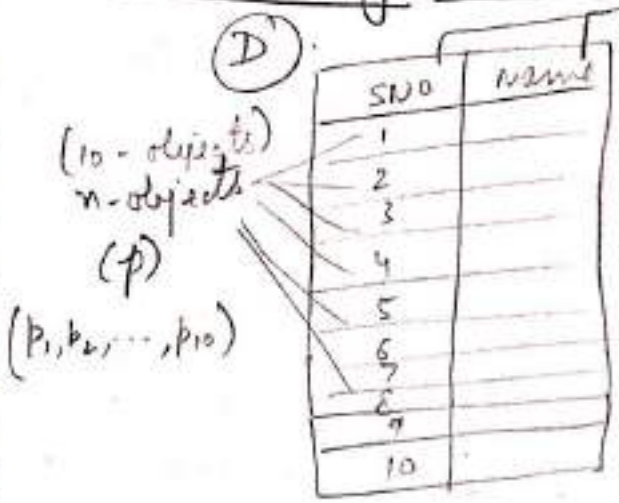
(I) Partitioning methods

d-variables / attributes, (2-variable, d-dimensions (2-d data set)



k - clusters. $(C_1, C_2, \cdots C_k)$

y (2 - cluster).

$k \leq n$

$(2 \leq 10)$

- This method organizes the objects of a set into several exclusive groups or clusters.
- Commonly used Partitioning methods :-
  - @ k-means
  - @ k-medoids

k-means : A Centroid based Technique.

- A Data set 'D' contains 'n' objects.
- Distribute the objects in D into k clusters, $C_1, C_2, \cdots C_k$.
- that is, $C_i \subset D$ & $C_i \cap C_j = \phi$ (for $1 \leq i, j \leq k$)
- objective function is used to assess the partitioning Quality is high intracluster similarity & low intercluster similarity.
- Centroid based partitioning technique uses the centroid (c_i) of the cluster (C_i) to represent that cluster.

- Centroid of a cluster is its center point.
- Centroid can be defined as mean or medoid of the objects (or points) assigned to the cluster
- The difference between an object $\underline{p} \in C_i$ and $\underline{c_i}$ (rep. of cluster), is measured by $\underline{dist(p, c_i)}$
- The Quality of cluster C_i can be measured by the within-cluster variation, which is sum of squared error b/w all objects in C_i and the centroid c_i, defined as

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist(p, c_i)^2 \quad -(1)$$

where, $\underline{E}$ is the sum of the squared error for all objects in the data set(D).

- $\underline{p}$ is point in space representing a given object.
- c_i is centroid of cluster C_i
- In other words, for each object in each cluster, the distance from the object to its cluster center is squared, & the distances are summed.
- This objective f" tries to make the resulting k-clusters as compact & as separate as possib

- If the no. of clusters 'k' and the dimensionality of the space 'd' are fixed, the problem can be solved in time $O(n^{dk+1} \log n)$. (where n is the no. of objects.)
- To overcome the computational cost (that takes so much time) for exact solution, greedy approaches are used in practice such as k-means.

k-means

- k-means algorithm defines the centroid of a cluster as the mean value of the points within the cluster.
- first, it randomly select 'k' of the objects in D, each of which initially represents a cluster mean or center. [If D is div. into k=2 clusters, then select 2 objects that represents individual clusters containing 1 cluster each.]
- for each of the remaining objects, an object is assigned to the cluster to which it is most similar, based on Euclidean distance between the object & the cluster mean.
- The k-mean algorithm then iteratively improves the within-cluster variation.

- For each cluster, it computes the new mean using the objects assigned to the cluster in the previous iteration.
- All the objects are then reassigned using the updated means as the new cluster centers.
- The iterations continue until the assignment is stable, ie, the clusters formed in the current round are the same as those formed in the previous round.

Algorithm - k-means

The mean Algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input :- k: the no. of clusters.
D: A data set containing 'n' objects.

