What is Clustering?

Clustering gives insight on the data by grouping the data points into clusters

Goal of clustering is to divide the population or set of points into number of groups such that data points within each group are similar to each other whereas data points in different groups are different to each other

It is grouping things based on their similarity

Clustering is the classification of objects into subsets (clusters), so that the data in each share some common trait

It is unsupervised learning

Use of clustering

Clustering is used to segregate data

Example market basket analysis is used to perform customer segmentation by using clustering

Gives insight about purchase habits of customers

Also useful for recommendation systems like netflix. Clusters of users are made. Then when majority of people watch a particular type of movie then other people in the same cluster are recommended it.

Clustering can be used for anamoly detection.

Also useful for image compression

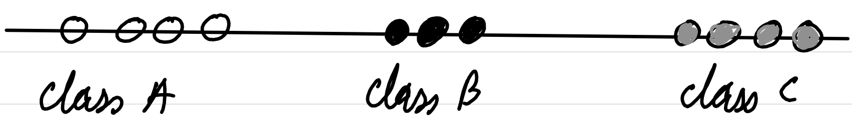
Social networks, linkedin recommendation system etc are based on ideas like clustering

K Means clustering

Suppose we have a data on a line

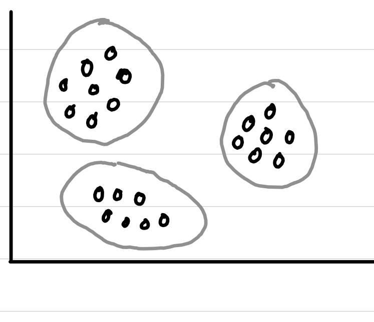
It is unlabelled data

We want to cluster it that means group it into parts like this



This clustering is done by K means clustering

Similarly for 2D



K means works on unlabelled data, that is unsupervised learning

The datapoints dont have any label with them, which means that the expected output is not present in trainny data

K means automatically identifies clusters based on the parameter values

K means is a very simple algorithm that works on distance between points & clusters

Its aim is to partition a set of objects into K clusters in such a way that the sum of squared distances between objects and their assigned cluster is minimized

The idea is that points which are close together will share common characteristics

The "closeness" can be in multiple dimentions

"Birds of same feather flock together"

Here, we want to split the data into 3 parts

So $k = 3$, k is the number of clusters

k means does not automatically find the value of k, unlike few algorithms of clustering that find number of clusters automatically and give no direct control over the number of clusters eg louvian community detection

Finding the optimal value of k requires calculation of clusters for every value of k & then chosing the most optimal value of k as per elbow plot

The advantage to providing the value of k, however is that you get control over the number of divisions you want to make.

k means clustering doesn't allow overlapping clusters

step 1 → select value of k
Here k = 3

step 2 → Select k random points.
These are our initial clusters
eg Here we select 3 data points

step 3 → Classify all points based on the k points

Classify using nearest neighbor to k.
That is measure the distance between a point and k initial clusters
Then classify the point as the cluster with the least distance

step 4 → Calculate mean of the clusters
mean = centroid

step 5 → Repeat step 3 by classifying points based on the Mean.
Repeat 3-5 till there is no update in classification

step 6 → Calculate the Variation between clusters

The cluster obtained may not be a good cluster. We can find that using the Variance of the cluster.

k-means algorithm cannot see the best clustering before the clustering is done. Hence, the thing that is done is to repeat the entire process of selecting random points and clustering over and over until the results get good.

step 7 → choose new random starting points and repeat the steps

Perform this for a number of iterations

step 8 → Select the clustering with the most Variance

① cluster the below data points

	Height	Weight	
1	185	72	
2	170	56	← Initial point C_1
3	168	60	
4	179	68	
5	182	72	← Initial point C_2
6	188	77	
7	180	71	$K=2$
8	180	70	
9	183	84	
10	180	88	
11	180	67	
12	177	76	

→

	Height	Weight	distance from C1	distance C2	Cluster
1	185	72	21.931	8	2
2	170	56	0	20	1
3	168	60	4.47	18.439	1
4	179	68	15	5	2
5	182	72	20	0	2
6	188	77	27.658	7.81	2
7	180	71	19.723	2.236	2
8	180	70	17.204	2.828	2
9	183	84	30.87	12.041	2
10	180	88	33.526	16.124	2
11	180	67	14.866	5.385	2
12	177	76	21.189	6.403	2

