23-Sep-2021

CS564 Mid Sem Assignment

Name: M.Maheeth Reddy
Roll: 180ICS31

Answers

Ans 1: Transforming data to one dimension using Fisher LDA:

For Class 1, $C_1 = \begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 3 & 3 \\ 4 & 5 \\ 5 & 5 \end{bmatrix}$

Class 2, $C_2 = \begin{bmatrix} 1 & 0 \\ 2 & 1 \\ 3 & 1 \\ 3 & 2 \\ 5 & 3 \\ 6 & 5 \end{bmatrix}$

Now, $\mu_1 = \frac{1}{|C_1|} \sum_{x_i \in C_1} x_i = \frac{1}{5} [15 \quad 18] = [3 \quad 3.6]$

$\mu_2 = \frac{1}{|C_2|} \sum_{x_i \in C_2} x_i = \frac{1}{6} [20 \quad 12] = [3.3 \quad 2]$

Scatter Matrix, $S = \sum_{j=1}^{n} (x_j - \bar{x})(x_j - \bar{x})^T$

[$\bar{x}$ is mean of x_j's]

After many calculations, we obtain

$$S_1 = \begin{bmatrix} 10 & 8 \\ 8 & 7.2 \end{bmatrix}$$

$$S_2 = \begin{bmatrix} 17.3 & 16 \\ 16 & 16 \end{bmatrix}$$

Now, $S_W = S_1 + S_2$

$$\Rightarrow S_W = \begin{bmatrix} 27.3 & 24 \\ 24 & 23.2 \end{bmatrix}$$

$$\Rightarrow S_W^{-1} = \begin{bmatrix} 0.40 & -0.42 \\ -0.42 & 0.48 \end{bmatrix}$$

The optimal line direction v is

$$v = S_W^{-1}(\mu_1 - \mu_2) = \begin{bmatrix} 0.40 & -0.42 \\ -0.42 & 0.48 \end{bmatrix} \begin{bmatrix} -0.3 \\ 1.6 \end{bmatrix}$$

$$v = \begin{bmatrix} -0.79 \\ 0.89 \end{bmatrix}$$

the projection of the initial 2D points on this line is $Y_1 = v^t c_1^t = [-0.79 \ 0.89] \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 3 & 5 & 5 \end{bmatrix}$

$$\Rightarrow \boxed{Y_1 = [0.99 \quad 1.09 \quad 0.3 \quad 1.14 \quad 0.5]}$$

$$Y_2 = v^t C_2^t = [-0.79 \quad 0.89] \begin{bmatrix} 1 & 2 & 3 & 3 & 5 & 6 \\ 0 & 1 & 1 & 2 & 3 & 5 \end{bmatrix}$$

$$\Rightarrow \boxed{Y_2 = [-0.79 \quad -0.69 \quad -1.48 \quad -0.59 \quad -1.28 \quad 0.29]}$$

Significance of Fisher linear discriminants for classification problems

Fisher Linear Discriminants tries to find the projection weight vector W such that,

$$J(W) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2} \text{ is maximum}$$

So, it is maximizing the ratio between the between-class variance to within-class variance. Quantitatively, it tries to find a large separation between the projected class means while also giving a variance within each class thereby minimizing the class overlap.

Using PCA for reducing dimensionality from two to one

Total means of the data = $\frac{1}{11}[35 \quad 30] = [3.18 \quad 2.72]$

Now, feature vectors are $(x_i - \mu)$ for all x_i in X (dataset)

Features = [[-2.18 -0.72],
[-1.18 -0.27],
[2.82 2.28]]

Now, we get the covariance matrix of the features

$$C(X) = \frac{1}{|X|} \sum_{f_i \in Features} (f_i - \mu_{f_i})(f_i - \mu_{f_i})^T$$

$$C(X) = \begin{bmatrix} 2.51 & 2.04 \\ 2.04 & 2.74 \end{bmatrix}$$

Now, we need to find the eigen values and eigen vectors of the equation $CY = \lambda Y$

eigenvalues : $|C - \lambda I| = 0$

Now, λ can take 0.58, 4.68. If we choose $\lambda = 4.68$ (the larger value), its eigenvector $\begin{bmatrix} 0.94 \\ 1 \end{bmatrix}$ will be the principal component.

On projecting the initial features on this vector, we have

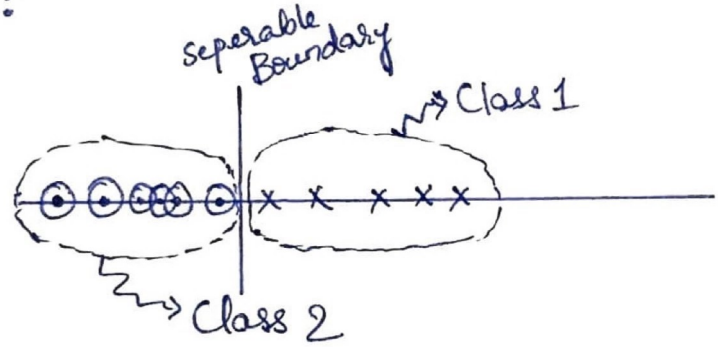
$$F_{n\times 2}\, Y_{2\times 1} = NF_{n\times 1} \quad (NF \rightarrow \text{new features})$$

The 1d features obtained after PCA are

[-2.79, -0.84, 0.10, 3.05, 4.0, -4.8, -2.8, -1.9, -0.9, 2,5]

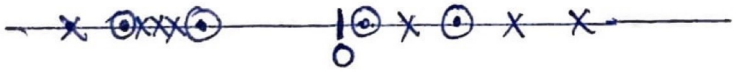
Classification Performance of Fisher LDA vs PCA on reduced dataset:

Fisher LDA:



LEGEND
X → Class 1
⊙ → Class 2

PCA:



If we observe the points in the reduced dimension produced by both Fisher LDA and PCA, clearly Fisher LDA is able to differentiate between the two classes in a better manner.

There is a mixup of class points due to PCA as dimensions are not reduced effectively.

Fisher LDA is better than PCA because :

Fisher LDA is a supervised learning technique with the aim of reducing dimensions ensuring minimum class separability. As per the working, LDA gives us the axes that account for the most variance between the individual classes whereas PCA accounts for the most variance in the whole dataset. Since LDA is more class-specific it performs better for classification as compared to PCA, by keeping original classes separated.

P.T.O

Ans 2: DBSCAN is an unsupervised learning algorithm that forms the clusters based upon the density of the data points or how the data is.

The parameters for DBSCAN are:

a) eps: It represents the radius of the neighbourhood around a data point.

b) minPts: Minimum number of data points that we want in the neighbourhood of a particular point to define a cluster.

Estimating the parameters:

minPts: As a rule of thumb, a minimum minPts can be derived from the number of dimensions 'D' in the dataset, as minPts $\geq D+1$. The low value of minPts $= 1$ doesn't make sense, because every point will be a core point. With minPts ≤ 2, the result will be the same as hierarchial clustering with single link metric, with the dendrogram cut at height 'ε'. Therefore, minPts ≥ 3. However, larger values are suitable for dataset with noise & will yield more significant clusters.

b) eps: If eps is too small, a large part of the data will not be clustered, whereas for a too high value of eps, clusters will merge and the majority of objects will be in the same cluster. In general, small values of eps are preferable.