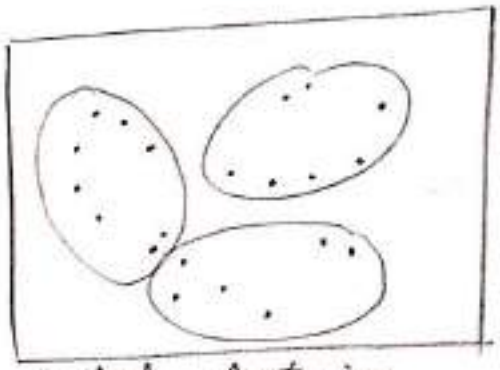
output :- A set of 'k' clusters that minimizes the squared error criterion.

method :-
(1) Arbitrarily choose k objects from D as the initial cluster centers.
(2) repeat
(3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster.
(4) update the cluster means, that is,

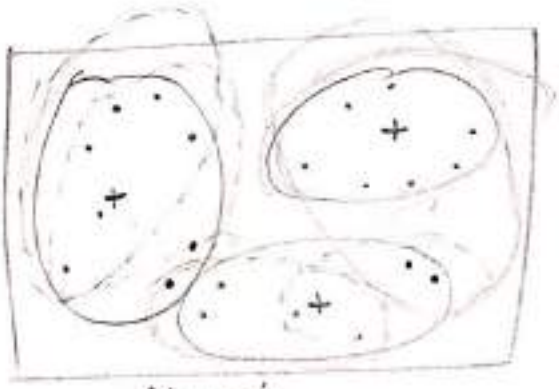
calculate the mean value of the objects for each cluster.

(5) until no change;

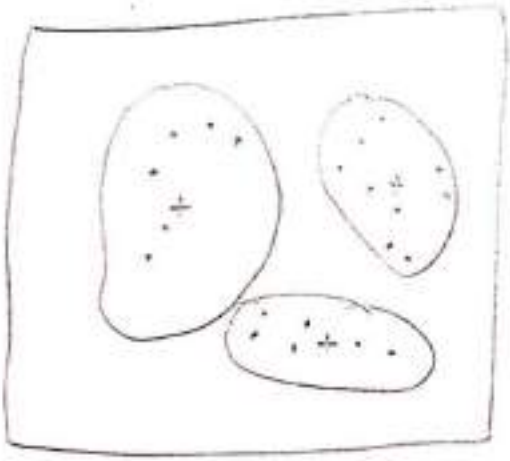
- The process of iteratively reassigning objects to clusters to improve the partitioning is referred to as iterative relocation.



initial clustering



iterate



final clustering

- k-means method is not guaranteed to converge to the global optimum & often terminates at the local optimum (solution).
- The result may depend on the initial random selection of cluster centers. [To obtain good results run K-means multiple times with different cluster initial cluster centers.].
- The time complexity of k-means algorithm is $O(nkt)$,
  $n \rightarrow$ 'n' is total no. of objects.
  $k \rightarrow$ 'k' is no. of clusters.
  $t \rightarrow$ 't' is the no. of iterations. [within a single cluster].
- Normally, $k \ll n$ & $t \ll n$. ∴ the method is relatively scalable & efficient in processing large data sets.

- k-means method can be applied only when the mean of a set of objects is defined (i.e values are numeric)
- But if we have nominal attributes, 'k-modes' method is a variant of k-means

which extends the k-means paradigm to cluster nominal data by replacing the means of clusters with modes (frequency).

- The k-means & the k-modes methods can be integrated to cluster data with mixed numeric & nominal values.

Disadv :-

- k means method is not suitable for discovering clusters of very different size.
- k-means is sensitive to noise & outlier data points because these values can influence the mean value.
- It is necessary for user to specify 'k' in advance i.e the no. of clusters to be formed in advance

Sol :- How to make k-means more scalable?

- To implement k-means method on large data sets, use a good-sized set of samples in clustering.
- Employ a filtering approach that uses a spatial hierarchical data index to save cost when computing means.
- Microclustering Idea :- first groups nearby objects into "microclusters" & then performs k-means clustering on the microclusters.

Drawback of k-means.

Example to show k-means algorithm is sensitive to outliers.