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

distance of point (185, 72) from centroid C1

$$d = \sqrt{(185-170)^2 + (72-56)^2}$$

$$= 21.931$$

New centroids → Mean of data points

$$\text{New } C_1 = \left(\frac{168+170}{2}, \frac{56+60}{2}\right) = (169, 58)$$

$$\text{New } C_2 = \left(\frac{183+179+182+188+180+180+183+180+180+177}{10}, \frac{72+68+72+77+71+70+84+88+67+76}{10}\right)$$

$$= (181.4, 74.5)$$

Now these centroids will be used in the next steps

The process keeps repeating until no new centroid is found

	Height	Weight	distance from C1 = (169, 58)	distance $C_2 = (181.4, 74.5)$	Cluster
1	185	72	21.26	4.38	2
2	170	56	2.23	21.73	1
3	168	60	2.23	19.74	1
4	179	68	14.14	6.92	2
5	182	72	19.10	2.57	2
6	188	77	26.87	7.03	2
7	180	71	17.02	3.76	2
8	180	70	16.27	4.7	2
9	183	84	29.52	9.63	2
10	180	88	31.95	13.57	2
11	180	67	14.21	7.629	2
12	177	76	19.69	4.648	2

Since clusters have not changed the final cluster is same

(Q2) Given list of points

A 2, 10
B 2, 5
C 8, 4
D 5, 8
E 7, 5
F 6, 4
G 1, 2
H 4, 9

Initial clusters A, D, G

Consider manhatten distance

→

Points	distance from initial clusters A	D	G	cluster
A	0	5	9	A
B	5	6	4	G
C	12	7	9	D
D	5	0	10	D
E	10	5	9	D
F	10	5	7	D
G	9	10	0	G
H	3	2	10	D

Cluster A → A
D → {C, D, E, F, H}
New centroid - {6, 6}

G → {B, G}
= {1.5, 3.5}

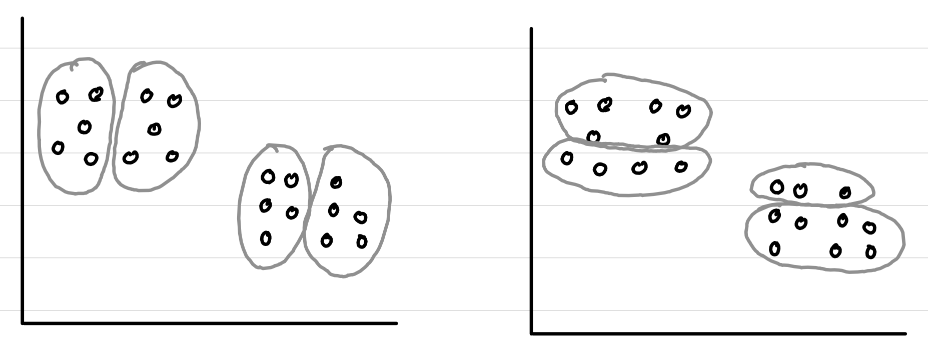
Modified clusters from new points

Points	New distances A	D	G	New cluster
A				
B				
C				
D				
E				
F				
G				
H				

And so on Repeat until the clusters do not change

Evaluation Metrics

The "variation" can be defined as
How good or how bad a cluster is



which is better?

We want the grouping in a way that

1. Minimizes distance in the group (Cohesion)
2. Maximizes distance amongst the groups. (Seperation)

High Cohesion & High seperation are ideal

Methods of cluster evaluation

1. Inertia → Intragroup distance
   Sum of distances of all points of a cluster from its centroid

2. WCSS → Within cluster sum of squares
   Sum of square of distances of all points of a cluster from its centroid

3. Dunn index → Takes into account both intra & inter distances

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

Dunn Index is the ratio of the minimum of all inter cluster distances and maximum of all intracluster distances

4. Silhouette score →

The silhoutte score & plot are used to evaluate the quality of a clustering solution

It is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (seperation)

Range is from -1 to +1

High value indicates that the object is well matched to its own cluster and poorly matched with its neighboring clusters

Used for measuring cluster quality for regular convex shaped clusters & not for irregular ones.