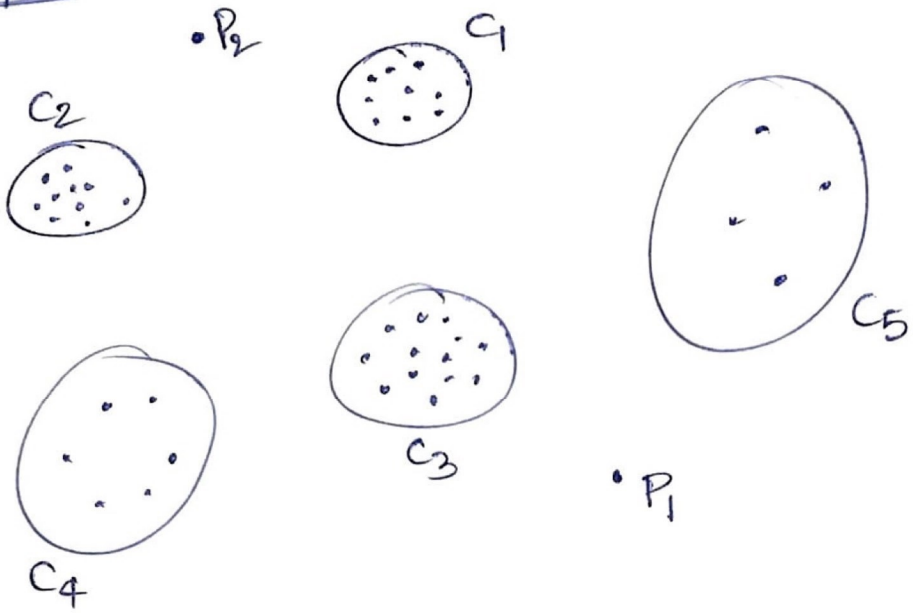
The value of eps can be chosen by using a K-distance graph, plotting distance to the K=minPts-1 nearest neighbour ordered from the largest to the smallest value. Good values of eps are where the plot shows an elbow. Alternatively, an OPTICS plot can be used to choose eps.

No, DBSCAN cannot find clusters with variable density and overlapping regions. This is due to the below reasons:

① If clusters have variable densities, DBSCAN cannot cluster well because the minPts-eps combination cannot be chosen appropriately for all clusters.

Consider the example in the next page,

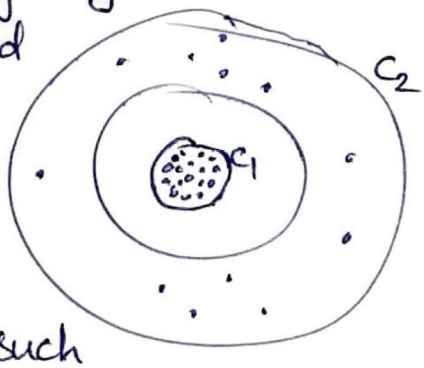
Example 1:



In this example, C_1, C_2, C_3 are very dense. C_4 is less dense region while C_5 is sparse region. P_1 and P_2 are outliers. As different datapoints are located in different density regions, it is impossible to obtain all the clusters simultaneously using one global density parameter. Because for C_1 we have to choose smaller eps, while for C_5 we need large eps.

(2) Consider the following example to see why DBSCAN struggles to find clusters in overlapping regions.

Example 2: Here, cluster C_1 is surrounded by another cluster C_2. C_1 has densely located points and C_2 has less densely located points. So, it is difficult to figure out optimal eps and minPts for such situations.



Ans 3: Given: Dataset with 10 companies' details like Charges filed (X_1), company size (X_2), status (Y)

We have to find the probability that a small company that has been charged is fraudulent, using Naive Bayes classification

So, required probability

$$= P(Y = \text{fraudulent} \mid X_1 = \text{yes} \cap X_2 = \text{small})$$

$$= \frac{P(Y = \text{fraudulent} \cap (X_1 = \text{yes} \cap X_2 = \text{small}))}{P(X_1 = \text{yes} \cap X_2 = \text{small})}$$

[Bayes theorem, $P(A|B) = \frac{P(A \cap B)}{P(B)}$]

$$= \frac{P(Y = \text{fraudulent} \cap X_1 = \text{yes} \cap X_2 = \text{small})}{P(X_1 = \text{yes} \cap X_2 = \text{small})}$$

$$= \frac{\frac{1}{10}}{\frac{2}{10}}$$

[only company 7 is having Y = fraudulent, X_1 = yes, X_2 = small and only companies 1, 7 have X_1 = yes and X_2 = small]

$$= \frac{1}{2} = 0.5$$

Hence, the probability that a small company charged with fraudulent financial reporting is fraudulent is 0.5

Ans 4 We have to use Genetic Algorithm to maximise $f(x) = x^2$ over $\{0, 1, 2, \ldots, 31\}$ with initial values of x as $\{13, 24, 8, 16\}$

String No.	x-value	Initial Population (given)	$f(x) = x^2$	pselect $\frac{f_i}{\Sigma f}$	Expected Count $\frac{f_i}{\bar{f}}$	Actual Count from Roulette Wheel
1	13	01101	169	0.16	0.63	1
2	24	11000	576	0.54	2.16	2
3	8	01000	64	0.06	0.24	0
4	16	10000	256	0.24	0.96	1

$\Sigma f = 169 + 576 + 64 + 256$

$\Rightarrow \boxed{\Sigma f = 1065}$

$\Rightarrow \boxed{\bar{f} = \left[\frac{\Sigma f}{4}\right] = 266}$ → rounded off

For the initial population

$\boxed{\begin{aligned} &\text{Sum, } \Sigma f = 1065 \\ &\text{Average, } \bar{f} = 266 \\ &\text{Max, } \max(f_i) = 576 \end{aligned}}$

For Mutation, we will replace the lowest number in the initial population i.e., 8 with the highest number i.e., 24.

For Crossover, string are randomly paired and mated Then we select crossing sites randomly to perform Crossover.

String No.	Mating Pool after reproduction	Mate (randomly selected)	Crossover Site (chosen randomly)	New Population	x-value	$f(x)=x^2$
1	0110|1	2	4	01100	12	144
2	1100|0	1	4	11001	25	625
3	11|000	4	2	11000	24	576
4	10|000	3	2	10000	16	256

$\sum f = 144+625+576+256$

$\Rightarrow \boxed{\sum f = 1601} \Rightarrow \boxed{\bar{f} = \frac{1601}{4} = 400}$

rounded off

After one crossover and mutation:

Sum is 1601
Average is 400
Max is 625

$\therefore f(x)=x^2$ has been maximized

Ans5 Need for Feature Selection:

① It enables the machine learning algorithm to learn faster.

② It reduces the complexity of a model and makes it easier to interpret.

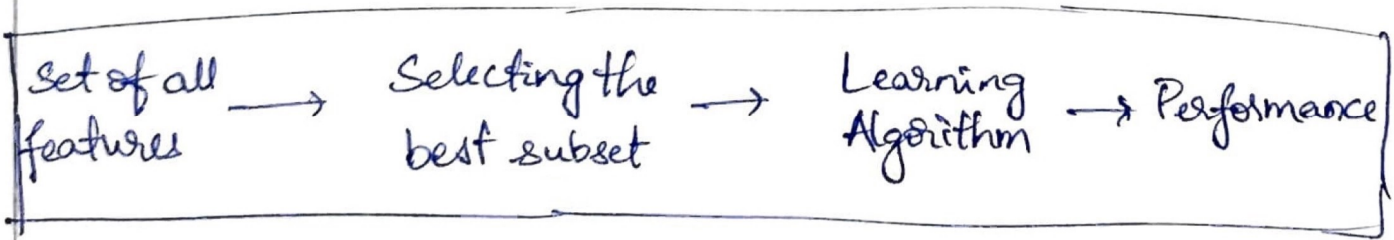
③ It improves the accuracy of a model if the right subset is chosen

④ It reduces overfitting

Feature Selection Strategies:

① Filter Methods

Filter Methods pick up the intrinsic properties of the features measured via univariate statistics instead of cross-validation performance. These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods

