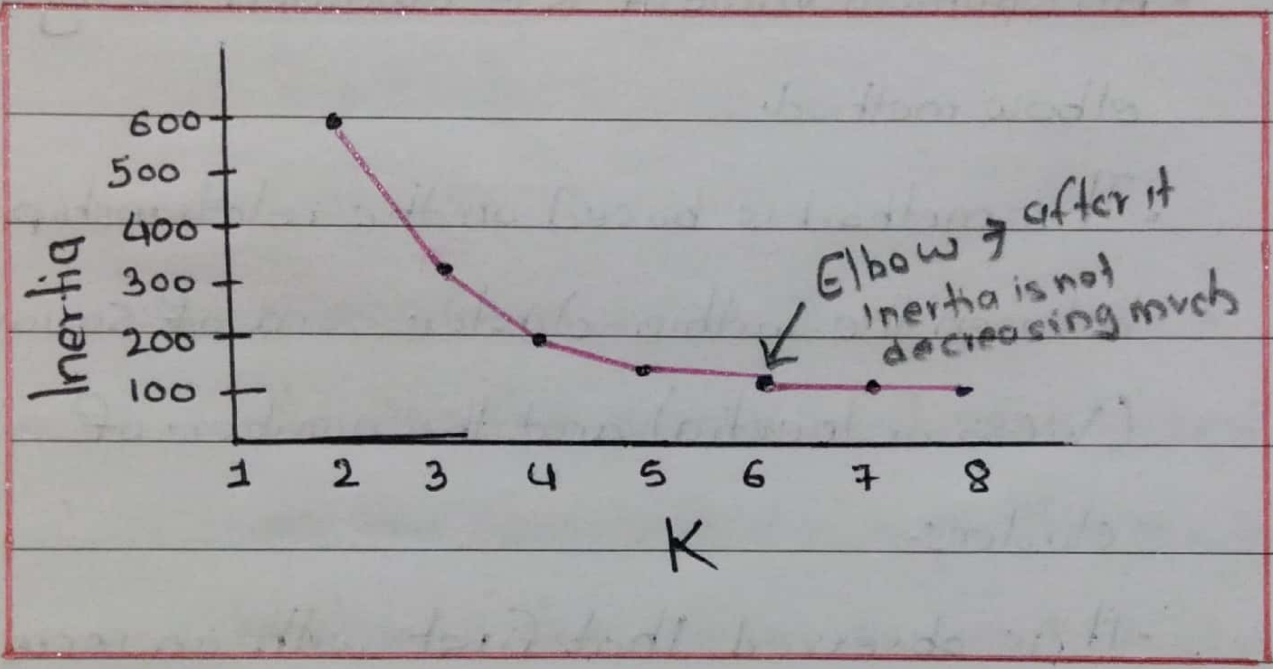
4. Repeat the steps 2 and 3 until the location of centroids stop changing and the cluster allocation of the points become constant.

Elbow Method:

- An optimum value of K is obtained using the elbow method.
- This method is based on the relationship between the within cluster sum of squared (WCSS or Inertia) and the number of clusters.
- It is observed that first with an increase in the number of clusters WCSS decreases steeply and then after a certain number of clusters, the drop in WCSS is not that prominent.
- The point after which the graph bet^{n}

WCSS and the number of clusters become comparatively become smother is termed as the elbow'.

- The number of cluster at the elbow point is optimum.



Challenges in K-Means:

① K-means do not behave well when cluster have varying shape, size, different densities or non spherical shape

② Need to specify the number of cluster before hand.

Improvement:

K-Means ++

- Modified version of K-Means

Steps:

1> Choose the first centroid randomly

2> For each datapoint, calculate the distance from the selected centroid.

3> Assign the probability to each data point proportional to the square of its distance from the selected centroid.

4> Choose the next centroid randomly, with a probability that the selecting points are faither away from the existing centroid.

5> Repeat 2-4, until all K centroids have been selected.

7) Assign data point to the closest centroid.

8) Recalculate the centroids as the mean of the data points in each cluster

9) Repeat 7-8 until centroid no longer move or maximum number of iteration is reached.

10) The final K-cluster, along with their centroids form the O/p.

Minibatch-KMeans

Instead of using full dataset at each iteration, the algorithm is cable of using minibatches moving centroid just slightly at each iteration.

- This speedup the algorithm by 3-4 times

```
from sklearn.cluster import MiniBatchKMeans
minibatch_kmeans = MiniBatchKMeans(n_clus=5)
minibatch_kmeans.fit(x)
```