- Consider ~~six~~ points in 1-D space having values 1, 2, 3, 8, 9, 10, 25.
- visually we make clusters {1, 2, 3}, {8, 9, 10} & {25}
- If k=2 (only 2 clusters) then partition is {{1, 2, 3}, {8, 9, 10, 25}}
- mean of cluster {1, 2, 3} = (1+2+3)/3 = 2
- mean of cluster {8, 9, 10, 25} = (8+9+10+25)/4 = 13

€

cluster variation =

$$(1-2)^2 + (2-2)^2 + (3-2)^2 + (8-13)^2 + (9-13)^2 + (10-13)^2 + (25-13)^2$$

$$= \boxed{196}$$

→ Consider the partitioning:-

cluster 1 = {1, 2, 3, 8} & → mean = ~~[illegible]~~ 3.5

cluster 2 = {9, 10, 25} → mean = 14.67

cluster variation :=

$$(1-3.5)^2 + (2-3.5)^2 + (3-3.5)^2 + (8-3.5)^2 + (9-14.67)^2 + (10-14.67)^2 +$$

$$(25-14.67)^2 = \boxed{189.67}$$

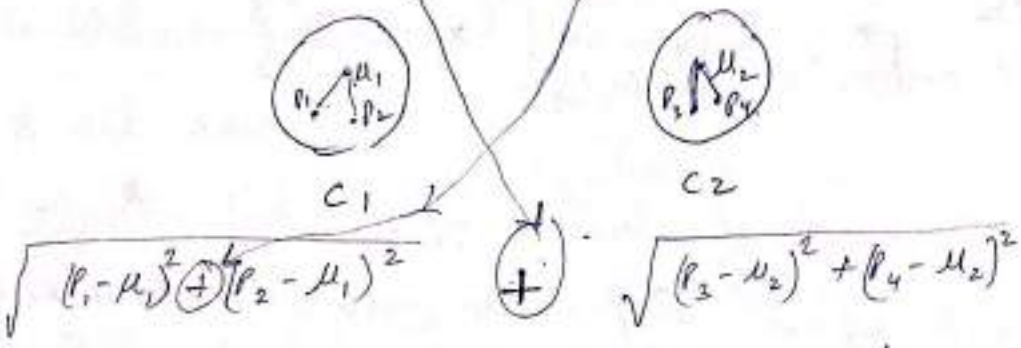
The latter one has the lowest cluster variation, ∴ k-means method assigns the value '8' to a cluster diff. from that containing 9 & 10 due to outlier point 25. Also the center of 2nd cluster '14.67' is far from all the members in the cluster.

K-means Algorithm

Scoring function used is Sum of ~~Squared~~ Squared Error (SSE).

$$SSE(C) = \sum_{i=1}^{K} \sum_{x_j \in C_i} \| x_j - \mu_i \|^2$$

(small → value zero)



The goal is to find the clustering that minimizes the SSE score:

$$C^* = \arg\min_C \{SSE(C)\}$$

[# $\arg\min_x f(x)$: value of x for which $f(x)$ attains minimum value.]

K-means employs a greedy iterative approach to find clustering that minimizes the SSE.

drawback.

It can converge to local optima instead of a globally optimal clustering.

K-means initializes the cluster means by randomly generating K points in the data space.

Each iteration of K-means consists of two steps-

(a) cluster assignment.

(b) centroid update.

Given the K cluster means, in the cluster assignment step, each point $x_j \in D$ is assigned to the closest mean. That is each point x_j is assigned to cluster C_{j^*}.

where $$j^* = \arg\min_{i=1}^{k} \{ \| x_j - \mu_i \|^2 \}$$

given a set of clusters C_i, $i = 1, \ldots, k$ in the centroid update step, new mean values are computed for each cluster from the points in C_i.
The cluster assignment and centroid update steps are carried out iteratively until we reach a fixed point or local minima.

Say, we can stop if $\sum_{i=1}^{k} \| \mu_i^t - \mu_i^{t-1} \|^2 \leq \epsilon$
where $\epsilon > 0$ is the convergence threshold
t denotes the current iteration and μ_i^t denotes the mean for cluster C_i in iteration 't'.

Drawbacks

1). It can converge to local optima instead of a globally optimal clustering.

2). Method starts with a random guess for the initial centroids, K-means run several times & the run with lowest SSE value is chosen to report the final clustering.

3) K-means generate convex-shaped clusters because the region in the data space corresponding to each cluster can be obtained as the intersection of half-spaces resulting from hyperplanes that bisect and are normal to the line segments that join pairs of cluster centroids.

4). In terms of computational complexity,

cluster assignment step takes → $O(nkd)$ time because for each of the n points, we have to compute its distance to each of the k clusters, which takes d operations in d-dimensions.

centroid recomputation step takes → $O(nd)$ time because we have to add at total of "n" d-dimensional points.

5) Hard / crisp clustering

6) sensitive to noise / outliers.

6) Assuming that there are 't' iterations, total time for K-means is given as $O(tnkd)$.