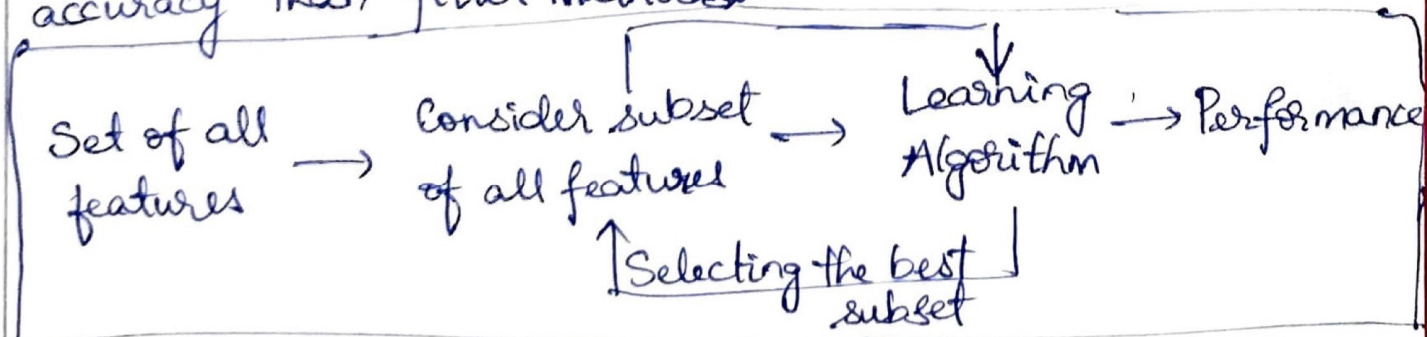


Some filter method techniques are:

① Information Gain
② Chi-Square Test
③ Fisher's Score
④ Correlation Coefficient
⑤ Variance Threshold
⑥ Mean Absolute Difference (MAD)
⑦ Dispersion Ratio

(II) Wrapper Methods

Wrappers require some method to search the space of all possible subsets of features, assessing their quality by learning and evaluating a classifier with that feature subset. The feature selection process is based on a specific machine learning algorithm that we are trying to fit on a given dataset. It follows a greedy search approach by evaluating all possible combinations of features against the evaluation criterion. The wrapper methods usually result in better predictive accuracy than filter methods.



Some Wrapper Method techniques are:

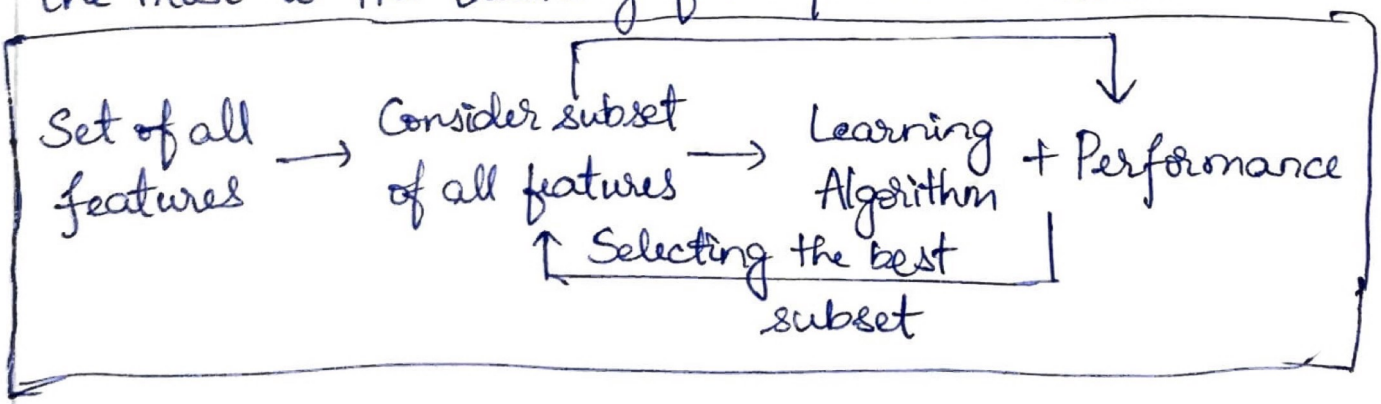
① Forward Selection ② Backward Elimination

③ Bi-directional elimination ④ Exhaustive Selection

⑤ Recursive Elimination

(III) Embedded Methods:

These methods encompass the benefits of both the Wrapper & Filter methods, by including interactions of features but also maintaining reasonable computational cost. Embedded Methods are iterative in the sense that takes care of each iteration of the model training process and carefully extracts those features which contribute the most to the training for a particular iteration.



Some Embedded Method Techniques are:

① Regularization ② Tree-based methods.

When we are given a huge data sample containing 2^{20} features, we prefer to use the Filter Methods over Wrapper or Embedded methods.

Reason:

Filter Methods are computationally less expensive than Wrapper or Embedded methods.

Wrapper methods search all possible subsets of features, it is not a good option for dealing with huge number of features.

Ans 6 K-medoid is more robust than K-means in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a K-means. However, there is a compromise in terms of complexity and efficiency as K-medoid algorithms are of $O(ik(n-k)^2)$ while K-means is of $O(ikn)$. A significant increase in complexity for better performance.

However, the given algorithm seems to solve the problem of complexities for performing local searches in place of universal searches done in the K-medoid algorithm. This might result in a local optima sometimes but increases the performance significantly based on our choice of subset size.

Cost Function:

While square mean error is a widely used cost function for its mathematical simplicity (smoothening of cost curve and easier gradient calculation), we will benefit by using a simple Euclidean Cost function because of the robustness it offers with outliers.

A squared error loss penalises a cluster center highly compared to Euclidean distance with outliers. This might result in a radical change of center location and might lead to miscalculations. Therefore, using a subtler cost function like a linear euclidean distance would not cause as much changes and therefore faster convergence.

Time Complexity:

The algorithm includes:

i) K-clusters

ii) i iterations to update the medoids for each clusters

iii) m*C comparisions for each iteration where m is the nearest neighbour size and C is the cluster size.

Therefore, Time Complexity is $O(k*i*m*C)$

P.T.O

Example: Consider the dataset

D: {(2,10), (2,5), (8,4), (5,8), (7,5), (6,4), (1,2), (4,9)}

K-means

Consider initial centroids as $C_1(2,5)$, $C_2(5,8)$, $C_3(7,5)$

Cost: $\|(X-C)\|^2$

X	Y	Distance to C_1	Distance to C_2	Distance to C_3
2	10	25	[13]	50
2	5	[–]	–	–
8	4	72	25	[2]
5	8	–	[–]	–
7	5	–	–	[–]
6	4	17	17	[2]
1	2	[10]	40	45
4	9	20	[2]	17

if a number is in box, then the point belongs to cluster having the centroid at that column

Updated Centroids: $C_1(1.5, 3.5)$, $C_2(3.67, 9)$, $C_3(7, 4.33)$

X	Y	Distance to C_1	Distance to C_2	Distance to C_3
2	10	42.5	3.79	57.149
2	5	2.5	18.79	25.449
8	4	42.5	43.75	1.1089
5	8	32.5	2.7689	17.469
7	5	32.5	27.09	0.449
6	4	20.5	30.429	1.109
1	2	2.5	56.129	41.429
4	9	36.5	0.109	30.809

We can see that centroids will remain the same hereafter

So, centroids are
$C_1(1.5, 3.5)$ $C_2(3.67, 9)$ $C_3(7, 4.33)$

Proposed Method:

Consider $C_1(2,5)$, $C_2(5,8)$, $C_3(7,5)$ as initial medoids

and m(nearest neighbours) = 1

cost: $\sum\sum d(x,m)$

X	Y	Distance to C_1	Distance to C_2	Distance to C_3
2	10	5	[3.605]	7.071
2	5	[=]	–	–
8	4	6.083	5	[1.414]
5	8	–	[=]	–
7	5	–	–	[=]
6	4	4.123	4.123	[1.414]
1	2	[3.1623]	7.21	6.708
4	9	4.47	[1.414]	5

Updation of Medoids:

Cluster 1:

Cost(2,5) → 3.1623

Cost(1,2) → 3.1623

∴ No Updates

Cluster 2:

Cost(5,8) → 5.019

Cost(4,9) → 3.65

∴ Medoid change to (4,9)

Cluster 3:

Cost(7,5) → 2.828

Cost(6,4) → 3.414

∴ No Update.