For a datapoint i in cluster C_i

Intra cluster distance

$$a(i) = \frac{1}{|C_i| - 1} \sum_{\substack{j \in C_i \\ i \neq j}} d(i,j)$$

Inter cluster distance

$$b(i) = \min_{j \neq i} \frac{1}{|C_j|} \sum_{j \in C_j} d(i,j)$$

ie smallest mean distance of i to all points in any other cluster

$|C_i|$ = number of points in cluster

Silhouette score for a datapoint i is

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \frac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases}$$

$$\therefore s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad \text{if } |C_i| > 1$$

$$s(i) = 0 \quad \text{if } |C_i| = 1$$

Silhoutte score close to 0 suggest overlapping clusters

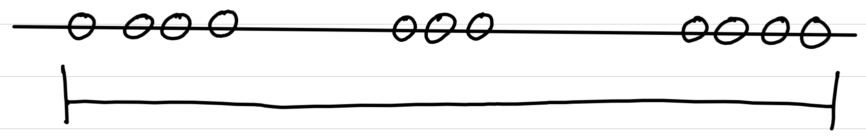
Negative score indicates poor clustering

(Elbow Plot)

Steps for selection of k

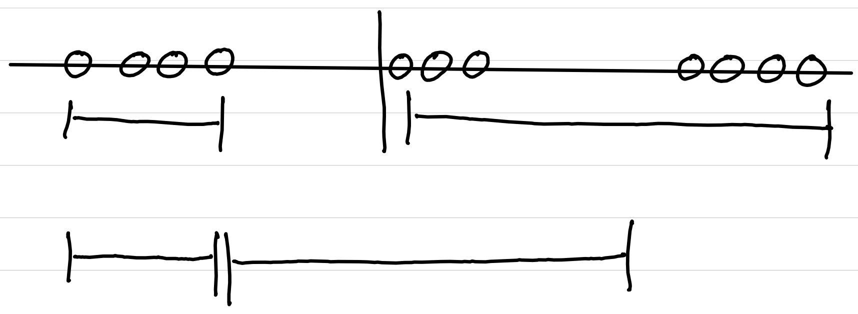
at k = 1, variation will be most

If we consider sum of square distances as "variation" or evaluation metric, then



large variance

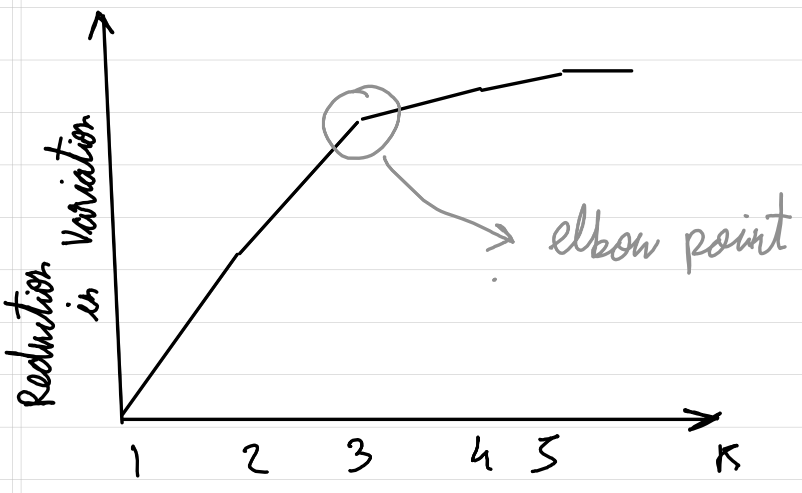
at k = 2, variation will be lesser



As value of k is increased, the variation will reduce

at k = N the variation will be 0 as there will be one point per cluster

Reduction in variation per value of k

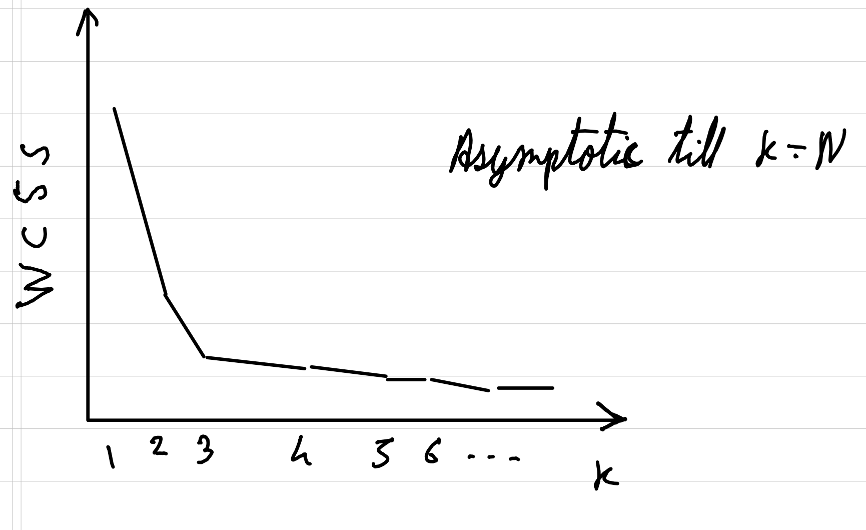


There is a huge reduction in the variation at k = 3

After 3, variation doesnt go down that quickly

This is an elbow plot & k = elbow in the plot.

If instead we plot WCSS vs k we get inverted graph



Mathematical Representation

K-Means aims to partition N observations into K-clusters in which each observation belongs to the cluster with nearest cluster centroid (mean)

In other words, it minimizes the within cluster dissimilarity using Objective function

$$O = \min_{C_1, C_2, C_3 \ldots C_K} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - \mu_i\|^2$$