7) I/O cost :- In terms of I/O cost it requires $O(t)$ full D/B scans, because we have to read the entire d/b in each iteration.

Algorithm.

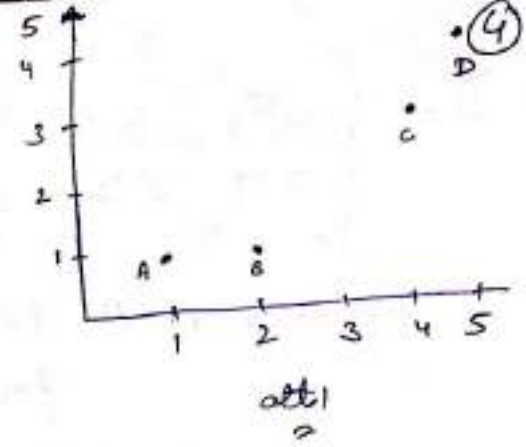
K-means(D, k, ε)

1. $t = 0$
2. Randomly initialize k centroids: $\mu_1^t, \mu_2^t, \ldots, \mu_k^t \in R^d$
3. repeat
4. $t \leftarrow t+1$
5. $C_j \leftarrow \emptyset$ for all $j = 1, \ldots, k$.
6. // cluster assignment step
   for each $x_j \in D$ do
7. $j^* \leftarrow \arg\min_i \{ \|x_j - \mu_i^t\|^2 \}$ // Assign x_j to closest centroid.
8. $C_{j^*} \leftarrow C_{j^*} \cup \{x_j\}$
   // centroid update step.
9. for each $i = 1$ to k do
10. $\mu_i^t \leftarrow \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j$
11. until $\sum_{i=1}^{k} \|\mu_i^t - \mu_i^{t-1}\|^2 \leq \varepsilon$

8). K-means has problems when clusters are of differing -
- sizes
- densities
- Non-globular shapes.

eg:-

objects	att.1	att.2
① medA	1	1
medB	2	1
medC	4	3
medD	5	4



Here $n=4$, $K=2$.

Initialization:

$M_1 = (1,1)$ $\quad M_2 = (2,1)$.

A	B	C	D
1	2	4	5
1	1	3	4

At $t=0$,

$D^0 = \begin{bmatrix} \sqrt{(1-1)^2+(1-1)^2} & \sqrt{(2-1)^2+(1-1)^2} & \sqrt{(4-1)^2+(3-1)^2} & \sqrt{(5-1)^2+(4-1)^2} \\ \sqrt{(1-2)^2+(1-1)^2} & \sqrt{(2-2)^2+(1-1)^2} & \sqrt{(4-2)^2+(3-1)^2} & \sqrt{(5-2)^2+(4-1)^2} \end{bmatrix}$ $E.d = \sqrt{(P_1-P_1)^2+(Q_2-P_2)^2}$

$P(P_1, P_2)$
$Q(Q_1, Q_2)$

(Distance matrix)

$$D_0 = \begin{array}{c} \\ M_1 \\ M_2 \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \end{array}$$

$SSE_0 = 0 + 0 + 2.83 + 4.24 = 7.07$

A → Distance of A from $M_1 = 0$

→ " " A from $M_2 = 1$

∴ A assigned to cluster 1

B → Distance of B from $M_1 = 1$

→ " " B from $M_2 = 0$

∴ B assigned to cluster 2.

similarly C & D assigned to cluster 2.

$G_1^0 = \begin{array}{c} \\ C_1 \\ C_2 \end{array} \begin{array}{c} \begin{array}{cccc} A & B & C & D \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \end{array}$

∴ At $t=0$, $C_1(A)$, $C_2(B,C,D)$.

At $t=1$ $\quad M_1 = (1,1)$ $\quad M_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$

$D^1 = \begin{bmatrix} \sqrt{(1-1)^2+(1-1)^2} & \sqrt{(2-1)^2+(1-1)^2} & \sqrt{(4-1)^2+(3-1)^2} & \sqrt{(5-1)^2+(4-1)^2} \\ \sqrt{\left(1-\frac{11}{3}\right)^2+\left(1-\frac{8}{3}\right)^2} & \sqrt{\left(2-\frac{11}{3}\right)^2+\left(1-\frac{8}{3}\right)^2} & \sqrt{\left(4-\frac{11}{3}\right)^2+\left(3-\frac{8}{3}\right)^2} & \sqrt{\left(5-\frac{11}{3}\right)^2+\left(4-\frac{8}{3}\right)^2} \end{bmatrix}$

$$D^1 = \begin{matrix} A & B & C & D \end{matrix}$$
$$\begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \begin{matrix} C_1 \\ C_2 \end{matrix}$$

Distance of A from $C_1 = 0$ $\quad SSE_1 = 0+1+0.47+1.89 = 3.36$

from $C_2 = 3.14$.