$C_1(2,5)$, $C_2(4,9)$ & $C_3(7,5)$ are the new medoids.

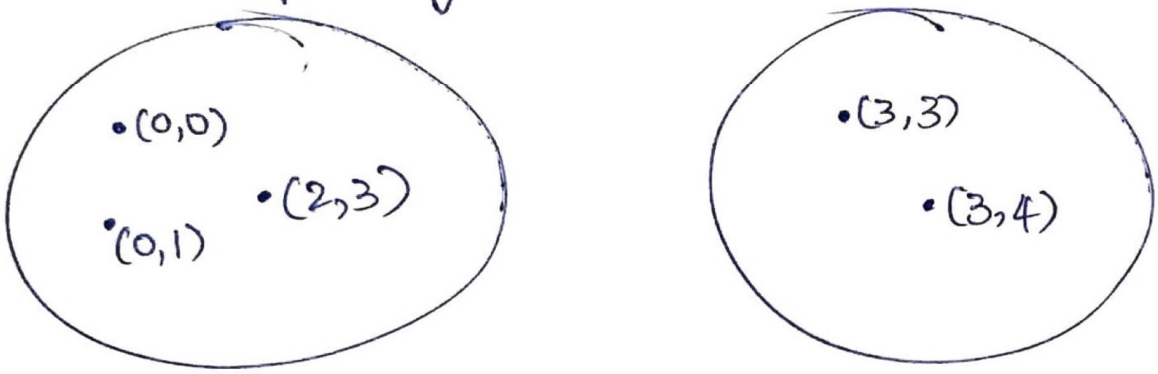
X	Y	Distance to C_1	Distance to C_2	Distance to C_3
2	10	5	2.236	7.071
2	5	—	—	—
8	4	6.083	6.403	1.414
5	8	4.24	1.414	3.605
7	5	—	—	—
6	4	4.123	5.38	1.414
1	2	3.1623	7.615	6.708
4	9	—	—	—

We can see that there is no cluster change, so no medoids will change. Hence, $C_1(2,5)$ $C_2(4,9)$ $C_3(7,5)$ are final medoids

So, both the methods result in different centroids/medoids.

Also, in K-means, the cost variation is higher and is easily affected by outliers.

Ans 7: We have the following clusters:



Cluster 1 Cluster 2

To compute the Silhouette for this clustering, we need to calculate the following metrics for each point P in each cluster:

$a(p)$ → average distance of point P to other points in its cluster (cohesion)

$b(p)$ → smallest average distance of point P to all points in any other cluster (separation)

$s(p)$ → $\frac{b(p) - a(p)}{\max(a(p), b(p))}$, silhouette value

Now, consider the point (0,0) in Cluster 1:

$$a((0,0)) = \frac{\text{Distance to } (0,1) + \text{Distance to } (2,3)}{2}$$

$$= \frac{(|0-0|+|0-1|)+(|0-2|+|0-3|)}{2} = \frac{1+5}{2} = 3$$

So, $\boxed{a((0,0)) = 3}$.

Since there are only 2 clusters here, b(p) will just be average distance of point p to all points in cluster 2.

$$b((0,0)) = \frac{\text{Distance to } (3,3) + \text{Distance to } (3,4)}{2} = \frac{6+7}{2}$$

$$\Rightarrow \boxed{b((0,0)) = 6.5}$$

$$\text{So, } s((0,0)) = \frac{b((0,0)) - a((0,0))}{\max(a(0,0), b(0,0))} = \frac{6.5-3}{\max(3, 6.5)} = \frac{3.5}{6.5} = \frac{7}{13}$$

$$\Rightarrow \boxed{s((0,0)) \approx 0.538}$$

Performing the calculations above for all points, we can write the results in a table as shown below:

	p	a(p)	b(p)	s(p)
Cluster 1	0,0	$(1+5)/2 = 3$	$\frac{(6+7)}{2} = 6.5$	$\frac{(6.5-3)}{6.5} = 0.538$
	0,1	$\frac{(1+4)}{2} = 2.5$	$\frac{5+6}{2} = 5.5$	$\frac{5.5-2.5}{5.5} = 0.545$
	2,3	$\frac{5+4}{2} = 4.5$	$\frac{1+2}{2} = 1.5$	$\frac{1.5-4.5}{4.5} = -0.667$
Cluster 2	3,3	$\frac{1}{1} = 1$	$\frac{6+5+1}{3} = 4$	$\frac{4-1}{4} = 0.75$
	3,4	$\frac{1}{1} = 1$	$\frac{7+6+2}{3} = 5$	$\frac{5-1}{5} = 0.8$

Interpreting the results

Observation:

Cluster 2 is a good cluster

Cluster 1 is not a good cluster

Reason:

In Cluster 1, point (2,3) has negative silhouette value of −0.667. This is because the point (2,3) is closer to Cluster2 but assigned to Cluster 1.

The remaining 4 points are assigned properly to their respective clusters, this is evident from their positive silhouette values.

Ans 8: Time Complexity of Single Linkage HCA:

In single-linkage HCA, at each step, we merge two clusters whose two closest members have the smallest distance.

The time complexity of single linkage HCA is $O(n^2)$.

We first compute all distances in $O(n^2)$. While doing this we also find the smallest distance for each data point and keep them in a next-best-merge array. In each of the $(n-1)$ merging steps, we then find the smallest distance in the next-best-merge-array. We merge the two identified clusters and update the distance matrix in $O(n)$. Finally, we update the next-best-merge array in $O(n)$ in each step. We can do the latter in $O(n)$ because if the best merge partner for k before merging i and j was either i or j, then after merging, the best merge partner for k is the merger of i and j.

Time Complexity of Complete Linkage HCA:

In this linkage HCA, at each step we merge those two clusters whose merger has the smallest diameter.

The time complexity for this algorithm is $O(n^2 \log n)$.

It takes $O(n^2 \log n)$ time to compute distance metric (n^2) and then sort the distances for each data point. After each merge iteration, the distance metric can be updated in $O(n)$. We pick the next pair to merge by finding the smallest distance that is still eligible for merging. If we do this by traversing the n sorted lists of distances, then, by the end of clustering, we would have done n^2 traversal steps. Adding all this up gives $O(n^2 \log n)$

Time Complexity of Average Linkage HCA:

The average linkage clustering calculates the distance between clusters in hierarchial cluster analysis. The linkage function specifying the distance between two clusters is completed as the average distance between objects from the first cluster and objects from the second cluster.

The time-complexity of average-link clustering is $O(n^2 \log n)$

We first compute all n^2 similarities for the singleton clusters and sort them for each cluster. This takes $O(n^2 \log n)$ time. In each of the $O(n)$ merge iterations, we identify the pair of clusters with the highest cohesion in $O(n)$, merge the pair and update cluster centroids, gammas, and cohesions of the $O(n)$ possible mergers of the just created cluster with the remaining clusters. For each cluster, we also update the sorted list of merge candidates by deleting the two just merged clusters and inserting its cohesion with the just created cluster. Each iteration thus takes $O(n \log n)$ time. Hence, $O(n^2 \log n)$ is overall time complexity.