cluster Mean

That is, for each cluster 1-K,

Take The sum of $\|x-\mu_i\|$ ie distance of point in the cluster from its centroid

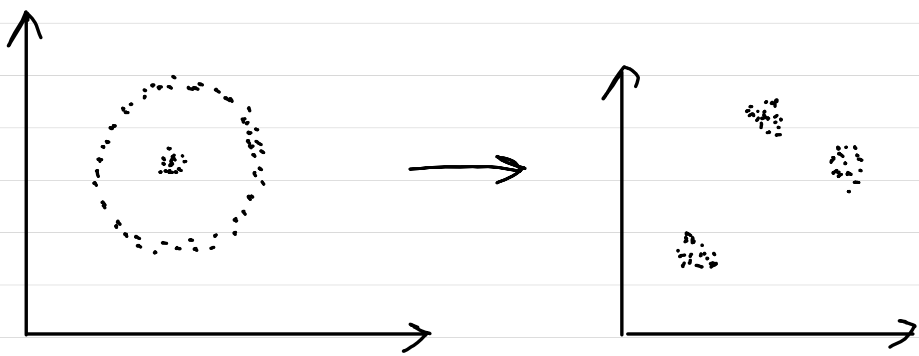
and take the total sum for minimization

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

No of points in cluster

Kernelized k-Means

Used to evaluate non linearly seperable data



The distance formula changes from

$$d(x_i, u_j) = \|x_i - u_j\|^2$$

$$= \left\| x_i - \frac{1}{|c_i|} \sum_{x \in c_i} x \right\|^2$$

to

$$d(x_i, u_j) = K(x, u_j) - \frac{2}{|C_j|} \sum_{x_p \in C_j} K(x_i, x_p)$$

$$+ \frac{1}{|C_j|^2} \sum_{x_q, x_p \in C_j} K(x_p, x_q)$$

K- Means Application

clustering applications

① Market segmentation
② Social Network Analysis
③ Image Segmentation

Other Applications →

① K Means can be used for identification of numbers

eg Classification of MNIST numbers. without even using the labels. accuracy of 90% can be found.

Images → 64 dimentional data

② Image compression

K -Means can be used for color compression

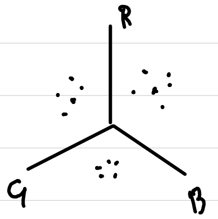
Consider an image with millions of colours.

Most Images will have large number of colours unused.

Many pixels may have similar or identical colours.

K-Means clustering used to group similar colors & reduce number of colors.- lossey compression

Colors are 3D datapoints (RGB)



Advantages → 1. Simple Model
2. scalable to large datasets

Disadvantages → 1. Dependes on initial values
2. Density variation gives trouble
3. Sensative to outliers
4. No overlapping clusters.
5. Only regular shaped clusters
6. No Nested clusters

The mean gives circular clusters. Other variations of k-means gives elliptical clusters

K-Means cannot handle complex data. it is a very simplistic limited Model

K-Means	K-NN
Unsupervised learning	Supervised learning
K = No of clusters	K = No of neighbours