$\therefore$ A assigned to C_1.

Distance of B from $C_1 = 1$

from $C_2 = 2.36$.

$\therefore$ assigned to C_1.

similarly C & D assigned to C_2.

$C_1(A,B)$ $\quad C_2(C,D)$.

$$G^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} C_1 \\ C_2 \end{matrix}$$

$$\begin{bmatrix} A & B & C & D \\ 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix}$$

t = 2

$$M_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right), \quad M_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right)$$

$$M_1 = (1.5, 1) \qquad M_2 = (4.5, 3.5)$$

$$D_2 = \begin{bmatrix} \sqrt{(1-1.5)^2+(1-1)^2} & \sqrt{(2-1.5)^2+(1-1)^2} & \sqrt{(4-1.5)^2+(3-1)^2} & \sqrt{(5-1.5)^2+(4-1)^2} \\ \sqrt{(1-4.5)^2+(1-3.5)^2} & \sqrt{(2-4.5)^2+(1-3.5)^2} & \sqrt{(4-4.5)^2+(3-3.5)^2} & \sqrt{(5-4.5)^2+(4-3.5)^2} \end{bmatrix}$$

$$D_2 = \begin{matrix} A & B & C & D \end{matrix}$$
$$\begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} C_1 \\ C_2 \end{matrix}$$

A → assigned to → C_1

B → " → C_1

C → " → C_2

D → " → C_2

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{matrix} C_1 \\ C_2 \end{matrix}$$

Since $G^1 = G^2$ No more iterations.

$$SSE_2 = \underset{A}{0.5} + \underset{B}{0.5} + \underset{C}{0.71} + \underset{D}{0.71} = 2.42$$

eg-2.

Subject / student	A	B
S1	1	1
S2	1.5	2
S3	3	4
S4	5	7
S5	3.5	5
S6	4.5	5
S7	3.5	4

K-means.

$n = 7, \quad K = 2.$

Sol^n. Initially.

$\mu_1 = (1, 1) \qquad \mu_2 = (5, 7).$

$t=0$

$$D^0 = \begin{matrix} & S1 & S2 & S3 & S4 & S5 & S6 & S7 \\ \end{matrix}$$

$$\begin{bmatrix} 0 & 1.11 & 3.60 & 7.21 & 4.71 & 5.31 & 3.9 \\ 7.21 & 6.10 & 3.60 & 0 & 2.5 & 2.06 & 3.35 \end{bmatrix} \begin{matrix} C_0 \\ C_1 \end{matrix}$$

$$G_1^0 = \begin{matrix} & S1 & S2 & S3 & S4 & S5 & S6 & S7 \\ \end{matrix}$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} C_0 \\ C_1 \end{matrix}$$

$SSE_0 = 12.62$

$t=1$

$$\mu_1 = \left(\frac{1+1.5}{2}, \frac{1+2}{2}\right) = (1.25, 1.5)$$

$$\mu_2 = \left(\frac{3+5+3.5+4.5+3.5}{5}, \frac{4+7+5+5+4}{5}\right) = (3.9, 5)$$

$$D^1 = \begin{matrix} & S1 & S2 & S3 & S4 & S5 & S6 & S7 \\ \end{matrix}$$

$$\begin{bmatrix} 0.0559 & 0.558 & 3.051 & 6.65 & 4.16 & 4.77 & 3.36 \\ 4.94 & 3.841 & 1.345 & 2.28 & 0.4 & 0.6 & 1.077 \end{bmatrix} \begin{matrix} C_0 \\ C_1 \end{matrix}$$

$$G_1^1 = \begin{matrix} & S_1 & S_2 & S_3 & S_4 & S_5 & S_6 & S_7 \\ \end{matrix}$$

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{matrix} C_0 \\ C_1 \end{matrix}$$

$SSE_1 = 6.315$

$G_1^1 = G_1^0$. Hence stop. Also $SSE_1 < SSE_0$.