Ans 9 Given a simple linear regression model,

$$Y = \beta_0 + \beta_1 X + \epsilon$$

where Y is the dependent variable,

X is the independent or explanatory variable,

β_0, β_1 are parameters of the model, regression coefficients (intercept and slope terms respectively)

ϵ is the unobservable error component that accounts for failure if data lies on straight line.

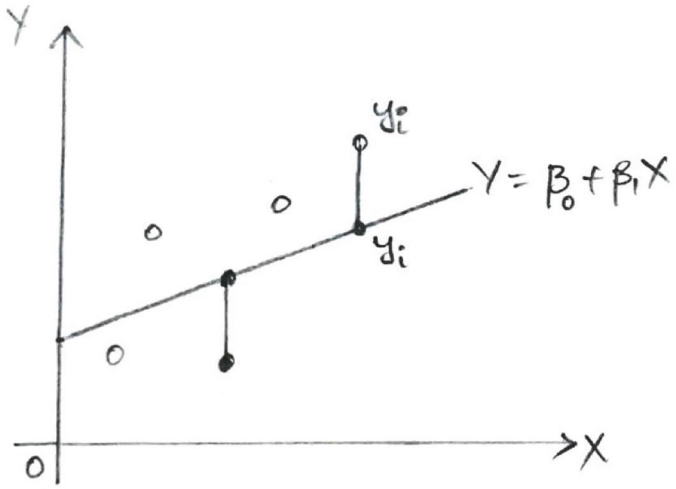
We need to determine β_0 & β_1 to dermine the statistical model $Y = \beta_0 + \beta_1 X + \epsilon$. For this, n pair of observations (x_i, y_i) $[i = 1 \text{ to } n]$ on (X, Y) are collected

For estimating the values of the parameters, various methods are available out of which least squares method is the most popular.

Least Square Estimation

The principle of least squares estimates the parameters by minimizing the sum of the difference between observations and the line in the scatter diagram. In the direct regression method, the vertical difference between the observations and the line is taken for minimization, to obtain the parameters β_0 and β_1. Here β_0 and β_1 are a part of the following transformation.

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i \quad \text{for } i = 1 \text{ to } n$$



Direct Regression

Minimizing the sum of squares of error, w.r.t β_0 and β_1

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

Partial Derivative of $S(\beta_0, \beta_1)$ w.r.t β_0 is

$$\frac{\partial S}{\partial \beta_0} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)$$

Partial Derivative of $S(\beta_0, \beta_1)$ w.r.t β_1 is

$$\frac{\partial S}{\partial \beta_1} = -2 \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i) x_i$$

We find solutions for $\beta_0 \& \beta_1$ by $\frac{\partial S}{\partial \beta_0} = 0$ and $\frac{\partial S}{\partial \beta_1} = 0$

This gives the ordinary least square estimates b_0, b_1 for β_0, β_1 respectively as:

$$\boxed{b_0 = \bar{y} - b_1 \bar{x}} \qquad \boxed{b_1 = \frac{S_{xy}}{S_{xx}}}$$

where, $S_{xy} = \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$

$S_{xx} = \sum_{i=1}^{n} (x_i - \bar{x})^2$

$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n}$

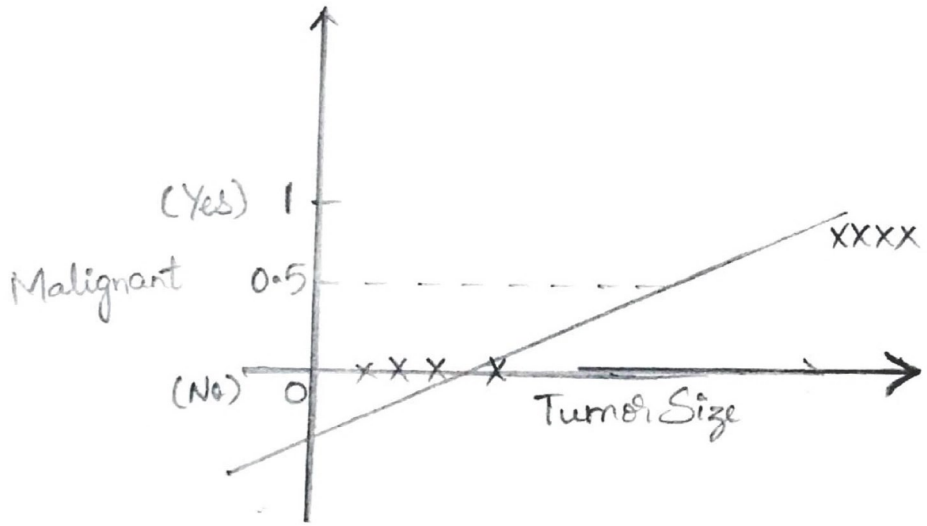
$\bar{y} = \frac{\sum_{i=1}^{n} y_i}{n}$

Also, we can verify that $S(\beta_0, \beta_1)$ has global minimum at (b_0, b_1)

Now, the fitted linear regression model is $y = b_0 + b_1 x$ and $\hat{y} = b_0 + b_1 x_i$ (for $i = 1$ to n) are the predicted values.

Converting Linear Regression Model to Logistic Regression Model

In the problems involving prediction of a discrete set of values called classes, we will need to model the probability of occurence of the class w.r.t input.



If we use Linear Regression for classifying tumors in the above graph we set the threshold to 0.5 and we are using the decision mapping function $h(x) = \beta_0 + \beta_1 x$

Now, our problem is: $h(x)$ can be greater than one or less than zero, but we want output as only two discrete values 0 or 1.

So, the desired characteristic feature of the hypothesis function is $0 \leq h_\theta(x) \leq 1$

Our task is: Given a binary output variable Y, we want to model the conditional probability $P(Y|X)$ as a function of x.

The desired features of such a function $p(x)$ are:

i) It should linear function of x

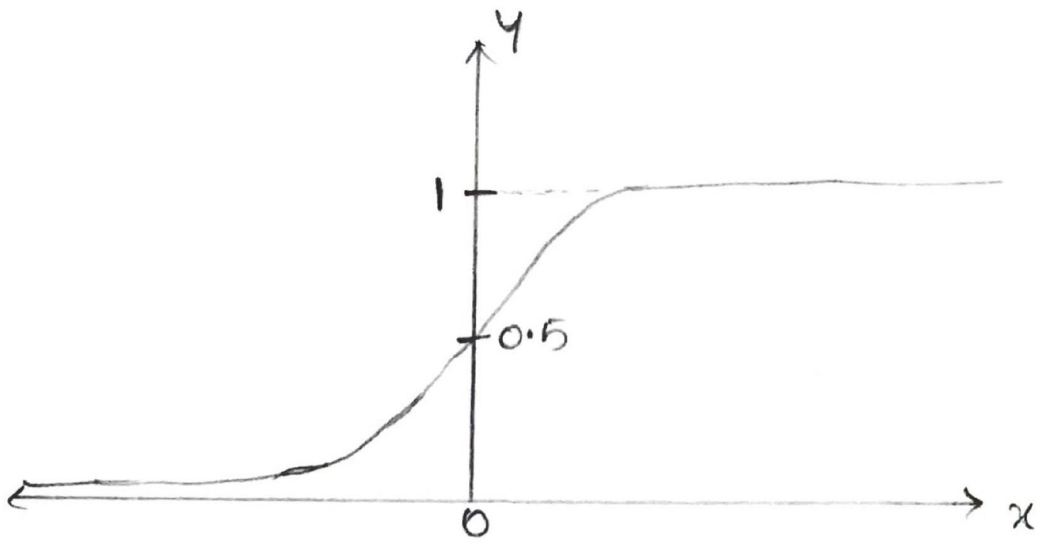
ii) $\log(p(x))$ should be a linear function of x, so that changing the input variable multiplies the probability by a fixed amount.

iii) $\log p$ has an unbound range in the logit transformation, $\log\left(\frac{p}{1-p}\right)$

Finally, we have

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 x$$

$$\Rightarrow \boxed{p(x; \beta_0, \beta_1) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}}$$



Hence, $p(x)$ is the estimated probability that $y=1$ on input x.

Hence, the logistic function for finding the probability of Y given X is as,

$$P(Y|X) = P(x; \beta_0, \beta_1) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$$

This is the basis of logistic regression

Finding linear regression equation for the dataset below:

x	2	4	6	8
y	3	7	5	10

$y = a + bx$, $a = \frac{\Sigma y \Sigma x^2 - \Sigma x \Sigma xy}{n(\Sigma x^2) - (\Sigma x)^2}$ $b = \frac{n\Sigma xy - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2}$

x	y	x^2	xy
2	3	4	6
4	7	16	28
6	5	36	30
8	10	64	80

$\Sigma x = 20$ $\Sigma y = 25$ $\Sigma x^2 = 120$

$\Sigma xy = 144$

Now, $a = \frac{25 \times 120 - 20 \times 144}{4 \times 120 - 400}$

$\Rightarrow a = 1.5$

$b = \frac{4 \times 144 - 20 \times 25}{4 \times 120 - 20^2} = 0.95$

$\Rightarrow b = 0.95$

So, the linear regression equation is

$$y = 1.5 + 0.95x$$

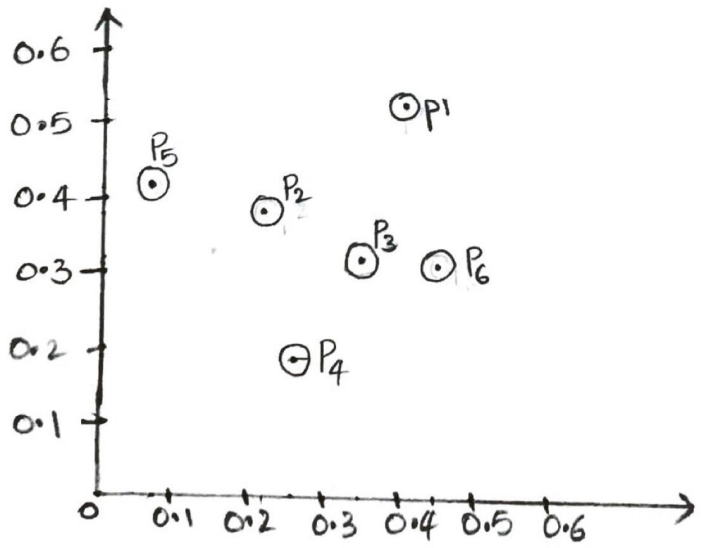
Ans 10

Given database D:

points	x	y
P1	0.40	0.53
P2	0.22	0.38
P3	0.35	0.32
P4	0.26	0.19
P5	0.08	0.41
P6	0.45	0.31

Performing Single Link Clustering:

Step-1: Plot the points in 2D space.



We will use this plot to draw the clusters later.

Step-2: Compute the distance matrix by calculating the euclidean distance between every pair of points in database

euclidean distance for points $P_1, P_2 = d(P_1, P_2)$

$$= \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2} = \sqrt{(0.40-0.22)^2+(0.53-0.38)^2}$$

$$= \sqrt{0.0549} = 0.2343$$

So, distance matrix is

P_1	0	–	–	–	–	–
P_2	0.24	0	–	–	–	–
P_3	0.22	0.15	0	–	–	–
P_4	0.37	0.20	0.15	0	–	–
P_5	0.34	0.14	0.28	0.29	0	–
P_6	0.23	0.25	(0.11)	0.22	0.39	0
	P_1	P_2	P_3	P_4	P_5	P_6

Step-3: Identify two clusters with shortest distance in the distance matrix and merge them together. Recompute the distance matrix, until all the points cluster into a single cluster.

P_3 and P_6 are the closest clusters ($d(P_3, P_6) = 0.11$, least in distance matrix). Updated distance matrix is:

P_1	0				
P_2	0.24	0			
(P_3, P_6)	0.22	0.15	0		
P_4	0.37	0.20	0.15	0	
P_5	0.34	(0.14)	0.28	0.29	0
	P_1	P_2	(P_3, P_6)	P_4	P_5

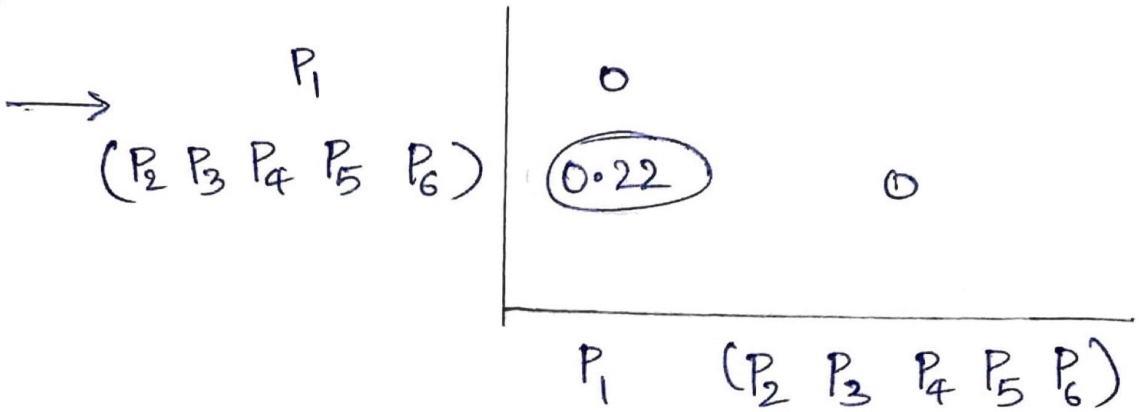
Now P_2, P_5 are closest clusters. After merging them we get,

P_1	0			
(P_2, P_5)	0.24	0		
(P_3, P_6)	0.22	(0.15)	0	
P_4	0.37	0.20	0.15	0
	P_1	(P_2, P_5)	(P_3, P_6)	P_4

Clusters (P_3, P_6), (P_2, P_5) are closest. after merging them.

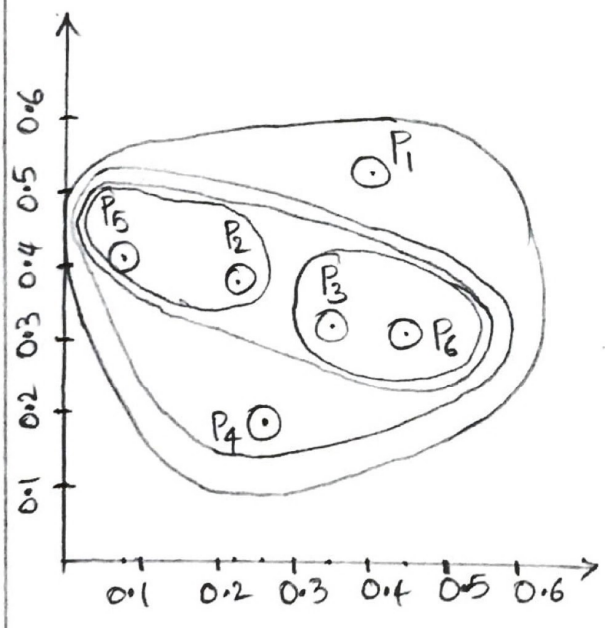
P_1	0		
(P_2, P_3, P_5, P_6)	0.22	0	
P_4	0.37	(0.15)	0
	P_1	(P_2, P_3, P_5, P_6)	P_4

Clusters P_4 , (P_2, P_3, P_5, P_6) are the closest. After merging, we get:

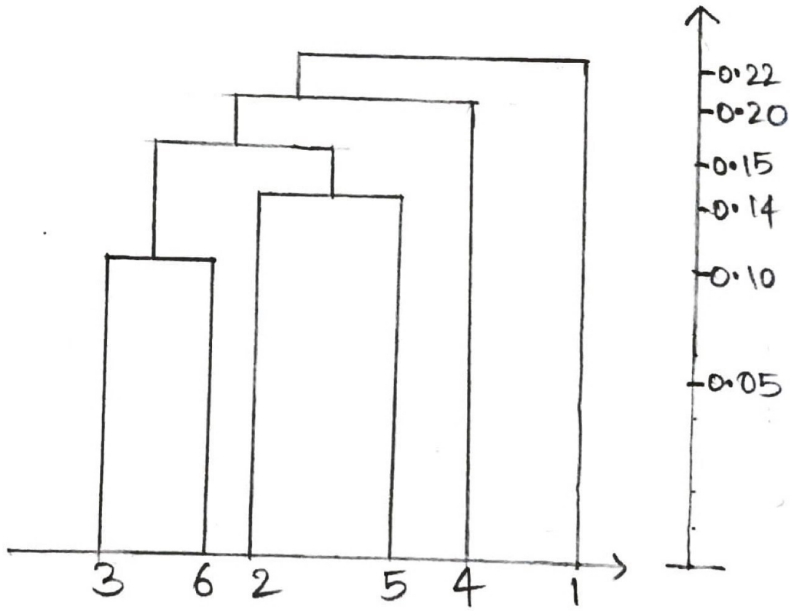


Now, all points have been clustered. See the following representations of clusters:

Clusters



Dendogram



Ans 11 Given dataset,

X	1	3	2.5	1.5	3	2.8	2.5	1.2	1	1	1	5	4
Y	2	4	4	2.5	5	4.5	4.5	2.5	3	5	2.5	6	3
	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}

Take $P_1(1,2)$ and calculate distances to all other points from P_1

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}	P_{11}	P_{12}	P_{13}
Distances from P_1	0	2.8	2.5	0.7	3.6	3.06	2.9	0.53	1	3	0.5	5.6	3.1

As eps = 0.6, for P_1 only two points P_8 and P_{11} are in the neighbourhood $dist(P_1, P_8) = 0.53 < eps$

$dist(P_1, P_{11}) = 0.5 < eps$

We summarize and find neighbours for all points:

(points and neighbours)

P_1	P_8, P_{11}
P_2	P_3, P_6
P_3	P_2, P_6, P_7
P_4	P_8, P_{11}
P_5	P_6
P_6	P_2, P_3, P_5, P_7

points	neigbours
P_7	P_3, P_6
P_8	P_1, P_4, P_9, P_{11}
P_9	P_8, P_{11}
P_{10}	<none>
P_{11}	P_1, P_4, P_8, P_9
P_{12}	<none>
P_{13}	<none>

So, the core points are
P_6, P_8, P_{11}

border points are
$P_1, P_2, P_3, P_4, P_5, P_7, P_9$

outliers are
P_{10}, P_{12}, P_{13}

Now, every core point will be assigned to a new cluster unless some of the core points share neighbourhood points, they will be included in same cluster. Every border point is assigned to cluster based on the core point in its neighbourhood.

Clusters:

Cluster-1: $P_2(3,4)$, $P_3(2.5, 4)$, $P_5(3,5)$, $P_6(2.8, 4.5)$, $P_7(3.5, 4.5)$

Cluster-2: $P_1(1,2)$, $P_4(1.5, 2.5)$, $P_8(1.2, 2.5)$, $P_9(1,3)$, $P_{11}(1, 2.5)$

Outliers (Noise): $P_{10}(1,5)$, $P_{12}(5,6)$, $P_{13}(4,3)$

Terminologies related to DBSCAN

(1.) **Direct density reachable:** A point is called direct density reachable if it has a core point in its neighbourhood.

Eg: $P_1(1,2)$, $P_8(1.2, 2.5)$ in the given problem

(2.) **Density Reachable:** If a point is connected through a series of core points

Eg: $P_9(1,3)$ and $P_4(1.5, 2.5)$ are connected through a core point $P_8(1.2, 2.5)$

(3.) **Density Connected:** Two points are called density connected if there is a core point which is density reachable from both the points.

Ans 12 Expectation – Maximization Algorithm

This algorithm is used for finding maximum likelihood estimates of parameters in stochastic models, where the model depends on unobserved latent or hidden variables. EM iterates between performing expectation of likelihood of all model parameters by including the hidden variables as if they were observed. Each maximization step involves the computation of the maximum likelihood estimates of the parameters by maximizing the expected likelihood found during the expectation step. The parameters produced by the maximization step are then used to begin another expectation step, and the process is repeated.

EM Clustering Algorithm : EM is an iterative method which alternates between two steps, expectation(E) and maximization(M). For clustering, EM makes use of the finite Gaussian mixture model and estimates a set of parameters iteratively until a desired convergence value is achieved.

The mixture is defined as a set of k probability distributions and each distribution corresponds to one cluster. An instance is assigned with a membership probability for each cluster.

The EM works as follows:

① Guess initial parameters: Mean and Standard Deviation (if normal distribution)

② Iteratively refine the parameters with E and M steps.

In the E step: compute the membership possibility for each instance based on the initial parameter values.

In the M step: Recompute the parameters based on the new membership possibilities.

③ Assign each instance to the cluster with which it has the highest membership possibility.

$X = \{1, 2, 3, 10, 11, 12\}$

Let $\mu_1 = 0.6$, $\mu_2 = 0.4$ (random assumption)

$\sigma_1 = 0.82$, $\sigma_2 = 0.82$

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \quad \text{——— (1)}$$

Then for two clusters say A(a) and B(b)

$$P(x_i|a) = \frac{1}{\sqrt{2\pi\sigma_a^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_a}{\sigma_a}\right)^2} \quad \text{——(2)}$$

$$P(x_i|b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} e^{-\frac{1}{2}\left(\frac{x-\mu_b}{\sigma_b}\right)^2} \quad \text{——(3)}$$

$$a_i = P(a|x_i) = \frac{P(x_i|a)P(a)}{P(x_i|a)P(a) + P(x_i|b)P(b)} \quad \text{——(4)}$$

$$b_i = P(b|x_i) = \frac{P(x_i|b)P(b)}{P(x_i|a)P(a) + P(x_i|b)P(b)} \quad \text{——(5)}$$

These are the probability of a point i

OR $$b_i = P(a|x_i) = 1 - a_i \quad \text{——(6)}$$

Updation

New mean formula

$$\mu_a = \frac{a_1x_1 + a_2x_2 + a_3x_3 + \cdots + a_nx_n}{a_1 + a_2 + \cdots + a_n} \quad \text{——(7)}$$

$$\mu_b = \frac{b_1x_1 + b_2x_2 + \cdots + b_nx_n}{b_1 + b_2 + \cdots + b_n} \quad \text{——(8)}$$

New Standard Deviation Formula

$$\sigma_a^2 = \frac{a_1(x_1-\mu_1)^2 + \ldots + a_n(x_n-\mu_n)^2}{a_1+a_2+\ldots+a_n} \quad \text{—— (9)}$$

$$\sigma_b^2 = \frac{b_1(x_1-\mu_1)^2 + \ldots + b_n(x_n-\mu_n)^2}{b_1+b_2+\ldots+b_n} \quad \text{—— (10)}$$

To estimate priors,

$$P(b) = \frac{b_1+b_2+\ldots+b_n}{n} \qquad P(a) = 1-P(b)$$

1st iteration

Prior probability

Assuming $P(a) = P(b) = 0.5$, the 2 clusters are both equally probable

$P(x_1|a) = 0.432$ $P(x_2|a) = 0.113$ $P(x_{x=3}|a) = 6.71\times10^{-3}$

$P(x|a) = 1.42\times10^{-29}$ $P(x_{x=11}|a) = 5.72\times10^{-36}$
$x=10$

$P(x_{x=12}|a) = 5.21\times10^{-43}$

$P(x_1|b) = 0.372$ $P(x_2|b) = 0.072$ $P(x_{x=3}|b) = 3.19\times10^{-3}$

$P(x_{x=10}|b) = 8.4\times10^{-31}$ $P(x_{x=11}|b) = 2.5\times10^{-37}$ $P(x_{x=12}|b) = 1.7\times10^{-44}$

$P(a) = P(b) = 0.5$

$$b_1 = \frac{0.432 \times 0.5}{0.432 \times 0.5 + 0.372 \times 0.5} = 0.537 \times 0.463$$

$b_2 = 0.610 \times 0.39$ $\quad$ $b_3 = 0.677 \times 0.323$

$b_{x=10} = 0.944 \times 0.056$ $\quad$ $b_{x=11} = 0.957 \times 0.043$

$b_{x=12} = 0.969 \times 0.032$

New prior probability of 'a'

$P(a) = 0.7821$ $\quad$ $P(b) = 0.217$

New $\mu_a = 7.53$ $\quad$ New $\mu_b = 0.381$

$$\text{New } \sigma_a^2 = \frac{3.07 + 3.42 + 13.89 + 5.76 + 11.52 + 19.34}{4.693}$$

$$= 12.14$$

$$\Rightarrow \sigma_a = \sqrt{12.14} = 3.48$$

$$\text{New } \sigma_b^2 = \frac{0.28 + 0.63 + 0.84 + 0.53 + 0.45 + 0.37}{1.307}$$

$$= 2.37$$

$$\Rightarrow \sigma_b = \sqrt{2.37} = 1.54$$

Iteration-2

$\sigma_a = 3.98, \quad \mu_a = 7.53$

$\sigma_b = 1.54, \quad \mu_b = 0.381$

$P(x_1|a) = 0.019 \quad P(x_2|a) = 0.032 \quad P(x_3|a) = 0.049$

$P(x_{10}|a) = 0.089 \quad P(x_{11}|a) = 0.069 \quad P(x_{12}|a) = 0.05$

$P(a) = 0.7821 \qquad P(b) = 0.217$

$P(x_1|b) = 0.07 \quad P(x_2|b) = 3.1 \times 10^{-5}$

$P(x_3|b) = 1.42 \times 10^{-11} \quad P(x_{10}|b) = 0$

$P(x_{11}|b) = 0 \qquad P(x_{12}|b) = 0$

$a_1 = 0.493 \quad a_2 = 0.99 \quad a_3 = 0.99$

$a_{10} = 1 \quad a_{11} = 1 \quad a_{12} = 1$

$b_1 = 0.507 \quad b_2 = 0.01 \quad b_3 = 0.01$

$b_{10} = 0 \quad b_{11} = 0 \quad b_{12} = 0$

New $\mu_a = 6.70 \quad \mu_b = 0.087$

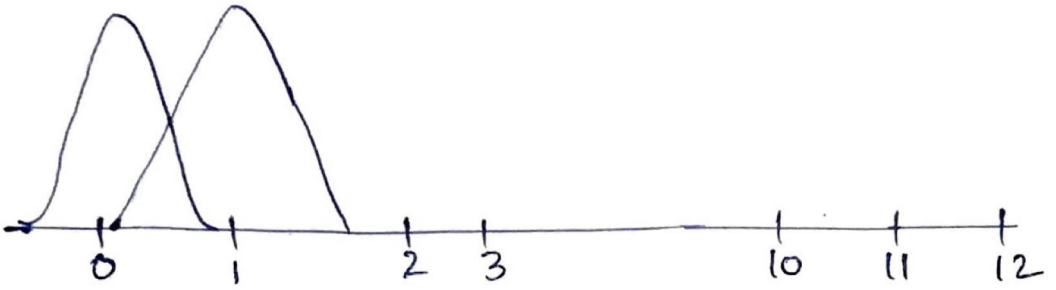
$\sigma_a^2 = 19.89 \implies \sigma_a = 4.45$

$\sigma_b^2 = 1.021 \implies \sigma_b = 1.01$

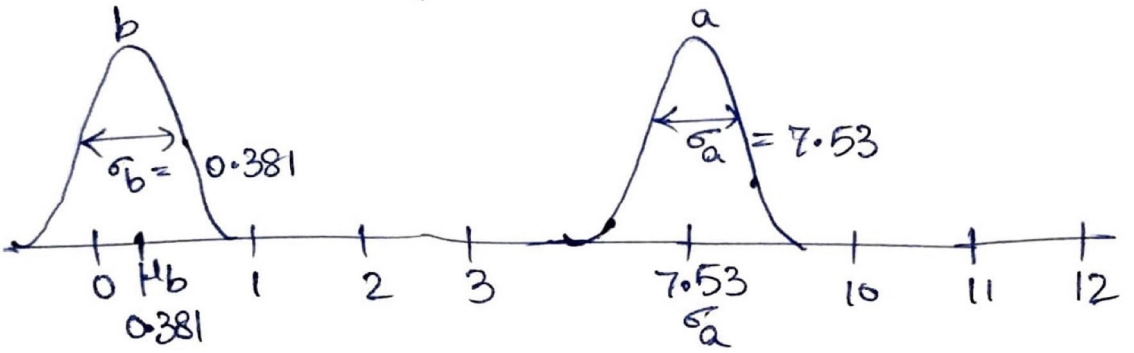
Visualization

Initially

$\mu_1 = 0.6$
$\mu_2 = 0.4$
$\sigma_1 = 0.82$
$\sigma_2 = 0.82$

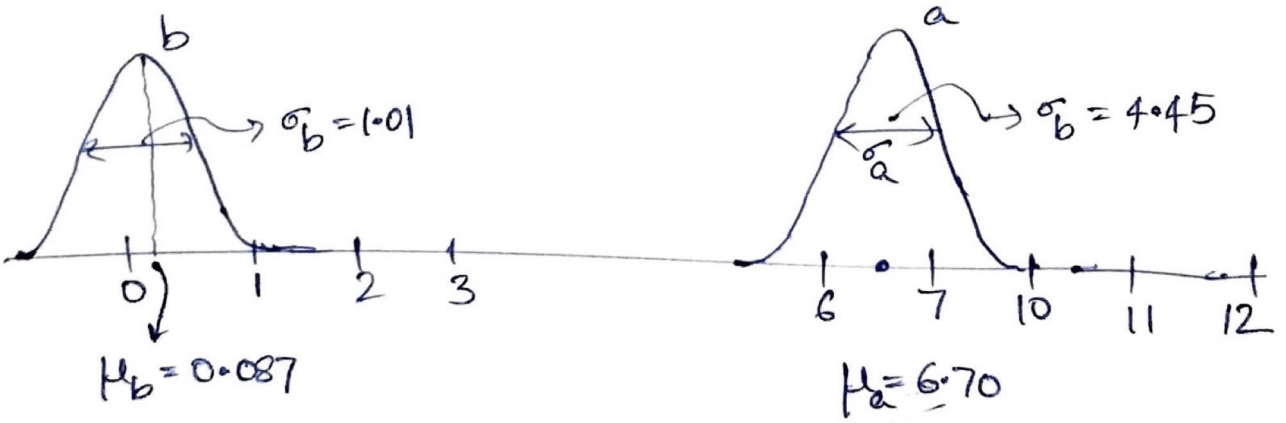


↓ After iteration-2



★ point : Cluster Repeated

↓ After next iteration



As seen in the diagram, the cluster improves every iteration.

It will come to saturation after multiple iteration by changing appropriate μ and σ for both probability distribution

THE END